SIAM J. OPTIMIZATION
Vol. 5, No. 1, pp. 1–12, February 1995

WHY A PURE PRIMAL NEWTON BARRIER STEP MAY BE INFEASIBLE*

MARGARET H. WRIGHT†

Abstract. Modern barrier methods for constrained optimization are sometimes portrayed conceptually as a sequence of inexact minimizations, with only a very few Newton iterations (perhaps just one) for each value of the barrier parameter. Unfortunately, this rosy image does not accurately reflect reality when the barrier parameter is reduced at a reasonable rate, as in a practical (long-step) method. Local analysis is presented indicating why a pure Newton step in a typical long-step barrier method for nonlinearly constrained optimization may be seriously infeasible, even when taken from an apparently favorable point; hence accurate calculation of the Newton direction does not guarantee an effective algorithm. The features described are illustrated numerically and connected to known theoretical results for well-behaved convex problems satisfying common assumptions such as self-concordancy. The contrasting nature of an approximate step to the desired minimizer of the barrier function is also discussed.

Key words. interior method, logarithmic barrier function, primal method, primal Newton step

AMS subject classifications. 65K05, 90C30

1. Introduction.

1.1. Background. Interior methods, most commonly based on barrier functions, have been applied with great practical success in recent years to many constrained optimization problems, especially linear and quadratic programming, and their popularity continues to grow. See, for example, the recent surveys [GON92] and [WR92]. For general nonlinearly constrained problems, an obvious approach is to use Newton's method for unconstrained minimization of the classical logarithmic barrier function.

For some special problem classes, various authors have proved that a pure Newton step is guaranteed to remain feasible and to produce a reduction in the barrier function when a distance measure for the current point—usually, a particular norm of the Newton step—is small enough. Such a characterization was given for linear programming problems in [GON91] and [RV89]. Similar criteria for the Newton step in quadratic and certain convex nonlinear programs are developed in, for example, [NN94], [ADRT], [JAR92], [DRT92], and [DH92]. The results in these papers do not, however, explain why the pure Newton step is unacceptable when the given norm is not sufficiently small.

Most barrier methods used in practice are "long-step" methods, meaning that the controlling barrier parameter is reduced toward zero at a "reasonable" rate (see §2.2). This paper analyzes why a pure Newton step for a typical long-step *primal* barrier subproblem, i.e., an unconstrained minimization subproblem expressed in the original problem variables, is likely to be infeasible, even under circumstances that appear at first to be favorable—in particular, when the current point lies on the barrier trajectory (the path of minimizers of the barrier function; see §1.2). Broadly speaking, infeasibility of the pure Newton step arises because of two factors: the asymptotic role of the active constraint Jacobian matrix in the barrier gradient and Hessian; and the relationship among the optimal multipliers, the active constraint values, and the barrier parameter.

* Received by the editors May 7, 1993; accepted for publication (in revised form) December 16, 1993.

† AT&T Bell Laboratories, Murray Hill, New Jersey 07974 (mhw@research.att.com).

1.2. Notation and assumptions. The problem of interest is

$$\underset{x\in\mathcal{R}^n}{\text{minimize}}\ f(x) \quad \text{subject to} \quad c_j(x) \geq 0,\ \ j = 1, \ldots, m, \tag{1}$$

where f and $\{c_j\}$ are smooth.

Much of our notation is standard. A local solution of (1) is denoted by x^*; $g(x)$ is the gradient of $f(x)$, and $H(x)$ its (symmetric) Hessian; $a_j(x)$ and $H_j(x)$ are the gradient and Hessian of $c_j(x)$; $A(x)$ is the $m \times n$ Jacobian matrix of the constraints, with jth row $a_j(x)^T$. The *Lagrangian function* associated with (1) is $L(x, \lambda) = f(x) - \lambda^T c(x)$. The Hessian of the Lagrangian with respect to x is $\nabla^2 L(x, \lambda) = H - \sum_{j=1}^{m} \lambda_j H_j(x)$.

We let $\hat{m}$ denote the number of constraints active at x^*, and $\mathcal{A}$ the set containing the indices of the active constraints. (It will usually be assumed that $\hat{m} > 0$.) Our discussion will consider only strictly feasible points, so that "active" and "inactive" refer to properties of constraints at x^*, and $\mathcal{A}$ is fixed for any given problem. At the point x, $\hat{A}(x)$ (the "Jacobian of the active constraints") is the $\hat{m} \times n$ matrix whose jth row is the gradient of the jth active constraint evaluated at x. The matrix $Z(x)$ refers to a matrix whose columns form an orthonormal basis for the null space of $\hat{A}(x)$, so that $\hat{A}(x)Z(x) = 0$ and $Z(x)^T Z(x) = I$.

Standard sufficient optimality conditions are assumed to hold at x^*:

(i) $g(x^*) = A^T(x^*)\lambda^*$, where λ^* is called the Lagrange multiplier vector;
(ii) $\lambda_j^* c_j(x^*) = 0$ for $j = 1, \ldots, m$;
(iii) $\lambda_j^* > 0$ if $j \in \mathcal{A}$, i.e., strict complementarity holds at x^*;
(iv) $\hat{A}(x^*)$ has full row rank;
(v) $Z^{*T} W^* Z^*$ is positive definite, where Z^* denotes $Z(x^*)$ and W^* denotes $\nabla^2 L(x^*, \lambda^*)$.

Under these conditions, x^* is an isolated local constrained minimizer of (1) and λ^* is unique; see, for example, [FM68] or [FLE87].

The logarithmic barrier function associated with (1) is

$$B(x, \mu) = f(x) - \mu \sum_{j=1}^{m} \ln c_j(x), \tag{2}$$

where μ is a positive scalar called the *barrier parameter*. This barrier function is defined only at strictly feasible points; it will be assumed that at least one point $\bar{x}$ exists where $c(\bar{x}) > 0$.

The gradient of the barrier function (2), denoted by g_B, is

$$g_B(x, \mu) = g(x) - \sum_{j=1}^{m} \frac{\mu}{c_j(x)}\, a_j(x) = g(x) - \mu A^T(x) C^{-1}(x) e, \tag{3}$$

where $e = (1, 1, \ldots, 1)^T$. The final form in (3) uses the convention that an uppercase version of a letter denoting a vector means the diagonal matrix whose diagonal elements are those of the vector. The barrier Hessian, denoted by H_B, has the form

$$H_B(x, \mu) = H(x) - \sum_{j=1}^{m} \frac{\mu}{c_j(x)}\, H_j(x) + \mu A^T(x) C^{-2}(x) A(x). \tag{4}$$

Assumptions (i)–(v) are well known to imply that, for sufficiently small μ:

(a) an isolated local unconstrained minimizer, denoted by either $x(\mu)$ or x_μ, of the barrier function (2) exists, at which

$$g_B(x_\mu, \mu) = 0, \quad \text{so that} \quad g(x_\mu) = \sum_{j=1}^{m} \frac{\mu}{c_j(x_\mu)}\, a_j(x_\mu); \tag{5}$$

(b) $H_B(x_\mu, \mu)$ is positive definite, and its smallest eigenvalue is bounded away from zero;

(c) if μ is regarded as a continuous parameter, x_μ defines a smooth trajectory converging to x^* as $\mu \to 0$. The points $\{x_\mu\}$ are said to lie on the *barrier trajectory*.

For proofs and additional details about the logarithmic barrier function, see, for example, [FM68], [GON92], and [WR92].

We use the following standard notation; see [PS82]. Let ϕ be a function of a positive variable h, with p fixed. If there exists a constant $\kappa_u > 0$ such that $|\phi| \leq \kappa_u h^p$ for all sufficiently small h, then $\phi = O(h^p)$. If there exists a constant $\kappa_l > 0$ such that $|\phi| \geq \kappa_l h^p$ for all sufficiently small h, then $\phi = \Omega(h^p)$. If there exist constants $\kappa_l > 0$ and $\kappa_u > 0$ such that $\kappa_l h^p \leq |\phi| \leq \kappa_u h^p$ for all sufficiently small h, then $\phi = \Theta(h^p)$.

2. The primal Newton barrier direction.

2.1. Newton's method applied to barrier functions. Suppose that we wish to minimize the barrier function (2) using Newton's method, and that the barrier Hessian is positive definite. Let x be the current iterate; the next iterate is then defined as

$$x + \alpha p_N, \quad \text{where} \quad H_B p_N = -g_B, \tag{6}$$

with H_B and g_B evaluated at x. The vector p_N is called the (primal) *Newton direction*; the modifier "primal" is used to emphasize that only the x variables are treated as independent.

In unconstrained minimization, the positive step α in (6) is chosen using a line search to produce a sufficient decrease in the function being minimized. For the barrier function (2), α must also retain strict feasibility in the next iterate, so that $c(x + \alpha p_N) > 0$.

An iteration for which $\alpha = 1$ in (6) is said to involve a *pure* Newton step. When the Hessian at the solution is positive definite, a successful application of Newton's method near the solution typically takes only pure Newton steps; choosing $\alpha = 1$ in this region is known to produce quadratic convergence as well as a sufficient decrease (see, for example, [DS83], [FLE87]).

Despite the favorable behavior of Newton's method for general problems, Newton's method has been known for many years to be problematical when applied to barrier functions because of inherent ill-conditioning in the Hessian. In [MUR71, LOOT69], it was shown that, when $0 < \hat{m} < n$, the barrier Hessian is ill-conditioned at points on the barrier trajectory for sufficiently small μ, and is asymptotically singular. (This ill-conditioning is one of the factors that led to the decline in popularity of barrier methods in the 1970s.) Recently, it was proved in [WR94] that the barrier Hessian is ill-conditioned in an entire neighborhood of x^*. Although this property might appear to imply that the Newton direction cannot be computed without substantial numerical error, it is also shown in [WR94] how a highly accurate approximation of the Newton step can be calculated near x^*. Active-set strategies for overcoming

the ill-conditioning for nonlinearly constrained problems are discussed in [WR76] and [NS93]; see [GMPS91] for a technique designed for linear and quadratic programs in standard form. Unfortunately, accurate calculation of the Newton direction does not guarantee an effective algorithm, as we shall show in the remainder of this paper.

2.2. Long- and short-step methods. A classical barrier algorithm (see, e.g., [FM68]) typically calculates accurate minimizers of the barrier function for a decreasing sequence of barrier parameters, and so moves from $x(\tilde{\mu})$ to $x(\mu)$, where $\tilde{\mu}$ exceeds μ by some "reasonable" factor, say 10. In more recent practical algorithms, the idea is to improve efficiency by performing only an inexact minimization of the barrier function for each particular barrier parameter. For any given value μ, Newton iterations of the form (6) are executed until some measure of improvement has been achieved; the barrier parameter is then reduced and the process repeated. The hope is that only a very small number of Newton iterations (perhaps even one) will be needed for each value of barrier parameter.

The complexity analyses given in [GON91], [RV89], [NN94], [ADRT], [JAR92], [DRT92], and [DH92] (among others) reflect a broad classification of barrier algorithms as "short-step" and "long-step"; see [GON92] and [DH92] for precise definitions of these terms and surveys of related complexity results. In short-step methods, the barrier parameter is reduced at a sufficiently slow rate to ensure that only a single pure Newton step needs to be performed for each value of the barrier parameter; in these methods, μ is typically multiplied at each step by a factor less than but very close to one—for instance, $1 - 1/(9\sqrt{m})$. In long-step methods, the barrier parameter is reduced by a more generous factor, say $1/10$, but the analysis assumes that several Newton steps (some involving a line search) are carried out for a given barrier parameter. Practical methods for nonlinearly constrained problems are invariably long-step methods.

Nonlinear problems are treated in [NN94], [JAR92], [DRT92], and [DH92] under certain assumptions on the problem functions—for example, convexity and κ-self-concordancy of the barrier function. As defined by Nesterov and Nemirovksii [NN94], a convex function ϕ from a region $\mathcal{F}^0 \in \mathcal{R}^n$ to $\mathcal{R}$ is κ-self-concordant in $\mathcal{F}^0$ if (i) ϕ is three times continuously differentiable in $\mathcal{F}^0$ and (ii) for all $y \in \mathcal{F}^0$ and all $h \in \mathcal{R}^n$, the following inequality holds:

$$| \nabla^3 \phi(y)[h,h,h] | \leq 2\kappa \big(h^T \nabla^2 \phi(y) h\big)^{3/2},$$

where $\nabla^3 \phi(y)[h,h,h]$ denotes the third differential of ϕ at y and h. (The logarithmic barrier functions associated with linear and convex quadratic programming are self-concordant with $\kappa = 1$.) The complexity of long-step methods for suitable convex nonlinear programs can then be analyzed using the H-norm, a distance measure defined using the positive definite barrier Hessian:

$$\|p\|_H^2 = \frac{1}{\mu}\, p^T H_B p; \tag{7}$$

note that our barrier function (2) differs by a factor of $1/\mu$ from those in the cited papers. When p is the Newton direction p_N, it follows from (6) that the H-norm of p_N satisfies

$$\|p_N\|_H^2 = -\frac{1}{\mu}\, p_N^T g_B. \tag{8}$$

The proofs in [NN94], [JAR92], [DRT92], and [DH92] show that a pure Newton step whose H-norm is sufficiently small—say, less than $1/(3\kappa)$—is guaranteed to be strictly feasible and to reduce the barrier function. However, these results do not explain *why* a pure Newton step may be unsuccessful in a long-step method for a general problem; we now broadly analyze the reasons.

2.3. Properties of the barrier gradient and Hessian. It is assumed henceforth that we are performing Newton's method to minimize $B(x,\mu)$ for a general nonlinear problem, that the current iterate is x, and that $\hat{m} > 0$. (Because we do *not* assume convexity and self-concordancy, only local results can be obtained.) The barrier Hessian is presumed to be positive definite at any point of interest; this property is assured for points sufficiently close to the barrier trajectory (see result (b) in §1.2).

We consider strictly feasible points x close to x^* in the sense that

$$c(x) > 0 \quad \text{and} \quad \|x - x^*\| \leq \delta \tag{9}$$

for suitable small δ. Observe that, for small enough δ in (9), each inactive constraint $c_j(x)$ is bounded above because of its smoothness in the closed bounded region defined by the second relation in (9); furthermore, $c_j(x)$ is bounded away from zero because of this same property and the fact that $c_j(x^*) > 0$. Thus, when constraint j is inactive at x^*, $|c_j(x)| = \Theta(1)$ for all x satisfying (9).

Under assumptions (i)–(v) of §1.2, it can be shown that $\|x^* - x(\mu)\| = \Theta(\mu)$; see, e.g., [FM68] or [WR92] for details. Since our intent is to move from x toward $x(\mu)$, $x(\mu)$ should be closer to x^* than x. We thus assume that

$$\mu = O(\delta). \tag{10}$$

Given that x satisfies (9) and μ satisfies (10), we first examine the structure of the barrier gradient (3). Full column rank of $\hat{A}^T(x^*)$, continuity of $g(x)$ and $A(x)$, and optimality conditions (i)–(ii) of §1.2 imply that the objective gradient satisfies

$$g(x) = \hat{A}^T(x)\hat{\lambda}^* + O(\delta), \quad \text{so that} \quad g(x) \approx \hat{A}^T(x)\hat{\lambda}^*. \tag{11}$$

As noted above, the elements of $C(x)$ corresponding to inactive constraints are $\Theta(1)$, so that the quantity $\mu/c_j(x)$ in (3) is $\Theta(\mu)$ when constraint j is inactive. Because all constraint functions are smooth, $\|A(x)\|$ is bounded above for all x satisfying (9). We conclude from the form of (3) and (11) that, near x^* (not necessarily near the barrier trajectory), the barrier gradient lies almost entirely in the range of $\hat{A}^T(x)$:

$$g_B \approx \hat{A}^T\hat{\lambda}^* - \mu\hat{A}^T\hat{C}^{-1}e. \tag{12}$$

Now consider the barrier Hessian (4). In a sufficiently small neighborhood of x^*, smoothness of the constraint functions implies that the portion of the matrix $\mu A^T C^{-2} A$ corresponding to inactive constraints is $\Theta(\mu)$. In analyzing the remainder of the barrier Hessian, we make the further assumption that x is close enough to the barrier trajectory so that the smallest active constraint value is not too small compared to μ; formally, this property means that

$$\min_{j \in \mathcal{A}} c_j(x) = \Omega(\mu), \quad \text{so that} \quad \max_{j \in \mathcal{A}} \frac{\mu}{c_j(x)} = O(1). \tag{13}$$

When (13) applies at x,

$$\left\| H(x) - \sum_{j=1}^{m} \frac{\mu}{c_j(x)} H_j(x) \right\| = O(1),$$

since the quotient μ/c_j is $\Theta(\mu)$ for inactive constraints and $O(1)$ for active constraints. Finally, the matrix $\mu\hat{A}^T\hat{C}^{-2}\hat{A}$ is $O(1/\mu)$ and hence dominates the barrier Hessian, which accordingly resembles a large matrix whose column space is the range of $\hat{A}^T$:

$$H_B \approx \mu\hat{A}^T\hat{C}^{-2}\hat{A}. \tag{14}$$

A more detailed discussion of the nature of the barrier Hessian under various assumptions is given in [WR94].

2.4. Approximating the Newton equations. We have just derived approximate expressions for g_B and H_B that apply when x and μ satisfy (9), (10), and (13). If we use (14) and (12) in (6), the Newton equations "look like" the following relation, which involves only vectors in the range of $\hat{A}^T$:

$$\mu\hat{A}^T\hat{C}^{-2}\hat{A}p_N \approx -\hat{A}^T\hat{\lambda}^* + \mu\hat{A}^T\hat{C}^{-1}e. \tag{15}$$

Since $\hat{A}^T$ has full column rank at x^*, it also has full column rank near x^*, and may be cancelled from both sides of (15). The Newton equations are thus (approximately)

$$\mu\hat{C}^{-2}\hat{A}p_N \approx -\hat{\lambda}^* + \mu\hat{C}^{-1}e. \tag{16}$$

Suppose that the jth constraint is active. The corresponding Newton "almost-equation" from (16) is

$$\frac{\mu}{c_j^2}\, a_j^T p_N \approx -\lambda_j^* + \frac{\mu}{c_j} \quad \text{or}$$

$$a_j^T p_N \approx c_j - \frac{c_j^2\lambda_j^*}{\mu}. \tag{17}$$

We now show why, if x is a previous point on the trajectory (or close to such a point), a pure Newton step is likely to be infeasible. Suppose that x is very close to $x(\tilde{\mu})$ for some suitably small but old barrier parameter $\tilde{\mu}$, where $\tilde{\mu} > \mu$. The full rank of $\hat{A}(x^*)$, optimality conditions (i)–(ii) and relation (5) imply that

$$\left|\frac{\tilde{\mu}}{c_j} - \lambda_j^*\right| = O(\tilde{\mu}), \tag{18}$$

which means that

$$\frac{\tilde{\mu}}{c_j} \approx \lambda_j^*.$$

Substituting for λ_j^* in (17), we obtain a relation that holds approximately for the Newton direction calculated at $x(\tilde{\mu})$ with barrier parameter μ:

$$a_j^T p_N \approx -c_j\Big(\frac{\tilde{\mu}}{\mu} - 1\Big). \tag{19}$$

When $\tilde{\mu}$ exceeds μ by some reasonable factor, i.e., the ratio $\tilde{\mu}/\mu$ is greater than (say) 2, the relationship (19) strongly suggests that $x + p_N$ will be infeasible, for the following reason. The step p_l from x to a zero (the boundary) of the locally linearized

jth constraint satisfies $c_j + a_j^T p_l = 0$, which represents a relation similar in form to (19), but with a coefficient of unity on $-c_j$:

$$a_j^T p_l = -c_j.$$

Since the desired minimizer $x(\mu)$ is strictly inside the boundary, the step p_μ from a point near $x(\tilde{\mu})$ to $x(\mu)$ must move toward but stop short of the boundary. One would therefore expect that, for each active constraint j,

$$a_j^T p_\mu \approx -\gamma c_j, \quad \text{with} \quad 0 \leq \gamma < 1. \tag{20}$$

In contrast, the factor multiplying $-c_j$ on the right-hand side of relation (19) is *larger* than one. Hence the Newton step is likely to move *beyond* the boundary and, consequently, to be infeasible.

Suppose, for example, that μ is smaller than $\tilde{\mu}$ by a factor of 10; this would be typical in a practical (long-step) barrier algorithm. According to (19), the Newton direction satisfies

$$a_j^T p_N \approx -9c_j, \tag{21}$$

and will thus tend to produce substantial infeasibility.

2.5. A numerical example. Consider the following (nonconvex) numerical example:

$$\begin{aligned} \text{minimize} \quad & -\tfrac{1}{8}x_1 + 2x_2 - x_3 \\ \text{subject to} \quad & -\tfrac{1}{2}x_1^2 - x_2^2 - x_3^2 + 5\tfrac{1}{8} \geq 0 \\ & x_2^3 \qquad\quad + 1 \quad \geq 0 \\ & x_1^2 + x_2^2 + x_3 - \tfrac{1}{2} \quad \geq 0. \end{aligned} \tag{22}$$

The optimal solution of interest is $x^* = (\frac{1}{2}, -1, 2)^T$, where the first and second constraints are active, with Lagrange multiplier vector $\lambda^* = (\frac{1}{4}, \frac{1}{2}, 0)^T$. All calculations given in this paper were performed on a Silicon Graphics 4D/440VGX using binary IEEE double-precision arithmetic (around sixteen decimal digits). All displayed numbers are correctly rounded to the number of digits shown.

Let the vector $d(x, \mu)$ be defined as $d_j(x, \mu) = \mu / c_j(x)$, $j = 1, \ldots, m$. For $\mu = 10^{-3}$,

$$x_\mu = \begin{pmatrix} 0.50108 \\ -0.99933 \\ 1.9992 \end{pmatrix}, \quad c_\mu = \begin{pmatrix} 3.9969 \times 10^{-3} \\ 1.9964 \times 10^{-3} \\ 2.7489 \end{pmatrix}, \quad d_\mu = \begin{pmatrix} 0.25019 \\ 0.50089 \\ 3.6378 \times 10^{-4} \end{pmatrix},$$

where c_μ denotes $c(x_\mu)$ and d_μ denotes $d(x_\mu, \mu)$. For $\mu = 10^{-4}$,

$$x_\mu = \begin{pmatrix} 0.50011 \\ -0.99993 \\ 1.9999 \end{pmatrix}, \quad c_\mu = \begin{pmatrix} 3.9997 \times 10^{-4} \\ 1.9996 \times 10^{-4} \\ 2.7499 \end{pmatrix}, \quad d_\mu = \begin{pmatrix} 0.25002 \\ 0.50009 \\ 3.6365 \times 10^{-5} \end{pmatrix}.$$

For $\mu = 10^{-5}$,

$$x_\mu = \begin{pmatrix} 0.500010 \\ -0.999993 \\ 1.99999 \end{pmatrix}, \quad c_\mu = \begin{pmatrix} 3.99997 \times 10^{-5} \\ 1.99996 \times 10^{-5} \\ 2.74999 \end{pmatrix}, \quad d_\mu = \begin{pmatrix} 0.250002 \\ 0.500009 \\ 3.63638 \times 10^{-6} \end{pmatrix}.$$

If we choose x as $x(10^{-3})$, and pick $\mu = 10^{-4}$, so that $\tilde{\mu}/\mu = 10$, we have

$$p_N = \begin{pmatrix} 2.1584 \times 10^{-2} \\ -6.0164 \times 10^{-3} \\ 3.2153 \times 10^{-3} \end{pmatrix}, \quad \hat{A}p_N = \begin{pmatrix} -3.5696 \times 10^{-2} \\ -1.8025 \times 10^{-2} \end{pmatrix},$$

which gives

$$\frac{a_1(x)^T p_N}{c_1(x)} = -8.931 \quad \text{and} \quad \frac{a_2(x)^T p_N}{c_2(x)} = -9.029, \tag{23}$$

as predicted by (21). Taking a step of unity along the Newton direction leads, as expected, to an infeasible point. Relationship (19) also applies for $x = x(10^{-3})$ and $\mu = 10^{-5}$, with $\tilde{\mu}/\mu = 100$; in this case,

$$\frac{a_1(x)^T p_N}{c_1(x)} = -97.8 \quad \text{and} \quad \frac{a_2(x)^T p_N}{c_2(x)} = -99.5.$$

When the barrier Hessian is positive definite, the H-norm of p_N (8) used in complexity analyses should be helpful as a local estimate of the quality of the Newton step. Applying (8) in problem (22) at $x(10^{-3})$ with $\mu = 10^{-4}$, we find that

$$\|p_N\|_H^2 = 161.5,$$

which is certainly *not* small.

Although cutting back the step taken along the Newton direction will eventually restore strict feasibility, such a strategy may end up at a point very near the boundary, where the Hessian will tend to be even more ill-conditioned (see [WR94]) and where the H-norm of the next Newton step is unlikely to be small. A special line search (see, e.g., [MW94]) can help to produce a good next iterate. Our first preference, however, would be to find a direction along which a unit step can be taken with impunity in a close neighborhood of the solution, where all the asymptotic properties of a well-behaved Newton method should apply.

2.6. The barrier trajectory direction. In light of the unfavorable relation (19), an obvious question is what value $a_j^T p$ "should" have if p is the step to $x(\mu)$ from $x(\tilde{\mu})$ rather than the Newton step. Recall that, along the barrier trajectory, the ratio μ/c_j converges to λ_j^*. We would like to choose p such that

$$\frac{\mu}{c_j(x+p)} \approx \lambda_j^*. \tag{24}$$

If, for an *active* constraint j, a good estimate of λ_j^* is available, say λ_j, a search direction p could be required to satisfy a relation like (24), but involving a linearized version of the constraint:

$$\frac{\mu}{c_j + a_j^T p} \approx \lambda_j \quad \text{or} \quad a_j^T p \approx -c_j + \frac{\mu}{\lambda_j}. \tag{25}$$

Linear constraints based on (25) appear in the barrier trajectory algorithm proposed by [WR76], in which the search direction solves an equality-constrained quadratic program with constraints

$$\hat{A}p = -c + \mu \hat{\Lambda}^{-1} e,$$

where $\hat{A}$ is a prediction of the active set and $\hat{\Lambda}$ is a set of associated multiplier estimates.

To see the form of the relationship (25) for points on the trajectory, suppose that the current point is $x(\tilde{\mu})$ for some suitably small $\tilde{\mu}$. The obvious candidate for the jth multiplier estimate is $\tilde{\mu}/c_j$; see (18). With this choice for λ_j, (25) becomes

$$a_j^T p = -c_j\Big(1 - \frac{\mu}{\tilde{\mu}}\Big), \tag{26}$$

and p is likely to stop short of the boundary (as in (20)) rather than produce infeasibility (as for the Newton direction; see (19)).

For problem (22), the step p_{34} from $x(10^{-3})$ to $x(10^{-4})$ and the step p_{35} from $x(10^{-3})$ to $x(10^{-5})$ are

$$p_{34} = \begin{pmatrix} -9.6706 \times 10^{-4} \\ -5.9926 \times 10^{-4} \\ 7.2091 \times 10^{-4} \end{pmatrix} \quad \text{and} \quad p_{35} = \begin{pmatrix} -1.0651 \times 10^{-3} \\ -6.5925 \times 10^{-4} \\ 7.9316 \times 10^{-4} \end{pmatrix}.$$

When $x = x(10^{-3})$ and $\mu = 10^{-4}$,

$$\frac{a_1(x)^T p_{34}}{c_1(x)} = -0.8996 \quad \text{and} \quad \frac{a_2(x)^T p_{34}}{c_2(x)} = -0.8993;$$

since $\mu/\tilde{\mu} = 0.1$, relation (26) holds approximately for the active constraints. Similarly, when $x = x(10^{-3})$ and $\mu = 10^{-5}$, with $\mu/\tilde{\mu} = .01$,

$$\frac{a_1(x)^T p_{35}}{c_1(x)} = -0.9896 \quad \text{and} \quad \frac{a_2(x)^T p_{35}}{c_2(x)} = -0.9893,$$

again approximating (26).

2.7. When a pure Newton step works well. Because the barrier Hessian is positive definite at $x(\mu)$, a pure Newton step must eventually be good when x is sufficiently close to $x(\mu)$. Given the poor results of §2.4 when x is taken as $x(\tilde{\mu})$ (a point that might intuitively appear to be close to $x(\mu)$), an obvious issue is the meaning of "sufficiently close." We have already mentioned the work of [NN94], [JAR92], [DRT92], and [DH92], where it is shown for certain convex problems that the Newton step is guaranteed to be successful when its H-norm is small. It is interesting to consider what information about the Newton direction can be deduced from the properties involving $\hat{A}$ given in §2.4.

Consider relation (17) for the Newton step in the form

$$\mu\Big(1 - \frac{a_j^T p_N}{c_j}\Big) \approx c_j \lambda_j^*.$$

If $|a_j^T p_N|$ is sufficiently small relative to c_j, so that

$$\left|\frac{a_j^T p_N}{c_j}\right| = \beta, \quad \text{where} \quad 0 < \beta < 1, \tag{27}$$

we may make the approximation

$$\frac{1}{1 - \dfrac{a_j^T p_N}{c_j}} \approx 1 + \frac{a_j^T p_N}{c_j} + \ldots. \tag{28}$$

When (28) holds, we may rewrite property (17) satisfied by the Newton direction p_N as

$$\mu \approx c_j\lambda_j^*\Big(1+\frac{a_j^Tp_N}{c_j}\Big) \approx c_j\lambda_j^* + \lambda_j^*a_j^Tp_N,$$

or, after rearrangement,

$$a_j^Tp_N \approx -c_j + \frac{\mu}{\lambda_j},$$

which is the same as relation (25) derived from properties of the barrier trajectory. Following this interpretation, the Newton direction can be a good approximation of the step to $x(\mu)$ only when the ratio $|a_j^Tp_N/c_j|$ is small enough for all active constraints. This corresponds to the property that a unit step along the direction stops short of the boundary of the linearized active constraints; see (20). It should be observed that any property involving a_j^Tp for the active constraints can be viewed as a condition on the component of the search direction in the range space of $\hat{A}^T$, since $\hat{A}p$ is unaffected by the portion of p in the null space of $\hat{A}$.

A property with the same flavor as (27) can also be derived by considering an approximation to the H-norm of the Newton step. Because the barrier Hessian is dominated by $\mu\hat{A}^T\hat{C}^{-2}\hat{A}$ near the trajectory (see (14)), the H-norm of the Newton step satisfies

$$\frac{1}{\mu}\,p_N^TH_Bp_N \approx p_N^T\hat{A}^T\hat{C}^{-2}\hat{A}p_N = \sum_{j\in\mathcal{A}}\Big(\frac{a_j^Tp_N}{c_j}\Big)^2,$$

and will be small only when the ratio $a_j^Tp_N/c_j$ is small in magnitude for every active constraint.

To illustrate these estimates, consider problem (22) with $\mu = 10^{-4}$ and a starting point of $x(10^{-3})$. At the first iteration, the ratio $|a_j^Tp_N/c_j|$ is approximately 9 for both active constraints (see (23)), the squared H-norm of p_N is 161.5, and the pure Newton step is infeasible. For the second iteration, we have

$$\frac{a_1^Tp_N}{c_1} = -3.40, \quad \frac{a_2^Tp_N}{c_2} = -3.36, \quad \|p_N\|_H^2 = 2.29,$$

and once again the Newton direction is infeasible. At iteration 3, the estimates start to become small, namely,

$$\frac{a_1^Tp_N}{c_1} = -0.339, \quad \frac{a_2^Tp_N}{c_2} = -0.285, \quad \text{and} \quad \|p_N\|_H^2 = .209.$$

At this iteration (and all subsequent iterations for this value of μ), the pure Newton step is feasible and produces a sufficient decrease in the barrier function.

3. Conclusions. Complexity analyses of barrier methods for nonlinear convex programming (in particular, [NN94], [JAR92], [DRT92], and [DH92]) indicate that pure Newton steps cannot necessarily be taken successfully unless the barrier parameter is reduced by only a tiny amount. Using local analysis for general nonlinearly constrained problems, we have seen why, following a computationally reasonable (long-step) reduction in the barrier parameter from $\tilde{\mu}$ to μ, a pure Newton step is almost certain to be infeasible even when the initial point is $x(\tilde{\mu})$.

This situation seems unsatisfactory because asymptotic results (see §2.6) indicate that, under these circumstances, we should be able to calculate a good step toward the next barrier minimizer. One way to interpret the difficulty is that a formulation in terms of only the primal variables essentially ignores available information about the Lagrange multiplier estimates.

If one is willing to predict the active set, a barrier trajectory algorithm (see [WR76] and [MW78]) might be appropriate. Techniques based on maintaining multiplier estimates for all the constraints also seem promising; for example, a recently proposed interior method for nonlinear convex programming [JS95] includes tests on the quality of multiplier estimates in its calculation of the search direction.

Finally, a strategy not requiring prediction of the active set is to apply Newton's method to solve some form of the nonlinear primal-dual equations, involving both x and multiplier estimates λ, that hold along the barrier trajectory. For example, the standard primal-dual equations appear in the algorithm proposed by McCormick [McC91] for convex programming:

$$g(x) = A(x)^T\lambda \qquad \text{and} \qquad c_j\lambda_j = \mu, \quad j = 1, \ldots, m.$$

An obvious advantage of a primal-dual formulation is that the associated matrix is not inherently ill-conditioned as the solution is approached. Several Newton-like strategies that are derived from primal-dual relations and avoid the difficulties described here are suggested in [CGT93]. Primal-dual methods for general nonlinearly constrained problems will be considered in more detail in a future paper.

Acknowledgment. I am very grateful to Kurt Anstreicher for pointers to related work. I also thank the referee for several helpful comments and suggestions.

REFERENCES

[ADRT] K. M. Anstreicher, D. den Hertog, C. Roos, and T. Terlaky, *A long step barrier method for convex quadratic programming*, Algorithmica, 10 (1993), pp. 365–382.

[CGT93] A. R. Conn, N. Gould, and Ph. L. Toint, *A note on using alternative second-order models for the subproblems arising in barrier function methods for minimization*, Report RC18898, IBM T. J. Watson Research Center, Yorktown Heights, NY, 1993.

[DH92] D. den Hertog, *Interior Point Approach to Linear, Quadratic and Convex Programming: Algorithms and Complexity*, Ph. D. thesis, Delft University of Technology, Delft, The Netherlands, 1992.

[DRT92] D. den Hertog, C. Roos, and T. Terlaky, *A large-step analytic center method for a class of smooth convex programming problems*, SIAM J. Optim., 2 (1992), pp. 55–70.

[DS83] J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[FM68] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley and Sons, New York, 1968; republished by the Society for Industrial and Applied Mathematics, Philadelphia, PA, 1990.

[Fle87] R. Fletcher, *Practical Methods of Optimization*, 2nd ed., John Wiley and Sons, Chichester, 1987

[GMPS91] P. E. Gill, W. Murray, D. B. Ponceleón, and M. A. Saunders, *Solving reduced KKT systems in barrier methods for linear and quadratic programming*, Report SOL 91-7, Department of Operations Research, Stanford University, Stanford, CA, 1991.

[Gon91] C. C. Gonzaga, *Large step path-following methods for linear programming, part 1: barrier function method*, SIAM J. Optim., 1 (1991), pp. 268–279.

[GON92] C. C. GONZAGA, *Path following methods for linear programming*, SIAM Review, 34 (1992), pp. 167–224.

[JAR92] F. JARRE, *Interior-point methods for convex programming*, Appl. Math. Optim., 26 (1992), pp. 287–311.

[JS95] F. JARRE AND M. A. SAUNDERS, *A practical interior-point method for convex programming*, SIAM J. Optim., 5 (1995), pp. 149–171.

[LOOT69] F. A. LOOTSMA, *Hessian matrices of penalty functions for solving constrained optimization problems*, Philips Res. Repts 24, Eindhoven, The Netherlands, pp. 322–331, 1969.

[MCC91] G. P. MCCORMICK, *The superlinear convergence of a nonlinear primal-dual algorithm*, Report T-550/91, Department of Operations Research, George Washington University, Washington, DC, 1991.

[MUR71] W. MURRAY, *Analytical expressions for the eigenvalues and eigenvectors of the Hessian matrices of barrier and penalty functions*, J. Optim. Theory Appl., 7 (1971), pp. 189–196.

[MW78] W. MURRAY AND M. H. WRIGHT, *Projected Lagrangian methods based on the trajectories of penalty and barrier functions*, Report SOL 78-23, Department of Operations Research, Stanford University, Stanford, CA, 1978.

[MW94] W. MURRAY AND M. H. WRIGHT, *Line search procedures for the logarithmic barrier function*, SIAM J. Optim., 4 (1994), pp. 229–246.

[NS93] S. G. NASH AND A. SOFER, *A barrier method for large-scale constrained optimization*, ORSA J. Comput., 5 (1993), pp. 40–53.

[NN94] Y. NESTEROV AND A. NEMIROVSKII, *Interior-Point Polynomial Algorithms in Convex Programming*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1994.

[PS82] C. R. PAPADIMITRIOU AND K. STEIGLITZ, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, NJ, 1982.

[RV89] C. ROOS AND J. P. VIAL, *Long steps with the logarithmic penalty barrier function in linear programming*, in Economic Decision-Making: Games, Economics and Optimization, J. Gabszevwicz, J. F. Richard, and L. Wolsey, eds., Elsevier, Amsterdam, 1989, pp. 433–441.

[WR76] M. H. WRIGHT, *Numerical Methods for Nonlinearly Constrained Optimization*, Ph.D. thesis, Stanford University, Stanford, CA, 1976

[WR92] ———, *Interior methods for constrained optimization*, in Acta Numerica 1992, A. Iserles, ed., Cambridge University Press, NY, 1992, pp. 341–407.

[WR94] ———, *Some properties of the Hessian of the logarithmic barrier function*, Math. Programming, 67 (1994), pp. 265–295.

SIAM J. OPTIMIZATION
Vol. 5, No. 1, pp. 13–51, February 1995

INTERIOR POINT METHODS IN SEMIDEFINITE PROGRAMMING WITH APPLICATIONS TO COMBINATORIAL OPTIMIZATION *

FARID ALIZADEH†

Abstract. This paper studies the *semidefinite programming* SDP *problem* , i.e., the optimization problem of a linear function of a symmetric matrix subject to linear equality constraints and the additional condition that the matrix be positive semidefinite. First the classical cone duality is reviewed as it is specialized to SDP is reviewed. Next an interior point algorithm is presented that converges to the optimal solution in polynomial time. The approach is a direct extension of Ye's projective method for linear programming. It is also argued that many known interior point methods for linear programs can be transformed in a mechanical way to algorithms for SDP with proofs of convergence and polynomial time complexity carrying over in a similar fashion. Finally, the significance of these results is studied in a variety of combinatorial optimization problems including the general 0-1 integer programs, the maximum clique and maximum stable set problems in perfect graphs, the maximum k-partite subgraph problem in graphs, and various graph partitioning and cut problems. As a result, barrier oracles are presented for certain combinatorial optimization problems (in particular, clique and stable set problem for perfect graphs) whose linear programming formulation requires exponentially many inequalities. Existence of such barrier oracles refutes the commonly believed notion that to solve a combinatorial optimization problem with interior point methods, its linear programming formulation is needed explicitly.

Key words. semidefinite programming, interior point methods, eigenvalue optimization, combinatorial optimization, maximum cliques, perfect graphs, graph partitioning

AMS subject classifications. 90C10, 90C25, 90C27, 15A18, 68R10

1. Introduction.

Consider the following optimization problem that we call the standard *semidefinite programming* (SDP) *problem*:

$$\min\{C \bullet X : \ A_i \bullet X = b_i \ \text{ for } i = 1, \ldots, m \ \text{ and } X \succeq 0\}, \tag{1.1}$$

where C, A_i's, and X are $n \times n$ matrices, and X is symmetric; the $\bullet$ operation is the inner product of matrices: $A \bullet B := \sum_{i,j} A_{ij} B_{ij} = \text{trace } A^T B$; and the "inequality" constraint $\succeq$ indicates the *Löwner* partial order; that is, for real symmetric matrices A and B, $A \succeq B$ (respectively, $A \succ B$), whenever $A - B$ is positive semidefinite (respectively, positive definite.)

The SDP problem is an extension of linear programming (LP). Specifically, if the condition that X is a diagonal matrix is added to the constraint set then (1.1) reduces to linear programming. Semidefinite programs arise in a wide variety of applications from control theory (see [63] and [20]) to combinatorial optimization (see §5 below) as well as structural computational complexity theory (see [21]). The oldest form of semidefinite programming is the evaluation of eigenvalues of a symmetric matrix. In fact, one can reformulate the classical theorems of Rayleigh–Ritz for the largest eigenvalue, and those of Fan for the sum of the first few eigenvalues of a symmetric matrix as semidefinite programs; see [53], [54] and §4. However, for these special cases, techniques of this paper do not seem to be appropriate since better algorithms from both theoretical and pragmatic points of view already exist. Most nontrivial

* Received by the editors October 25, 1991; accepted for publication (in revised form) August 30, 1993. This research was supported in part by National Science Foundation grant CDA-9211106, Air Force Office of Scientific Research grant AFOSR-87-0127, National Science Foundation grant DCR-8420935, and the Minnesota Supercomputer Institute.

† International Computer Science Institute, 1947 Center Street, Berkeley, California, 94704-1105. Present Address, Rutgers University, RUTCOR, P.O. Box 5062, New Brunswick New Jersey 08903-5062 (alizadeh@rutcor.rutgers.edu).

semidefinite programs (those that are not equivalent to evaluation of eigenvalues of a symmetric matrix by a simple transformation) arise in the form of minimizing the largest, or sum of the first few largest, eigenvalues of the matrix X subject to some linear constraints on X. An early example of such problems was studied by Donath and Hoffman in connection with graph bisection and graph partitioning problems [17], [18]; see §5. Cullum, Donath, and Wolfe studied the problem of minimizing the sum of the first few eigenvalues of a linearly constrained matrix in [15]. They analyzed this problem from the point of view of nonsmooth optimization. Also Fletcher studied a similar problem and derived expressions for the subgradients of the sum of the first few eigenvalues of a symmetric matrix and formulated optimality conditions for this problem. In the same spirit as Fletcher, Overton [51] studied the largest eigenvalue of a symmetric matrix as a convex, but nondifferentiable function. Based on earlier work [24], in [51] Overton derived a quadratically convergent algorithm for the problem of minimizing the largest eigenvalue of an affinely constrained matrix. This work was further extended in [52] where both second order methods based on sequential quadratic programming and first order methods based on sequential linear programming for large scale problems were developed.

The algorithms contained in the above works are in the same spirit as the simplex method for LP in that they are all active set methods and traverse the boundary of the feasible set to converge to the optimal solution. For that reason their worst case computational complexity is likely to be at least as bad as that of the simplex method, though in practice they may be quite good.

Semidefinite programs, however, are polynomial time solvable if an a priori bound on the size of their solution is known. This point was implicit in [41] for a special instance of the SDP problem. It was proved in the work of Grötschel, Lovász, and Schrijver, [30]. Polynomial time solvability of SDP is a direct consequence of the general results based on the ellipsoid method for convex programming. The main point essentially is that optimization of a linear function over a convex set endowed with a separation oracle and an a priori bound on the objective can be achieved in polynomial time using the ellipsoid method. For the SDP problem, the separation oracle is to determine whether a given symmetric matrix is positive semidefinite and if not provide a separating hyperplane. Cholesky factorization or eigenvalue and eigenvector evaluations easily provide polynomial time oracles for this task. See [32] for a thorough treatment.

The ellipsoid method, however, has not proven practical in most applications, including SDP. A more recent development is the possibility of using interior point methods to obtain polynomial time algorithms for semidefinite programs. The earliest work in this direction to our knowledge is that of Nesterov and Nemirovskii [48]. In this important work the authors develop a general approach for using interior point methods for solving convex programming problems that is based on the concept of *p-selfconcordant* barrier functions. See [50] for a more recent complete treatment of this subject. Nesterov and Nemirovskii show that for any convex set K that is endowed with a p-selfconcordant barrier function, there is an interior point algorithm that optimizes a linear function on K. Furthermore, every $O(\sqrt{p})$ iteration of this algorithm results in an interior point with half the distance to the optimal solution. As a special case, Nesterov and Nemirovskii show that linear programs with p inequality constraints, quadratic programs with p convex quadratic constraints, and semidefinite programs over $p \times p$ matrices all admit p-selfconcordant barriers. Therefore, the authors extend the revolutionary result of Karmarkar [36] to a rather general class of convex programs.

In this paper we study interior point methods for semidefinite programs from an alternative point of view. Our work [1] started somewhat later than, and independent of, Nesterov and Nemirovskii [48]. Nesterov and Nemirovskii obtain their complexity theorems by specializing their general results to SDP. We, on the other hand, take a specific interior point algorithm for LP (i.e., Ye's projective potential reduction method [66]) and extend it to SDP. Furthermore, we argue that many known interior point LP algorithms can also be transformed into an algorithm for SDP in a *mechanical way*; proofs of convergence and polynomial time computability extend in a similar fashion. Later Jarre in [35] and Vandenberghe and Boyd in [63] developed similar interior point algorithms for special forms of SDP.

Polynomial time interior point methods for SDP have some interesting consequences for combinatorial optimization problems. To solve such a problem by the ellipsoid method, an explicit listing of all of the inequalities in its LP formulation is not needed. Rather, one only needs a *separation oracle* and an initial ellipsoid containing its feasible region to start the process. However, it is generally believed that to apply interior point methods to the same combinatorial optimization problem one needs to have the explicit listing of all of the inequalities in the LP formulation; see [32] and [27]. For instance, Goldfarb and Todd in their survey article on linear programming write:

> "... it appears that its [Karmarkar's new algorithm] theoretical implications are far more limited than those of the ellipsoid method. Indeed, Karmarkar's algorithm requires the linear programming problem to be given explicitly with all its constraints and variables listed, and does not appear directly susceptible to column or constraint generation. Thus it cannot be used to provide polynomial algorithms for several combinatorial optimization problems that have been successfully analyzed by the ellipsoid method."

With hindsight we show that this common belief is not completely accurate. Specifically, in this article we present examples of combinatorial optimization problems whose LP formulations require exponentially many inequalities, and yet one can design interior point algorithms that solve them in polynomial time. In fact, we should emphasize that the general results of Nesterov and Nemirovskii imply that in principle one can apply interior point methods to solve combinatorial optimization problems without explicit knowledge of their LP formulation. All that is required is a polynomial time computable *self-concordant barrier oracle* with a polynomially bounded parameter. The most interesting example is the clique and stable set problem in a class of graphs known as *perfect graphs*. In §5 we construct such a barrier indirectly by an SDP formulation of the problem due to Lovász. This example is particularly interesting because presently no LP formulation of the stable set and clique problems for perfect graphs with polynomially bounded number of inequalities is known.

LP interior point methods have been used by Goldberg et al. [26] to derive *sublinear* time *parallel* algorithms for the bounded weight assignment problem. We show that maximum stable sets for perfect graphs can be computed in randomized sublinear parallel time. Furthermore, based on the work of Lovász and Schrijver [42], we argue that in a branch and bound scheme for 0-1 programs, interior point SDP algorithms may efficiently yield much sharper bounds than possible from LP relaxations of such problems.

In §2 we review the so-called cone duality theory as specialized to semidefinite programs. This theory, though quite classical, is somewhat forgotten in optimization literature. It turns out that at least for SDP, cone duality, which is a generalization of

LP duality, is most appropriate for interior point methods. (This point of view is also expressed in the latest edition of Nesterov and Nemirovskii [50].) In §3 we develop an interior point algorithm which, as we mentioned, is a direct extension of Ye's projective potential reduction method. Furthermore, we propose a recipe to extend *mechanically* many known interior point algorithms for LP into similar algorithms for SDP. In this section we also review some differences between SDP and LP as far as interior point methods and polynomial time algorithms in general are concerned. In §4 we build on the results of Overton and Womersley [53], [54] and derive semidefinite programming formulation for various eigenvalue optimization problems. We also state complementary slackness results for these problems. Finally, in §5 we study some applications of SDP interior point methods to various combinatorial optimization problems. These include 0-1 integer programs of [42], maximum clique and maximum stable set problems in graphs, and various partitioning and cut problems in graphs.

Notation and terminology. Unless otherwise stated the following convention and terminology is used throughout this article.

The SDP problem refers to any optimization problem with any mixture of (symmetric) matrix and scalar-valued variables that has a linear objective function and any combination of linear equality or (either component-wise $\geq$ or Löwner $\succeq$) inequality constraints.

We use lower case boldface letters to name column vectors and upper case letters to name matrices. In particular, $\mathbf{1}$ and $\mathbf{0}$ denote vector of all ones and the zero vector, respectively, and I and 0 denote the identity and zero matrices, respectively. Also, $\mathrm{Diag}(\mathbf{x})$ denotes the diagonal matrix made up of the vector $\mathbf{x}$; $\mathbf{diag}(X)$ is the vector made up of diagonal entries of matrix X. For a vector $\mathbf{x}$, x_j is its jth coordinate; similarly, X_{ij} is the i,j entry of matrix X. We sometimes refer to members of $\Re^n$ as n-vectors. $\Re^{\frac{n\times n}{2}}$ is the set of symmetric $n \times n$ matrices.

The ith largest eigenvalue of a symmetric matrix X is $\lambda_i(X)$ (or sometimes another lower case Greek letter, e.g., $\omega_i(X)$); its ith largest eigenvalue *absolute-valuewise* is $\lambda^i(X)$ or $\omega^i(X)$. The *Löwner* partial order $\succeq$ and the dot product $\bullet$ were defined above; the symbol $\geq$ is used for componentwise comparison between two matrices or two vectors.

For matrices, $\|X\|$ and $\|X\|_2$ are the Frobenius and the spectral norms of X, respectively. Recall that in case of symmetric matrices, $\|X\|_2$ equals the spectral radius $\rho(X) = |\lambda^1(X)|$ and

$$\|X\| = \left(\sum_{i,j} X_{ij}^2\right)^{\frac{1}{2}} = \left(\sum_i \lambda_i(X)^2\right)^{\frac{1}{2}}.$$

For vectors, $\|\mathbf{x}\|$ and $\|\mathbf{x}\|_\infty$ are the Euclidean and the maximum norms of $\mathbf{x}$; also $\|\mathbf{x}\|_p := (\sum |x_i^p|)^{1/p}$ is the p-norm of $\mathbf{x}$.

If A is a $p \times q$ matrix then $\mathbf{vec}\, A$ is a pq column vector made up of columns of A stacked on each other. If $\mathbf{v}$ is a pq-vector then $\mathrm{Mat}_{pq}\, \mathbf{v}$ is a $p \times q$ matrix whose ith column is made up of the entries at $(i-1)p+1$ through ip in $\mathbf{v}$; if p and q are clear from the context we drop them from the subscript. For instance the set of relations $A_i \bullet X = b_i$, for $i = 1, \ldots, m$ may be written as $\mathcal{A}\mathbf{vec} X = \mathbf{b}$, where $\mathcal{A} \in \Re^{m \times n^2}$, that is row i of $\mathcal{A}$ is $\mathbf{vec}^T(A_i)$. Also, $\mathrm{Mat}(\mathcal{A}^T\mathbf{y}) = \sum y_i A_i$.

$A \otimes B$ is the Kronecker product of matrices: if $A \in \Re^{m\times n}$ and $B \in \Re^{p\times q}$ then $A \otimes B \in \Re^{np\times mq}$ is an $m \times n$ block matrix whose i,j block is $a_{ij}B$. We use the

following facts occasionally:

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD) \quad \text{and} \quad \mathbf{vec}(ABC) = (C^T \otimes A)\mathbf{vec}(B).$$

See Graham's text [29] for definitions and properties of Kronocker products.

If I and J are subsets of integers from 1 to p and from 1 to q, respectively, then $A_{I,J}$ is the submatrix of A whose rows are taken from those rows of A indexed by I and whose columns are indexed by J. $A_{I.}$ and $A_{.J}$ indicate rows indexed by I and columns indexed by J, respectively. Also if $A \in \Re^{m \times p}$ and $B \in \Re^{m \times q}$ then $[A|B]$ is an $m \times (p+q)$ matrix whose columns are made up of columns of A followed by columns of B.

We use := to define or name the left-hand side in terms of the right-hand side; in algorithms := is used for assignment.

For any convex cone $\mathcal{K}$, its polar cone $\mathcal{K}^*$ is the set

$$\mathcal{K}^* := \{\mathbf{x} : \text{ for all } \mathbf{a} \in \mathcal{K},\ \mathbf{a}^T\mathbf{x} \geq 0\}.$$

Unless otherwise stated, we use $\mathcal{P}$ for the cone of positive semidefinite matrices. Note that $\mathcal{P}^* = \mathcal{P}$. (This fact is direct consequence of Fejer's theorem in [33].)

$G = (V, E)$ is a simple undirected graph without loops or multiple edges. A *stable set* S in G is a subset of vertices that are mutually nonadjacent. A *clique* K in G is a subset of vertices that are all mutually adjacent. A k-partite graph is one whose vertices can be partitioned into k subsets V_j, for $j = 1, \ldots, k$, where each V_j is a stable set. A *clique covering* of G is a collection K_j, $j = 1, \ldots, k$ of sets of vertices, where each K_j is a clique, and $\cup_j K_j = V$.

2. Duality theory. A duality theory quite similar to that of LP may be constructed for the SDP problem. In this section we state the theory for the standard form SDP. We also include proofs of basic results to make the paper self-contained. Duality theory for more general forms of SDP follows exactly as in LP.

We should mention that LP duality has been extended to optimization problems over convex cones in many works. It is easy to see that any cone $\mathcal{K} \subseteq \Re^n$, which is closed, pointed (that is, $\mathcal{K} \cap (-\mathcal{K}) = \{0\}$), and convex, induces a partial order $\geq_{\mathcal{K}}$ on $\Re^n$: $\mathbf{x} \geq_{\mathcal{K}} \mathbf{y}$ if and only if $\mathbf{x} - \mathbf{y} \in \mathcal{K}$. For instance, the nonnegative orthant and the positive semidefinite matrices induce the componentwise $\geq$ and the Löwner $\succeq$ partial orders, respectively. The duality theory in LP can be extended to generalized LP problems where $\geq_{\mathcal{K}}$ replaces $\geq$ in the primal problem and $\geq_{\mathcal{K}^*}$ replaces $\geq$ in the dual problem.

Duffin in [19] was the first one to study such generalized duality theories. Later, Hurwicz [34], Ben-Israel, Charnes and Kortanek [9], Borwein and Wolkowicz [11], [12], and Wolkowicz [64], among others, developed alternative formulations of the duality theory. For a comprehensive treatment of generalized duality theory from the point of view of infinite dimensional linear programs, see the text of Anderson and Nash [3] and for alternative extensions refer to [11], [12]. It is worth mentioning that Anderson and Nash in [3] study the duality theory from the point of view of basic feasible solutions and extend the "tableau based" proofs of LP duality. The latest version of the Nesterov and Nemirovskii text [50] also treats *cone duality* for the general convex programs. Papers of Overton and Womersley [54] and Fletcher [23] treat duality theory for the eigenvalue optimization problem from the point of view of subdifferentials. Such an approach is related to the Karush–Kuhn–Tucker duality theory and relies on derivatives or subgradients. Also Lovász in [41], Grötschel,

Lovász, and Schrijver [30]–[32], and Shapiro in [61] study more or less the same duality theory as we do, but their treatment is restricted to a special form of SDP.

We now proceed to state and prove duality for the SDP problem. However observe that the following development—in particular, the weak duality Lemma 2.1, Lemma 2.2, the extended Farkas Lemma 2.3, and the strong duality Theorem 2.8—actually apply to generalized LP over any closed, pointed convex cone $\mathcal{K}$ (in the primal) and $\mathcal{K}^*$ (in the dual.)

In semidefinite programming it is convenient to assume that C and A_i are symmetric. There is no loss of generality in this assumption. If C is not symmetric, since $C^T \bullet X = C \bullet X$, we can replace C by $\frac{1}{2}(C + C^T)$. The same argument holds for the A_i's. These assumptions of symmetry allow us to formulate the pair of primal and dual standard SDP problems:

$$
\begin{array}{llll}
\underline{\text{Primal}} & & \underline{\text{Dual}} & \\
\min & C \bullet X & \max & \mathbf{b}^T\mathbf{y} \\
\text{s.t.} & A_i \bullet X = b_i \quad \text{for } i = 1, \ldots, m & \text{s.t.} & C - \sum_{i=1}^m y_i A_i \succeq 0. \\
& X \succeq 0 & &
\end{array} \tag{2.1}
$$

Notice the similarity of the primal and dual SDP pair to the corresponding LP pair. First we state the weak duality lemma.

LEMMA 2.1. *Let X be any feasible matrix for primal and $\mathbf{y}$ any feasible vector for dual. Then $C \bullet X \geq \mathbf{b}^T\mathbf{y}$.*

Proof. We have

$$
\begin{aligned}
C \bullet X - \sum_{i=1}^m b_i y_i &= C \bullet X - \sum_{i=1}^m (A_i \bullet X) y_i \\
&= \left(C - \sum_{i=1}^m y_i A_i \right) \bullet X \\
&\geq 0.
\end{aligned}
$$

The last inequality is true because the inner product of two positive semidefinite matrices is nonnegative due to self-polarity of the positive semidefinite cone. □

We now state generalizations of the Farkas lemma. Such generalizations for arbitrary convex cones have been studied as early as 1958 by Hurwicz [34]. See [3] for references on the history and various extensions of the Farkas lemma to nonpolyhedral cones. Here we study the relevant forms of this lemma in the special case of SDP.

It is not possible to generalize the classical Farkas lemma to nonpolyhedral cones without additional qualifications. The difficulty arises from the fact that affine transformations of closed cones are not necessarily closed, and therefore the appropriate strong forms of separation theorems cannot be invoked. (For polyhedral cones however closedness is preserved under affine transformation.) For our purposes we need to have that the set

$$K_1 := \mathcal{A}(\mathcal{P}) = \{\mathcal{A}\mathbf{vec}X : X \succeq 0\}$$

is closed.[1] One class of sufficient conditions for closedness of K_1 is based on assuming that certain sets associated with $\mathcal{P}$ have nonempty interiors. Such conditions are sometimes referred to as *Slater-type constraint qualifications.* Though these conditions are not the weakest possible, they are sufficient for the purposes of this paper. In

[1] Alternative extensions without closedness assumption are treated in [11], [12], [64].

any case, we need to assume nonemptyness of the interior for both primal and dual problems so that we have a valid interior point algorithm. Furthermore, in §3 we show how any pair of primal and dual semidefinite programs may be transformed into an equivalent pair whose K_1 has nonempty interior as long as an a priori bound on the size of primal and dual feasible sets are known. The following is a lemma of Slater-type constraint qualifications:

LEMMA 2.2. *If* $\mathrm{Mat}(\mathcal{A}^T\mathbf{y}) \succ 0$ *for some* $\mathbf{y} \in \Re^m$, *then* K_1 *is closed.* (*Recall that* $\mathrm{Mat}(\mathcal{A}^T\mathbf{y}) = \sum y_i A_i$.)

Proof. Let $\mathcal{L}$:=$\{\mathrm{Mat}(\mathcal{A}^T\mathbf{y}) : \mathbf{y} \in \Re^m\}$. The condition in the lemma says that

$$\mathcal{L} \cap \mathrm{Int}\, \mathcal{P} \neq \emptyset.$$

Thus any translate of the linear subspace $\mathcal{L}$ also intersects $\mathcal{P}$ and its interior. This is equivalent to saying that every symmetric $n \times n$ matrix can be written as the sum of two matrices, one of which is positive semidefinite and the other belongs to $\mathcal{L}$. Therefore, $\Re^{\frac{n\times n}{2}} = \mathcal{P} + \mathcal{L}$. Taking the polar we have

$$\{0\} = \mathcal{P} \cap \mathcal{L}^{\perp}.$$

Here $\mathcal{L}^{\perp}$ is the set $\{X : \mathcal{A}\,\mathbf{vec}X = \mathbf{0}\}$. Hence we have that $X = 0$ is the only solution of the system $\mathcal{A}\,\mathbf{vec}X = 0$, and $X \succeq 0$ and by Theorem 9.1 of [59, p. 73] we conclude that K_1 is closed. □

Now we state the most common form of the Farkas lemma as given in Schrijver's text [60] and as extended to the positive semidefinite cone.

LEMMA 2.3 (The extended Farkas lemma). *Let* $\mathbf{b} \in \Re^m$ *and* $\mathcal{A} \in \Re^{m\times n^2}$ *be a matrix such that its rows* $\mathcal{A}_{i\cdot}^T = \mathbf{vec}A_i$ *where* A_i *are symmetric for* $i = 1, \ldots, m$. *Furthermore, let there be an* m*-vector* $\mathbf{y}$ *such that* $\mathrm{Mat}(\mathcal{A}^T\mathbf{y}) \succ 0$. *Then there exists a symmetric matrix* $X \succeq 0$, *with* $\mathcal{A}\,\mathbf{vec}X = \mathbf{b}$ *if and only if* $\mathbf{y}^T\mathbf{b} \geq 0$ *for all* $\mathbf{y}$ *for which* $\mathrm{Mat}(\mathcal{A}^T\mathbf{y}) \succeq 0$.

Proof. For the only if part we have,

$$\mathbf{b}^T\mathbf{y} = (\mathcal{A}\,\mathbf{vec}X)^T\mathbf{y} = \mathrm{Mat}(\mathcal{A}^T\mathbf{y}) \bullet X \geq 0.$$

(The last inequality is due to self-polarity of the positive semidefinite cone.) To prove the if part, suppose that the system $\mathcal{A}\,\mathbf{vec}X = \mathbf{b}$ and $X \succeq 0$ is infeasible. Then $\mathbf{b} \notin K_1 = \{\mathcal{A}\,\mathbf{vec}X : X \succeq 0\}$. By Lemma 2.2, K_1 is a closed cone and thus there must exist a hyperplane, specifically a linear half-space, that separates $\mathbf{b}$ and K_1, i.e., there exists some vector $\mathbf{y}$ such that $\mathbf{b}^T\mathbf{y} < 0$ and $(\mathcal{A}\,\mathbf{vec}X)^T\mathbf{y} \geq 0$ for all $X \succeq 0$, (see [59, Thm. 11.7, p. 100]). But this means that $X \bullet \mathrm{Mat}(\mathcal{A}^T\mathbf{y}) \geq 0$ for all $X \succeq 0$, which is equivalent to $\mathrm{Mat}(\mathcal{A}^T\mathbf{y}) \succeq 0$, and the if part of the theorem is proved. □

We may formulate and prove several other variants of the Farkas lemma in a similar vein, all of which are extensions of lemmas for the componentwise inequalities as given, for example, in Schrijver's text [60]. Related extensions for infinite programs have been studied in [34] and [13] and in the case of matrix variables in [14]. In all of these extensions we need to assume either some closedness criteria, or the lemma must be modified by using cones other than $\mathcal{P}$ (as in [64], for instance.) We mention here a few more.

LEMMA 2.4. *Let* $\mathcal{A} \in \Re^{n^2\times m}$ *be a matrix whose columns are linearly independent and are of the form* $\mathbf{vec}A_i$ *for symmetric* A_i, *and* $B \in \Re^{\frac{n\times n}{2}}$. *Assume that there exists some symmetric matrix* $Y \succ 0$ *such that* $(\mathbf{vec}Y)^T\mathcal{A} = \mathbf{0}$. *Then* $\mathrm{Mat}(\mathcal{A}\mathbf{x}) \preceq B$ *has a solution in* $\mathbf{x}$ *if and only if* $B \bullet Y \geq 0$ *for all* $Y \succeq 0$ *for which* $(\mathbf{vec}Y)^T\mathcal{A} = \mathbf{0}$.

LEMMA 2.5. *Let $A \in \Re^{m\times n}$ and $B \in \Re^{m\times m}$. Suppose there exist some matrix Y such that $A^TYA \succ 0$. Then the system $AXA^T = B$ and $X \succeq 0$ has a solution if and only if for all symmetric matrices Y, $A^TYA \succeq 0$ implies that $B \bullet Y \geq 0$.*

LEMMA 2.6. *Let $A \in \Re^{m\times n}$ and $B \in \Re^{m\times m}$. Suppose there exist some matrix Y such that $A^TYA = 0$ and $Y \succ 0$. Then the system $AXA^T \preceq B$ has a solution if and only if for all symmetric matrices $Y \succeq 0$ and $A^TYA = 0$ implies that $B \bullet Y \geq 0$.*

LEMMA 2.7. *Let $A \in \Re^{m\times n}$ and $B \in \Re^{m\times m}$. Suppose there exist some matrix Y such that $A^TYA \succ 0$. Then the system $AXA^T \succeq B$ and $X \succeq 0$ has a solution if and only if for all symmetric matrices $Y \succeq 0$ and $A^TYA \succeq 0$ implies that $B \bullet Y \geq 0$.*

A strong duality theorem similar to LP holds for SDP. We say the primal problem in (2.1) is *feasible* if the set $\{X \in \Re^{\frac{n\times n}{2}} : \mathcal{A}\mathbf{vec}X = \mathbf{b}, \text{ and } X \succeq 0\}$ is nonempty; otherwise we say it is infeasible. Feasibility is defined similarly for the dual in (2.1). Recall that infimum over the empty set is by definition $+\infty$ and similarly supremum over the empty set is $-\infty$. Furthermore, the primal (respectively, dual) problem in (2.1) is *unbounded* if the infimum (respectively, supremum) over the feasible set is $-\infty$ (respectively, $+\infty$).

THEOREM 2.8. *Let*

$$z_1{:=}\inf\{C \bullet X : \mathcal{A}\mathbf{vec}X = \mathbf{b}, \text{ and } X \succeq 0\} \text{ and}$$
$$z_2{:=}\sup\{\mathbf{b}^T\mathbf{y} : C - \mathrm{Mat}(\mathcal{A}^T\mathbf{y}) \succeq 0\}.$$

Assume that there is an m-vector $\mathbf{y}$ such that $\mathcal{A}^T\mathbf{y} \succ 0$. Then $z_2 = z_1$.

Proof. Notice that the dual problem is always feasible, because in the proof of Lemma 2.2 we showed that $\Re^{\frac{n\times n}{2}} = \mathcal{P} + \mathcal{L}$ and, in particular, there are some $\mathbf{y}$ and $S \succeq 0$ such that $\mathrm{Mat}(\mathcal{A}^T\mathbf{y}) + S = C$. If $z_1 = -\infty$ (i.e., the primal problem is unbounded) then by the weak duality lemma $z_2 = -\infty$, and the dual problem is infeasible, which is a contradiction. If $z_2 = +\infty$ (i.e., the dual problem is unbounded) then by the weak duality Lemma 2.1 $z_1 = +\infty$ (i.e., the primal is infeasible) and the theorem is proved. Conversely, if $z_1 = +\infty$, then the primal problem is infeasible and the extended Farkas Lemma 2.3 implies that for some vector $\mathbf{y}_1$ we have

$$\mathrm{Mat}\mathcal{A}^T\mathbf{y}_1 \preceq 0 \text{ and } \mathbf{b}^T\mathbf{y}_1 > 0. \tag{2.2}$$

But (2.2) implies that the dual problem is unbounded since to any dual-feasible pair $\mathbf{y}$ one can add an arbitrarily large positive multiple of $\mathbf{y}_1$ and obtain another feasible solution with larger objective function value. Therefore, $z_2 = z_1 = +\infty$. Thus, we may assume that both z_1 and z_2 are finite. Suppose $z_2 < z_1$. Then the system

$$\begin{aligned} &C \bullet X = z_2, \\ &\mathcal{A}\mathbf{vec}X = \mathbf{b}, \\ &X \succeq 0 \end{aligned}$$

is infeasible. Therefore, by the extended Farkas lemma 2.3, there exists a scalar y_0 and m-vector $\mathbf{y}$ such that

$$y_0C + \sum_{i=1}^{m} y_iA_i \succeq 0 \quad \text{and} \quad z_2y_0 + \mathbf{b}^T\mathbf{y} < 0, \tag{2.3}$$

where $\mathbf{vec}A_i$ is the ith row of $\mathcal{A}$. Now, for y_0 one of the following holds.

1. If $y_0 = 0$, (2.3) is equivalent to

$$\mathrm{Mat}(\mathcal{A}^T\mathbf{y}) \succeq 0 \quad \text{and} \quad \mathbf{b}^T\mathbf{y} < 0,$$

which by the extended Farkas lemma implies that $\mathcal{A}\mathbf{vec}X = \mathbf{b}$ and $X \succeq 0$ is infeasible and thus $z_1 = \infty$.

2. If $y_0 > 0$, then dividing both relations in (2.3) by y_0 we get

$$C - \mathrm{Mat}(\mathcal{A}^T(-\mathbf{y}/y_0)) \succeq 0 \quad \text{and} \quad z_2 - \mathbf{b}^T(-\mathbf{y}/y_0) < 0,$$

which means z_2 is not an optimal solution of the dual problem.

3. If $y_0 < 0$, then dividing both relations in (2.3) by $-y_0$ we get

$$-C + \mathrm{Mat}(\mathcal{A}^T(-\mathbf{y}/y_0)) \succeq 0 \quad \text{and} \quad -z_2 + \mathbf{b}^T(-\mathbf{y}/y_0) < 0.$$

In fact, since we have strict inequality, we must have

$$-C + \mathrm{Mat}(\mathcal{A}^T(-\mathbf{y}/y_0)) \succeq 0 \quad \text{and} \quad -z_2 + \mathbf{b}^T(-\mathbf{y}/y_0) < -\epsilon$$

for some $\epsilon > 0$. But also, by optimality of z_2, there must exist a $\mathbf{y}^*$ such that

$$C - \mathrm{Mat}(\mathcal{A}^T\mathbf{y}^*) \succeq 0 \quad \text{and} \quad z_2 - \mathbf{b}^T\mathbf{y}^* < \epsilon.$$

Adding the last two sets of relations we get

$$\mathrm{Mat}(\mathcal{A}^T(-\mathbf{y}/y_0 - \mathbf{y}^*) \succeq 0 \quad \text{and} \quad \mathbf{b}^T(-\mathbf{y}/y_0 - \mathbf{y}^*) < 0,$$

which again by the extended Farkas lemma implies that the primal problem is infeasible and $z_1 = \infty$. Hence the assumption $z_2 < z_1$ results in contradiction. Since by the weak duality lemma, Lemma 2.1, we have $z_2 \leq z_1$ we conclude that $z_2 = z_1$. □

It is also possible to derive a "complementary slackness" theorem. In fact, Grötschel, Lovász, and Schrijver in [31] and Shapiro in [61] mention the complementary slackness theorem for a more restricted form of SDP. Note that when the strong duality theorem is true and both primal and dual problems are bounded and feasible then the duality gap $X \bullet S$ vanishes. However, in SDP, as in LP, a stronger form of complementary slackness results from this observation. First note the following easy lemma.

LEMMA 2.9. *Let A and B be symmetric $n \times n$ matrices. If $A \succeq 0$, $B \succeq 0$, then $A \bullet B = 0$ if and only if $AB = 0$.*

Proof. Let $B = U\Omega U^T$ be the eigenvalue decomposition of B, with $\Omega = \mathrm{Diag}(\omega_i)$ and $\omega_i \geq 0$ for $i = 1, \ldots, n$. Set $C{:=}U^TAU$, thus $C \succeq 0$, and in particular, its diagonal elements $C_{ii} \geq 0$. We only need to show that $C\Omega = 0$. From $A \bullet B = 0$ we have $C \bullet \Omega = 0$ and therefore, $\sum_{i=1}^n C_{ii}\omega_i = 0$. Since all the summands are nonnegative, it follows that they are all zero. Thus we have the following.

(i) If $\omega_i > 0$ then $C_{ii} = 0$, and by $C \succeq 0$, the entire row and column i is zero.

(ii) If $C_{ii} > 0$, then $\omega_i = 0$.

Now suppose $(C\Omega)_{ij} \neq 0$ for some i, j. Then $C_{ij}\omega_j \neq 0$, which by (i) we must have that the entire column j is zero, and so $C_{ij} = 0$, a contradiction.[2] □

Now the complementary slackness theorem is immediate.

THEOREM 2.10. *Let X^* be a feasible matrix for the primal, and $\mathbf{y}^*$ a feasible vector for the dual in* (2.1). *Define $S^*{:=}C - \mathrm{Mat}(\mathcal{A}^T\mathbf{y}^*)$. Then X^* and $\mathbf{y}^*$ are primal and dual optimal, respectively, if and only if*

$$X^*S^* = 0. \tag{2.4}$$

[2] D. E. Knuth and an anonymous referee suggested the following slightly shorter proof: $0 = A \bullet B = \mathrm{trace}\, A^{1/2}BA^{1/2}$. Since $A^{1/2}BA^{1/2} \succeq 0$ and its trace is zero, the matrix product itself must equal zero, and therefore $AB = 0$. It was pointed out to me that this lemma appears as an exercise in the text of Lancaster and Tismenetsky [39].

Notice that, in contrast with LP, componentwise multiplication in the complementary slackness theorem is replaced by the ordinary matrix multiplication. The complementary slackness theorem for SDP can be restated in the following way which makes it quite similar to the LP variant.

COROLLARY 2.11. *Let X^* be a feasible matrix for the primal problem in* (2.1) *with eigenvalues $\lambda_1, \ldots, \lambda_n$; and $S^*{:=}C - \mathrm{Mat}(\mathcal{A}^T\mathbf{y}^*)$ be feasible for the dual problem with eigenvalues $\omega_1, \ldots, \omega_n$. Then X^* and S^* are primal and dual optimal, respectively, if and only if they commute and there is a permutation π of eigenvalues of S^* such that*

$$\lambda_i \omega_{\pi_i} = 0 \quad \textit{for } i = 1, \ldots, n.$$

Recall our convention that λ_i and ω_i are the ith largest eigenvalues of X and S, respectively; this point necessitates the permutation π in the statement of the corollary.

Proof. X^* and S^* are optimal if and only if $X^*S^* = 0$. Thus, X^* and S^* commute with each other and therefore, they share a system of eigenvectors. Let columns of U be a joint system of orthonormal eigenvectors of X^* and S^*, i.e.,

$$X^* = U\mathrm{Diag}(\lambda_1, \ldots, \lambda_n)U^T \quad \text{and} \quad S^* = U\mathrm{Diag}(\omega_{\pi_1}, \ldots, \omega_{\pi_n})U^T$$

for some permutation π. The corollary follows immediately by multiplying the right-hand sides of these two identities. □

One can extend the notion of *strict complementarity* in LP to SDP. This can be stated by saying that in the preceding corollary *exactly* one of λ_i or ω_{π_i} corresponding to eigenvector $\mathbf{u}_i$ be zero for each $i = 1, \ldots, n$. Equivalently we may require that $\mathrm{Rank}(X^*) + \mathrm{Rank}(S^*) = n$. However, unlike standard LP, where, in the absence of nondegeneracy, one could say that precisely m components of the optimal solution $\mathbf{x}^*$ is nonzero, it is not clear generally how to predict $\mathrm{Rank}(X^*)$ or $\mathrm{Rank}(S^*)$ before solving the SDP problem. All we can say is that $\mathrm{Rank}(X^*) < n$ as the optimum of the primal SDP problem is attained on the boundary of the semidefinite cone. In §4 we encounter another negative effect of the unpredictability of the rank of the optimal solution in the context of interior point methods. We should also mention the paper of Pataki that studies facial structure of feasible sets of SDPs and partially characterizes "degeneracy" in semidefinite programs [55]. Similar to LP, the complementary slackness Theorem 2.10 may be used as a basis for primal-dual algorithms. Indeed in this paper, our interior point algorithm is a primal-dual method that maintains a primal feasible X_k and dual feasible S_k and each iteration moves X_kS_k closer to the zero matrix. The norm $\|X_kS_k\|$ is an indication of how close our current solution is to the optimum. In general, the set of equations

$$\begin{aligned} \mathcal{A}\mathbf{vec}X &= \mathbf{b}, \\ \mathcal{A}^T\mathbf{y} + S &= C, \\ XS &= 0 \end{aligned} \tag{2.5}$$

is a system of $n(n+1)+m$ equations in the same number of unknowns.[3] In the absence of degeneracy one can apply, for instance, Newton's method or some quasi-Newton method to solve this system. Since SDP is a convex program, the real solutions of this system are global optima of the corresponding SDP problem.

As in LP, semidefinite programs may arise in a variety of forms; the standard form (2.1) is just one type. Sometimes we may have positive semidefinite constraints

[3] Actually, one can reduce the number of unknowns by writing $X = U\mathrm{Diag}(\mathbf{x})U^T$ and $S = U\mathrm{Diag}(\mathbf{s})U^T$ and requiring U to be an orthogonal matrix.

TABLE 2.1

Rules for taking dual of a mixed SDP *program. Variables in one program gives rise to constraints in another and vice-versa.*

	MIN			MAX
	matrix or scalar, ≥ 0	$\longleftrightarrow$	C	matrix or scalar, $\leq$
V	matrix or scalar, ≤ 0	$\longleftrightarrow$	O	matrix or scalar, $\geq$
A	matrix, $\succeq 0$	$\longleftrightarrow$	N	matrix $\preceq$
R	matrix, $\preceq 0$	$\longleftrightarrow$	S	matrix, $\succeq$
	matrix or scalar, unrestricted	$\longleftrightarrow$	T	matrix or scalar, $=$
C	matrix or scalar, $\geq$	$\longleftrightarrow$		matrix or scalar, ≥ 0
O	matrix or scalar, $\leq$	$\longleftrightarrow$	V	matrix or scalar, ≤ 0
N	matrix, $\succeq$	$\longleftrightarrow$	A	matrix, $\succeq 0$
S	matrix, $\preceq$	$\longleftrightarrow$	R	matrix, $\preceq 0$
T	matrix or scalar, $=$	$\longleftrightarrow$		matrix or scalar, unrestricted

imposed on linear combinations of matrices (as in the dual problem in (2.1), for example). Sometimes we may have componentwise inequalities $\geq$ on scalar or matrix variables in addition to Löwner inequalities. We may have several matrix expressions constrained to be positive semidefinite. Finally, we may have some or all of these. Of course, as in LP, it is possible to convert all such problems to the standard form, usually by introducing new scalar and matrix variables and new constraints. However, it is more convenient to apply duality directly, as with linear programs in general form. It is easy to show that the rules for obtaining the dual are a straightforward extension of these rules for the LP problem. The main addition is that constraints that involve semidefinite relations on matrix-valued expressions give rise to matrix-valued dual variables with semidefinite constraints. These rules are summarized in Table 2.1; this table is a direct generalization of a similar table in the text of Bazaraa, Jarvis, and Sherali [8].

3. An interior point algorithm. In this section we develop a potential reduction method for solving the primal problem so that, within $O(\sqrt{n}|\log\epsilon|)$ iterations, we get an approximate solution with at least ϵ relative accuracy, if ϵ is sufficiently small. Our development closely follows Ye's projective algorithm for LP [66]. Ye's complexity analysis is also extended to semidefinite programs.

3.1. Potential functions and projective transformations. First, recall that the interior of the cone of positive semidefinite matrices is the set of positive-definite matrices; therefore, all interior points are nonsingular. The boundary of the cone consists of singular semidefinite matrices and so, some of the eigenvalues of the boundary matrices are zero. In particular, optimal solutions of the primal problem in (2.1) are singular.

We assume that there is a positive definite matrix X feasible for the primal and a positive definite matrix S feasible for the dual. Therefore the optimal solution for both primal and dual are finite and is attained on some feasible point. Later, in §3.4, we show how to transform any primal-dual pair to an equivalent one where an initial interior primal–dual solution is available.

Let $q > 0$, and $\underline{z}$ be a given constant known to be a lower bound on the optimal value z^* of the primal problem in (2.1). Let X be an interior primal feasible matrix, $\mathbf{y}$ an interior dual feasible vector, and $S{:=}C - \sum_{i=1}^{m} y_i A_i$; thus, $X \succ 0$ and $S \succ 0$. Define the *primal potential function*

$$\phi(X, \underline{z}) = q \ln(C \bullet X - \underline{z}) - \ln\det X, \tag{3.1}$$

and the *primal-dual potential function*

$$\psi(X,S) = q\ln(X \bullet S) - \ln\det(XS). \tag{3.2}$$

For motivation, one may think of semidefinite constraints $X \succeq 0$ expressed as $\lambda_i(X) \geq 0$ for $i = 1,\ldots,n$. When the standard logarithmic barrier is applied to these constraints we get $\sum_{i=1}^{n} \ln\lambda_i(X) = \ln\det X$.

The strategy of the algorithm is to generate a sequence of interior primal feasible matrices X_k, and a sequence of interior dual vector-matrix pairs $(\mathbf{y}_k, S_k)$, such that the sequence $\psi(X_k, S_k)$ decreases at least like an arithmetic progression. With an appropriate choice of q, this would imply that the duality gap $C \bullet X_k - \mathbf{b}^T\mathbf{y}_k$ decreases at least like a geometric progression with k; in particular, it becomes a constant fraction of the original gap after $O(\sqrt{n})$ iterations.

Before describing the algorithm we state the following lemma which is a direct generalization of a similar lemma that appears in the analysis of most interior point LP methods. (Recall that $\rho(X)$ is the spectral radius of matrix X, which equals its largest eigenvalue when X is positive semidefinite.)

LEMMA 3.1. *Let X be a symmetric $n \times n$ matrix. If $0 \prec X \prec I$, then*

$$\ln\det X \geq \text{trace } X - n - \frac{\text{trace } (X-I)^2}{2[1-\rho(X-I)]}.$$

Proof. In most interior-point LP algorithms it is shown that if $\|\mathbf{x}-\mathbf{1}\|_\infty < 1$, then

$$\sum_{j=1}^{n} \ln x_j \geq (\mathbf{1}^T\mathbf{x} - n) - \frac{\|\mathbf{x}-\mathbf{1}\|^2}{2(1-\|\mathbf{x}-\mathbf{1}\|_\infty)},$$

which is easily proved by expanding $\ln x$, (see, for example, Karmarkar [36] or Ye [67]). Now to prove the lemma, simply substitute $\lambda_j(X)$ for x_j. □

We use a projective transformation to bring the current iterate to the center, except that the center here is the identity matrix (in contrast with LP in which the center is $\mathbf{1}$). An important point is that the transformation should map the set of symmetric matrices to itself. This is needed so that the transformed problem remains a meaningful SDP problem. Let $X_0 \succ 0$ be our current interior primal feasible point. To find a symmetry-preserving projective transformation that maps X_0 to the identity matrix I, let L_0 be any $n \times n$ matrix such that $L_0 L_0^T = X_0$. There are infinitely many choices for L_0. For instance, it could be a Cholesky factor of X_0 or it could be its square root, $X_0^{1/2}$. We shall see shortly that it does not matter how we select L_0 as it will not affect the algorithm's behavior and performance. Fix integer r. Define $\mathcal{T} : \Re^{\frac{n\times n}{2}} \to \Re^{\frac{n\times n}{2}} \times \Re^r$, such that $\mathcal{T}(X) = (\overline{X}, \overline{\mathbf{x}})$. Then

$$\overline{X} := \frac{(n+r)L_0^{-1} X L_0^{-T}}{r + X_0^{-1} \bullet X} \quad \text{and} \quad \overline{\mathbf{x}} := \left(\frac{n+r}{r + X_0^{-1} \bullet X}\right)\mathbf{1}. \tag{3.3}$$

Also, the inverse transformation is given by

$$X = \mathcal{T}^{-1}(\overline{X}, \overline{\mathbf{x}}) := \frac{L_0 \overline{X} L_0^T}{\sum \overline{x}_j / r}. \tag{3.4}$$

Under $\mathcal{T}$, the primal SDP problem is transformed into

$$\begin{array}{ll} \min & \overline{C} \bullet \overline{X} + \overline{\mathbf{c}}(\underline{z})^T \overline{\mathbf{x}} \\ \text{s.t.} & \overline{\mathcal{A}}\mathbf{vec}\,\overline{X} + \overline{A}\overline{\mathbf{x}} = \mathbf{0}, \\ & \text{trace}\,\overline{X} + \mathbf{1}^T\overline{\mathbf{x}} = n + r, \\ & \overline{X} \succeq 0, \\ & \overline{\mathbf{x}} \geq \mathbf{0}, \end{array} \tag{3.5}$$

where

$$\overline{C} := L_0^T C L_0, \tag{3.6}$$

$$\overline{\mathbf{c}}(\underline{z}) := -(\underline{z}/r)\,\mathbf{1}, \tag{3.7}$$

$$\overline{A}_i := L_0^T A_i L_0, \tag{3.8}$$

$$\overline{\mathcal{A}} := \mathcal{A}(L_0 \otimes L_0), \tag{3.9}$$

$$\overline{A} := (-1/r)\,\mathbf{b}\mathbf{1}^T. \tag{3.10}$$

Note that $\overline{A}$ is an $m \times r$ rank-one matrix. The transformed problem may be viewed as a mixed linear and semidefinite program. We may define the following primal potential function for the transformed problem

$$\overline{\phi}(\overline{X}, \overline{\mathbf{x}}, \underline{z}) := q \ln\left[\overline{C} \bullet \overline{X} + \overline{\mathbf{c}}(\underline{z})^T \overline{\mathbf{x}}\right] - \ln\det \overline{X} - \sum_{j=1}^{r} \ln \overline{x}_j. \tag{3.11}$$

The following invariant property holds for the potential functions under projective transformations.

LEMMA 3.2. *If* $\overline{x}_1 = \cdots = \overline{x}_r$, *and* $q = n + r$ *and* $X := \mathcal{T}^{-1}(\overline{X}, \overline{\mathbf{x}})$ *then*

$$\phi(X, \underline{z}) - \phi(X_0, \underline{z}) = \overline{\phi}(\overline{X}, \overline{\mathbf{x}}, \underline{z}) - \overline{\phi}(I, \mathbf{1}, \underline{z}). \tag{3.12}$$

Also, the following result is easily proved by expanding $\overline{\phi}$ and applying Lemma 3.1; later we use it to prove the reduction in the primal-dual potential function.

COROLLARY 3.3. *For* $q = n + r$ *we have*

$$\overline{\phi}(\overline{X}, \overline{\mathbf{x}}, \underline{z}) - \overline{\phi}(I, \mathbf{1}, \underline{z}) \leq (n+r) \ln\left(\frac{\overline{C} \bullet \overline{X} + \overline{\mathbf{c}}(\underline{z})^T \overline{\mathbf{x}}}{\text{trace}\,\overline{C} + \overline{\mathbf{c}}(\underline{z})^T \mathbf{1}}\right) + \frac{\|\overline{X} - I\|^2 + \|\overline{\mathbf{x}} - \mathbf{1}\|^2}{2(1 - \|\overline{X} - I\| + \|\overline{\mathbf{x}} - \mathbf{1}\|)}.$$

3.2. A potential reduction algorithm. Similar to LP, in (3.5) we replace the inequality constraints $\overline{X} \succeq 0$ and $\overline{\mathbf{x}} \geq \mathbf{0}$ by an inscribed "ball" constraint, except that for the SDP problem the ball is centered at $(I, \mathbf{1})$. Therefore, (3.5) is replaced by the "ball optimization" problem

$$\begin{array}{ll} \min & \overline{C} \bullet \overline{X} + \overline{\mathbf{c}}(\underline{z})^T \overline{\mathbf{x}} \\ \text{s.t.} & \overline{\mathcal{A}}\mathbf{vec}\,\overline{X} + \overline{A}\overline{\mathbf{x}} = \mathbf{0}, \\ & \text{trace}\,\overline{X} + \mathbf{1}^T\overline{\mathbf{x}} = n + r, \\ & \|X - I\|^2 + \|\mathbf{x} - \mathbf{1}\|^2 \leq \beta^2 < 1, \end{array} \tag{3.13}$$

where β is a fixed constant between 0 and 1 to be determined shortly. Once we solve this problem and map the result back to the original space, we get a point that serves as a candidate for the next iterate. The solution of (3.13) is given by

$$\begin{pmatrix} \mathbf{vec}\overline{X}_1 \\ \overline{\mathbf{x}}_1 \end{pmatrix} := \begin{pmatrix} \mathbf{vec}I \\ \mathbf{1} \end{pmatrix} - \beta \frac{P(\underline{z})}{\|P(\underline{z})\|}, \tag{3.14}$$

and the candidate for the new primal iterate is given by

$$X(\underline{z}) := \mathcal{T}^{-1}(\overline{X}_1, \overline{\mathbf{x}}_1), \tag{3.15}$$

where

$$P(\underline{z}) := \mathcal{P}_{\mathcal{A}'} \begin{pmatrix} \mathbf{vec}\overline{C} \\ \overline{\mathbf{c}}(\underline{z}) \end{pmatrix}, \quad \mathcal{A}' := \begin{pmatrix} \overline{\mathcal{A}} & \overline{A} \\ (\mathbf{vec}\, I)^T & \mathbf{1}^T \end{pmatrix}, \tag{3.16}$$

and $\mathcal{P}_{\mathcal{A}'}(\mathbf{u})$ is the projection of the (n^2+r)-vector $\mathbf{u}$ to the null space of $\mathcal{A}'$. After expansion, the projection $\mathcal{P}_{\mathcal{A}'}$ in (3.16) becomes

(3.17)

$$P(\underline{z}) = \left(I - \frac{[\mathbf{vec}^T I | \mathbf{1}^T][\mathbf{vec}^T I | \mathbf{1}^T]^T}{n+r}\right)\left(I - [\overline{\mathcal{A}}|\overline{A}]^T([\overline{\mathcal{A}}|\overline{A}][\overline{\mathcal{A}}|\overline{A}]^T)^{-1}[\overline{\mathcal{A}}|\overline{A}]\right)\begin{pmatrix} \mathbf{vec}\overline{C} \\ \overline{\mathbf{c}}(\underline{z}) \end{pmatrix}.$$

Now define

$$\begin{aligned} \mathbf{y}(\underline{z}) &:= \left([\overline{\mathcal{A}}|\overline{A}][\overline{\mathcal{A}}|\overline{A}]^T\right)^{-1}[\overline{\mathcal{A}}|\overline{A}]\begin{pmatrix} \mathbf{vec}\overline{C} \\ \overline{\mathbf{c}}(\underline{z}) \end{pmatrix} \\ &= \left(\mathcal{A}(X_0 \otimes X_0)\mathcal{A}^T + (1/r)\,\mathbf{b}\mathbf{b}^T\right)^{-1}[\mathcal{A}\mathbf{vec}(X_0 C X_0) + (\underline{z}/r)\,\mathbf{b}] \end{aligned} \tag{3.18}$$

and

$$S(\underline{z}) := C - \mathrm{Mat}(\mathcal{A}^T\mathbf{y}(\underline{z})). \tag{3.19}$$

$S(\underline{z})$ and $\mathbf{y}(\underline{z})$ serve as candidates for the new dual iterates. In terms of these quantities, $P(\underline{z})$ may be written as

$$P(\underline{z}) = \begin{pmatrix} \mathbf{vec}(L_0^T S(\underline{z}) L_0) \\ \frac{\mathbf{b}^T\mathbf{y}(\underline{z}) - \underline{z}}{r}\mathbf{1} \end{pmatrix} - \frac{C \bullet X_0 - \underline{z}}{n+r}\begin{pmatrix} \mathbf{vec}I \\ \mathbf{1} \end{pmatrix}. \tag{3.20}$$

Observe that $X(\underline{z})$, $S(\underline{z})$, and $\mathbf{y}(\underline{z})$ are all independent of L_0; in fact in actual computation we do not need to have L_0 explicitly.

The main result to be proved is that first, at least one of the following holds.

1. Either $X(\underline{z}) \succ 0$ and thus primal feasible, or
2. $S(\underline{z}) \succ 0$ and therefore $(\mathbf{y}(\underline{z}), S(\underline{z}))$ is dual feasible.

Second, choosing either one of the feasible candidates reduces the value of the primal-dual potential function ψ by a constant amount. Observe that $\mathcal{P}_{\mathcal{A}'}$ is a projector, that is, $\mathcal{P}_{\mathcal{A}'}^2 = \mathcal{P}_{\mathcal{A}'}$. Therefore, from (3.14) we get

$$\overline{C} \bullet (\overline{X}_1 - I) + \overline{\mathbf{c}}(\underline{z})^T(\overline{\mathbf{x}} - \mathbf{1}) = -\beta\|P(\underline{z})\|.$$

Hence, noting that $\ln(1+x) \le x$, for nonnegative x, Corollary 3.3 implies the following.

COROLLARY 3.4. *Let $q = n + r$ and $\overline{X}_1$ and $\overline{\mathbf{x}}$ be as in* (3.14). *Then*

$$\overline{\phi}(\overline{X}, \overline{\mathbf{x}}, \underline{z}) - \overline{\phi}(I, \mathbf{1}, \underline{z}) \le -(n+r)\beta\frac{\|P(\underline{z})\|}{\mathbf{c}(\underline{z})^T\mathbf{1} + \mathrm{trace}\ \overline{C}} + \frac{\beta^2}{2(1-\beta)}.$$

Let Δ_0 be the size of the duality gap in the current iterate; that is,

$$\Delta_0 := C \bullet X_0 - \underline{z}$$

and let

$$\Delta_1 := S(\underline{z}) \bullet X_0 = C \bullet X_0 - \mathbf{b}^T \mathbf{y}(\underline{z}).$$

Thus Δ_1 should be interpreted as the value of the duality gap if we choose $\mathbf{y}(\underline{z})$ as our new dual iterate. Before deriving the amount of reduction in the potential function we prove the following lemma.

LEMMA 3.5. *If there is some real number α with $0 < \alpha < 1$, such that*

$$\|P(\underline{z})\| \leq \alpha \frac{\Delta_0}{n+r},$$

then $S(\underline{z}) \succ 0$, and $\mathbf{b}^T \mathbf{y}(\underline{z}) > \underline{z}$. Furthermore,

$$\left\| L_0^T S(\underline{z}) L_0 - \frac{\Delta_1}{n} I \right\| \leq \frac{\Delta_1}{n} \alpha \sqrt{\frac{n + n^2/r}{n + n^2/r - \alpha^2}} \tag{3.21}$$

and

$$\left| \frac{n+r}{n} \frac{\Delta_1}{\Delta_0} - 1 \right| \leq \frac{\alpha}{\sqrt{n + n^2/r}}. \tag{3.22}$$

Proof. Suppose $S(\underline{z}) \not\succ 0$. Then $L_0^T S(\underline{z}) L_0$ is not positive definite and so some of its eigenvalues are less than or equal to 0. Thus, from (3.20) we have

$$\|P(\underline{z})\| \geq \rho \left(\frac{\Delta_0}{n+r} I - L_0^T S(\underline{z}) L_0 \right) \geq \frac{\Delta_0}{n+r},$$

a contradiction. Also, If $\mathbf{b}^T \mathbf{y}(\underline{z}) \leq \underline{z}$ then from (3.20) we have

$$\|P(\underline{z})\| \geq \frac{\Delta_0}{n+r} - \frac{\mathbf{b}^T \mathbf{y}(\underline{z}) - \underline{z}}{r} \geq \frac{\Delta_0}{n+r},$$

which is again a contradiction. Now from (3.20) we have

$$P(\underline{z}) = \begin{pmatrix} \left[\mathbf{vec}(L_0^T S(\underline{z}) L_0) - \frac{\Delta_1}{n} I \right] - \left[\frac{\Delta_0}{n+r} - \frac{\Delta_1}{n} \right] \mathbf{vec} I \\ \left[\frac{\Delta_0 - \Delta_1}{r} - \frac{\Delta_0}{n+r} \right] \mathbf{1} \end{pmatrix}.$$

Since $I \bullet [(L_0^T S(\underline{z}) L_0) - (\Delta_1/n) I] = 0$, we have

$$\begin{aligned} \|P(\underline{z})\|^2 &= \left\| L_0^T S(\underline{z}) L_0 - \frac{\Delta_1}{n} I \right\|^2 + n \left(\frac{\Delta_0}{n+r} - \frac{\Delta_1}{n} \right)^2 + r \left(\frac{\Delta_0 - \Delta_1}{r} - \frac{\Delta_0}{n+r} \right)^2 \\ &= \left\| L_0^T S(\underline{z}) L_0 - \frac{\Delta_1}{n} I \right\|^2 + \left(n + \frac{n^2}{r} \right) \left(\frac{\Delta_1}{n} - \frac{\Delta_0}{n+r} \right)^2. \end{aligned} \tag{3.23}$$

If (3.21) is false, then from (3.23) we have

$$\begin{aligned} \|P(\underline{z})\|^2 &> \left(\frac{\Delta_1}{n} \right)^2 \alpha^2 \frac{n + n^2/r}{n + n^2/r - \alpha^2} + \left(n + \frac{n^2}{r} \right) \left(\frac{\Delta_1}{n} - \frac{\Delta_0}{n+r} \right)^2 \\ &\geq \alpha^2 \left(\frac{\Delta_0}{n+r} \right)^2. \end{aligned} \tag{3.24}$$

(The last inequality is proved by taking the right-hand side of the first inequality as a quadratic function in Δ_1/n and minimizing it.) But (3.24) contradicts the assumption of the lemma, so (3.21) must be true. Finally, since (3.24) is false, we have

$$\left(n+\frac{n^2}{r}\right)\left(\frac{\Delta_1}{n}-\frac{\Delta_0}{n+r}\right)^2 \le \alpha^2\left(\frac{\Delta_0}{n+r}\right)^2,$$

from which (3.22) follows. □

Now we may prove the potential reduction theorem.

THEOREM 3.6. *Let X_0 be any interior feasible matrix for the primal problem (2.1) and $\mathbf{y}_0$ interior feasible for the dual. Let also, $r:=\lceil\sqrt{n}\rceil$ and $q:=n+r$, $S_0:=C-\sum_{i=1}^m y_i A_i$, $\underline{z}_0:=\mathbf{b}^T\mathbf{y}_0$, $X(\underline{z}):=\mathcal{T}^{-1}(\overline{X}_1,\overline{\mathbf{x}}_1)$, as in (3.4), $\mathbf{y}_1:=\mathbf{y}(\underline{z}_0)$, and $S_1:=S(\underline{z}_0)$. Then there exists an absolute constant δ such that either*

$$\psi(X(\underline{z}),S_0) \le \psi(X_0,S_0)-\delta$$

or

$$\psi(X_0,S_1) \le \psi(X_0,S_0)-\delta.$$

Furthermore, if we set $\alpha=0.55$ and $\beta:=0.3$, then $\delta>0.1$.

Proof. If for some constant, $0<\alpha<1$

$$\|P(\underline{z})\| \ge \alpha\frac{\Delta_0}{n+r},$$

then

$$\begin{aligned}\psi(X(\underline{z}),S_0)-\psi(X_0,S_0) &= \phi(X(\underline{z}),\underline{z}_0)-\phi(X_0,\underline{z}_0)\\ &= \overline{\phi}(\overline{X}(\underline{z}),\overline{\mathbf{x}}_1,\underline{z}_0)-\overline{\phi}(I,\mathbf{1},\underline{z}_0)\\ &\le -\beta\alpha+\frac{\beta^2}{2(1-\beta)}\end{aligned}$$

(the last inequality is true by Corollary 3.4). Otherwise, the conditions of Lemma 3.5 are satisfied. Also applying Lemma 3.1 to $(n/\Delta_1)L_0^T S_1 L_0$, and setting $\gamma:=\alpha\sqrt{\frac{n+n^2/r}{n+n^2/r-\alpha^2}}$ we have

$$\begin{aligned}n\ln X_0\bullet S_1-\ln\det X_0S_1 &= n\ln\left(\frac{nX_0\bullet S_1}{\Delta_1}\right)-\ln\det\frac{nX_0S_1}{\Delta_1}\\ &= n\ln n-\ln\det\frac{nX_0S_1}{\Delta_1}\\ &\le n\ln n+\frac{\|nL_0^TS_1L_0/\Delta_1-I\|^2}{2(1-\|nL_0^TS_1L_0/\Delta_1-I\|)}\\ &\le n\ln X_0\bullet S_0-\ln\det X_0S_0+\frac{\gamma^2}{2(1-\gamma)},\end{aligned}\tag{3.25}$$

where the last relation results from applying the arithmetic-geometric mean inequality to the eigenvalues of X_0S_0 (which are all real). By (3.22) of Lemma 3.5 we have

$$\Delta_1<\left(1-\frac{r}{n+r}-\frac{n}{n+r}\frac{\alpha}{\sqrt{n+n^2/r}}\right)\Delta_0.$$

Thus,

$$r \ln \frac{X_0 \bullet S_1}{X_0 \bullet S_0} = r \ln \frac{\Delta_1}{\Delta_0} \leq \frac{r^2}{n+r} \left(-1 + \frac{\alpha}{\sqrt{r + r^2/n}} \right). \tag{3.26}$$

Adding (3.25) and (3.26) we get

$$\psi(X_0, S_1) - \psi(X_0, S_0) \leq \frac{r^2}{n+r} \left(-1 + \frac{\alpha}{\sqrt{r + r^2/n}} \right) + \frac{\gamma^2}{2(1-\gamma)}. \tag{3.27}$$

It is easily verified that choice of $\alpha = 0.55$, $\beta = 0.3$, and $\delta = 0.1$ is consistent with all the conditions of the theorem. □

Based on this result we present the projective version of the algorithm displayed in Fig. 3.1. Note that in this algorithm, β^* and $\underline{z}^*$ are obtained by line search on the potential function. We justify this in the next subsection. Also it should be realized that this algorithm is only a prototype and in a practical implementation one must apply substantial simplifications to eliminate redundant use of storage and algebraic operations, especially regarding symmetric matrices.

ALGORITHM SDP:
Input:
An $n \times n$ matrix X_0, interior feasible for the primal problem in (2.1);
an m-vector $\mathbf{y}_0$ interior feasible for the dual problem;
a constant $\epsilon > 0$.
Output:
A primal feasible solution X and dual feasible solution $\mathbf{y}$ such that
$C \bullet X - \mathbf{b}^T\mathbf{y} < \epsilon$.
Method:
1. Set $k = 0$ and $\alpha = 0.55$.
2. Set $\underline{z}_0 = \mathbf{b}^T\mathbf{y}_0$.
3. Set S_k:=$C - \text{Mat}(\mathcal{A}^T\mathbf{y}_0)$.
4. While $C \bullet X_k - \mathbf{b}^T\mathbf{y}_k \geq \epsilon$ do
begin
Compute $S(\underline{z}_k)$ from (3.19) and $P(\underline{z}_k)$ from (3.16).
If $\|P(\underline{z}_k)\| \geq \alpha(C \bullet X_k - \underline{z}_k)/(n+r)$ then
(a)Find β^*:=$\text{argmin}_{0 \leq \beta \leq 1} \psi(X_k - \beta L_k P(\underline{z}_k) L_k^T, S_k)$,
using a line search procedure.
(b) Set $(\overline{X}_{k+1}, \overline{\mathbf{x}}_{k+1}) = (I, \mathbf{1}) - \beta^* P(\underline{z})$,
and set X_{k+1}:=$\mathcal{T}^{-1}(\overline{X}_{k+1}, \overline{\mathbf{x}}_{k+1})$.
(c) Set S_{k+1}:=S_k, and $\underline{z}_{k+1}$:=$\underline{z}_k$.
Else
(d) Find $\underline{z}^*$:=$\text{argmin}_{\underline{z} \leq \underline{z}_k} \psi(X_k, S(\underline{z}))$ by a line search.
(e) Set $S_{k+1} = S(\underline{z}^*)$.
(f) Set $X_{k+1} = X_k$, and $\underline{z}_{k+1} = \mathbf{b}^T\mathbf{y}(\underline{z}^*)$.
Set $k = k + 1$.
end.

FIG. 3.1. *A projective potential reduction algorithm.*

The following theorem now shows that by using Algorithm 3.1, one can get the duality gap to less than ϵ in a number of iterations k, which is dependent on $|\log\epsilon|$, $\sqrt{n}$, and the value of the potential function at the initial solution.

THEOREM 3.7. *Let X_0, $\mathbf{y}_0$, and S_0:=$C - \text{Mat}(\mathcal{A}^T\mathbf{y}_0)$ be given initial interior points for the primal and dual semidefinite programming problems in* (2.1). *Also let $r = \lceil \sqrt{n} \rceil$ and $q = n + r$ in the primal-dual potential function ψ, and assume that $\psi(X_0, S_0) \leq O(\sqrt{n}E)$ for some constant E. If an algorithm generates a sequence of interior primal and dual points X_j, $\mathbf{y}_j$ (and thus S_j) such that $\psi(X_j, S_j) \geq$*

$\psi(X_{j+1}, S_{j+1}) + \delta$ for some fixed number δ then, after $k = O(\sqrt{n}|\log\epsilon|)$ iterations, for primal and dual solutions X_k, $\mathbf{y}_k$, and S_k, we have

$$C \bullet X_k - \mathbf{b}^T \mathbf{y}_k < 2^E \epsilon.$$

Proof. Each iteration reduces the potential function by at least δ. Thus, if $\psi(X_0, S_0) < O(\sqrt{n}E)$ then after $O(\sqrt{n}|\log\epsilon|)$ iterations we have

$$\begin{aligned}\psi(X_k, S_k) &< \left(\sqrt{n}E - \sqrt{n}|\log\epsilon|\right) \\ &= (\sqrt{n}[\log 2^E - |\log\epsilon|] \\ &\leq \sqrt{n}|\log(2^E\epsilon)|.\end{aligned}$$

Therefore,

$$\begin{aligned}\sqrt{n}\log X_k \bullet S_k &< -n\log X_k \bullet S_k + \log\det X_k S_k + \sqrt{n}|\log(2^E\epsilon)| \\ &< -n\log n + \sqrt{n}|\log(2^E\epsilon)|.\end{aligned}$$

The last inequality comes from applying the arithmetic-geometric inequality to the eigenvalues of $X_k S_k$, which are real, as both matrices are positive definite. Therefore $\log X_k \bullet S_k < |\log(2^E\epsilon)|$. Since $X_k \bullet S_k = C \bullet X_k - \mathbf{b}^T\mathbf{y}_k$, the theorem follows. □

In other words, if we start our potential reduction algorithm at a pair $(X_0, \mathbf{y}_0)$ with $\psi(X_0, S_0) = \sqrt{n}E$, then after $O(\sqrt{n}(E+|\log\epsilon|))$ iterations we will have a solution with duality gap less than ϵ. Therefore, for all $\epsilon < 2^{-E}$ the term $|\log\epsilon|$ dominates E and so the number of iterations is bounded by $O(\sqrt{n}|\log\epsilon|)$. Also observe that this proof solely depends on the reduction of the potential function ψ. We must guarantee a reduction of at least δ in each iteration, but larger reductions may speed up the algorithm without affecting its worst case complexity. Therefore, in steps 4(a) and 4(d) of the algorithm in Fig. 3.1, we allow a line search to find a step length β^* and $\underline{z}^*$, which maximizes the reduction in the potential function.

3.3. Feasibility, boundedness, and polynomial-time computability. To complete our analysis we must study feasibility of the SDP problem and bounds on the norms of the optimal primal and dual solutions. The situation is somewhat different from LP. First, let us assume that all entries in the primal and dual problems (2.1) are integers. In contrast with LP, the optimal solution of (2.1) is not necessarily a rational number. Therefore we need to specify an error tolerance, ϵ, and ask for a pair of primal and dual solutions X and S such that the duality gap $X \bullet S \leq \epsilon$.[4] If ϵ is also a rational number, define L, the *size* of the SDP problem, as the number of bits in the binary representation of ϵ and entries of C, $\mathcal{A}$, and $\mathbf{b}$; see [32] for a complete definition of "size" of a problem. One might expect that the interior point method developed in the previous sections leads to an algorithm that runs in time polynomial in m, n, and L. However, this is not generally true as the solution itself may be exponentially large. To see this, consider the optimization problem

$$\min\{x_n : \ x_1 \geq 2, \text{ and } x_i \geq x_{i-1}^2 \text{ for } i = 2, \ldots, n\}. \tag{3.28}$$

Clearly, $x_i = 2^{2^{i-1}}$ for $i = 1, \ldots, n$ is the solution of this problem, which requires exponential number of bits. Now (3.28) can be written as the following semidefinite

[4] Since X, S, and $\mathbf{y}$ are solutions of the algebraic system of equations: $XS = 0, \mathcal{A}\mathbf{vec}X = \mathbf{b}$, and $\mathcal{A}^T\mathbf{y} + S = C$, there are algebraic solutions among all optimal solutions of an SDP problem with integral input.

program:

$$\begin{array}{ll} \min & x_n \\ \text{s.t.} & x_1 \geq 2 \\ & \begin{pmatrix} x_i & x_{i-1} \\ x_{i-1} & 1 \end{pmatrix} \succeq 0 \quad \text{for } i = 2, \ldots, n. \end{array}$$

This SDP problem can be easily turned into a standard form SDP whose input size (taking $\epsilon = 1$, say) is polynomial in n and whose output requires exponential number of bits. Therefore no algorithm can solve it in polynomial time.[5]

In many cases, including all of the combinatorial optimization problems described below, one may be able to put an a priori bound on the norms of the optimal solutions. For instance, we may be able to prove that the feasible sets of both the primal and dual solutions are inside balls of radius, say, R (a rational number) centered at the origin. In such cases we can show that the interior point algorithm developed earlier can produce in polynomial time (in $|\log \epsilon|$, L and $\ln R$) primal and dual solutions whose duality gap is smaller than ϵ. Notice that in the ellipsoid method such an a priori bound is assumed by requiring that the initial ellipsoid be a ball of radius R centered at the origin. Let L' be the number of bits in the binary expansion of some R. Then, similar to LP, one can always transform the pair of primal and dual problems (2.1) to another pair for which initial interior feasible points are readily available. We extend the construction suggested by Kojima, Mizuno, and Yoshise in [38] which, in turn, is based on Megiddo [43].

Consider the following pair of primal and dual problems:

$$\begin{array}{ll} \min & C \bullet X + Mx_1 \\ \text{s.t.} & \mathcal{A}\mathbf{vec}(X) + [\mathbf{b} - \mathcal{A}\mathbf{vec}(X_0)]x_1 = \mathbf{b}, \\ & [\mathrm{Mat}(\mathcal{A}^T\mathbf{y}_0) + S_0 - C] \bullet X + x_2 = N, \\ & X \succeq 0, \\ & x_1, x_2 \geq 0, \end{array} \tag{3.29}$$

and

$$\begin{array}{ll} \max & \mathbf{b}^T\mathbf{y} - Ny_1 \\ \text{s.t.} & \mathrm{Mat}(\mathcal{A}^T\mathbf{y}) + S + [C - \mathrm{Mat}(\mathcal{A}^T\mathbf{y}_0) - S_0]y_1 = C, \\ & [\mathbf{b} - \mathcal{A}\mathbf{vec}(X_0)]^T\mathbf{y} + y_2 = M, \\ & S \succeq 0, \\ & y_1, y_2 \geq 0, \end{array} \tag{3.30}$$

where X_0 and S_0 are arbitrary positive definite $n \times n$ matrices, $\mathbf{y}_0$ an arbitrary m-vector, and M and N are large enough positive numbers to ensure that $y_2 \geq 0$ and $x_2 \geq 0$. Clearly, $X{:=}X_0$, $x_1{:=}1$, and $x_2{:=}N - (\mathrm{Mat}(\mathcal{A}^T\mathbf{y}_0) + S_0 - C) \bullet X_0$ are interior feasible for the primal (3.29) (with large enough N); and $S{:=}S_0$, $\mathbf{y}{:=}\mathbf{y}_0$, $y_1{:=}1$, and $y_2{:=}M - (\mathbf{b} - \mathcal{A}\mathbf{vec}(X_0))^T\mathbf{y}_0$ are interior feasible for the dual problem in (3.30) (for large enough M). By choosing $X_0 = S_0 = I$, $x_1 = y_1 = 1$, $\mathbf{y}_0 = \mathbf{0}$, it suffices to choose M and N such that

$$N > \max\left(n - \sum_i C_{ii},\ \sum_i X^*_{ii} - C \bullet X^*\right),$$

[5] I am indebted to Motakuri Ramana for bringing to my attention an error in [1],[2] where I had claimed that the norm of the solution to any SDP problem is bounded by 2^L. Ramana essentially provided this counterexample.

TABLE 3.1
Correspondence between LP *and* SDP.

LP	SDP
unknown vector: $\mathbf{x}$	unknown symmetric matrix: X
inequality constraints: $\geq$	Löwner constraints: $\succeq$
dual variable: $\mathbf{y}$	dual variable: $\mathbf{y}$
dual slack vector: $\mathbf{s}$	dual slack symmetric matrix: S
$\mathbf{1}$	I
linear scaling: $\mathbf{x} \rightarrow (x_i/(\mathbf{x}_0)_i)_{i=1}^n = [\mathrm{Diag}(\mathbf{x}_0)]^{-1}\mathbf{x}$	linear scaling: $X \rightarrow L_0^{-1} X L_0^{-T} = \mathrm{Mat}[(L_0^{-1} \otimes L_0^{-1})\mathbf{vec}(X)]$
projective scaling: $\mathbf{x} \rightarrow \frac{c_1[\mathrm{Diag}(\mathbf{x}_0)]^{-1}\mathbf{x}}{c_2+\mathbf{1}^T[\mathrm{Diag}(\mathbf{x}_0)]^{-1}\mathbf{x}}$	projective scaling: $X \rightarrow \frac{c_1 L_0^{-1} X L_0^{-T}}{c_2+\mathrm{trace}\ L_0^{-1} X L_0^{-T}}$
barrier function: $\sum \ln x_i$	barrier function: $\ln \det X$
norms: $\|\mathbf{x}\|$ $\|\mathbf{x}\|_\infty$ $\|\mathbf{x}\|_p$	norms: $\|X\|$ $\|X\|_2$ $(\sum \lvert\lambda_i(X)\rvert^p)^{1/p}$

$$M > \max\left(0,\ \mathbf{b}^T\mathbf{y}^* - \sum_i y_i^* \mathrm{trace}\ (A_i)\right).$$

For instance we may set $N = M = 2^{L+L'}$. It is easy to see that if the optimal value of x_1 is not zero, then the original primal is infeasible (the proof is exactly like the one given by Kojima et al. in [38]). Similarly, if the optimal value of y_1 is not zero, then the original dual is infeasible. Otherwise, the optimal X^* and $\mathbf{y}^*$ are also optimal for the original primal and dual problems, respectively. Furthermore, It is easily verified that the value of the primal-dual potential function ψ at the initial point is bounded by $O(\sqrt{n}(L+L'))$. So, for the general SDP problem, any algorithm that reduces the primal-dual potential function ψ by a constant amount may find, in $O(\sqrt{n}\max(L, L', |\log\epsilon|))$ iterations, a pair of primal and dual feasible solutions whose duality gap is less than ϵ; if $\epsilon < 2^{-L-L'}$, then the number of iterations is bounded by $O(\sqrt{n}|\log\epsilon|)$.

3.4. A correspondence between proofs in linear and semidefinite programming. The remarkable similarity between the algorithm presented here and Ye's LP algorithm in [66] suggests that other LP interior point methods may also be extended to SDP problems. Proofs of convergence and polynomial time complexity may be extended as well. The correspondence is summarized in Table 3.1. Given an interior point algorithm for LP we may construct, in a mechanical way, an algorithm for the SDP problem by replacing any references to the entries under the LP column with the corresponding entry under the SDP column. Proofs of convergence or polynomial time complexity may also be extended mechanically in the same manner. We have already verified this claim on the approaches of Gonzaga [28] and Ye [67] (see [2]). Extension to primal-dual methods such as Monteiro and Adler [44] is more challenging. This table itself may be summarized by the following rule: In any LP algorithm, replace any implicit or explicit reference to x_i (or s_i) by a reference to $\lambda_i(X)$ (or $\lambda_i(S)$). Furthermore, in any scaling, replace affine or projective transformations by corresponding *symmetry preserving* transformation on matrices. Notice that these same rules were implicitly used to derive various duality and complementary slackness theorems for SDP from the corresponding theorems for LP.

Similar techniques may be applied to more general problems. For instance, one can define a semidefinite analog of convex quadratic programming or, more generally, a semidefinite analog of the linear complementarity problem. Also, one can study a semidefinite analog of linear fractional programs. For the linear version of all these problems, interior point methods have been developed (see [45], [37], and [4], for example) and one can apply the conversion rules mentioned above to obtain interior point methods for their semidefinite variants. Details are omitted here.

3.5. Differences between SDP and LP interior point algorithms. Thus far, we have emphasized the similarity of linear and semidefinite interior point methods. There are however, important distinctions and some favorable circumstances in LP do not extend to SDP. We have already seen the differences between LP and SDP when we studied irrationality and a priori bounds on the number of bits in the optimal solutions. We list other distinctions that must be studied carefully before a serious practical implementation of interior point SDP algorithms is attempted.

1. In the absence of degeneracy one can predict that precisely m entries of the optimal vector $\mathbf{x}^*$ are nonzero in the standard linear program with coefficient matrix $A \in \Re^{m \times n}$. Recall that in each iteration of a primal interior point algorithm, the main computational effort is in obtaining $(A \operatorname{Diag}(\mathbf{x})^2 A^T)^{-1}\mathbf{v}$, where $\mathbf{v}$ is some vector. Therefore, if A is of rank m and reasonably well conditioned, this computation is fairly straightforward and typically no numerical difficulties should arise. In SDP however, even if we assume strict complementarity, (i.e., $\operatorname{Rank}(X^*) + \operatorname{Rank}(S^*) = n$,) we still do not know what $\operatorname{Rank}(X^*)$ is going to be before solving the SDP problem. Furthermore, let $\operatorname{Rank}(X^*) = r$. Since the main computational work in SDP interior point methods is computing $(\mathcal{A}(X \otimes X)\mathcal{A}^T)^{-1}\mathbf{v}$, even if $\mathcal{A}$ is full rank and reasonably well conditioned, $\mathcal{A}(X \otimes X)\mathcal{A}^T$ may converge to a singular matrix unless $m \leq r^2$, which is not guaranteed. The same issue arises if we use dual or primal-dual interior point algorithms.

2. The main reason that interior point methods in LP are practically competitive —aside from the small number of iterations—is that if the matrix AA^T is sparse, so is ADA^T for any diagonal matrix D; in fact, ADA^T and AA^T have precisely the same nonzero structure. Therefore, once a good order of elimination is obtained for AA^T, the same order works for all subsequent iterations of the interior point algorithm. This is is not the case for SDP. In general even if $\mathcal{A}\mathcal{A}^T$ is sparse, the matrix $\mathcal{A}(X \otimes X)\mathcal{A}^T$ may not be sparse at all. It is not clear how factorization of $\mathcal{A}(X_k \otimes X_k)\mathcal{A}^T$ could be of any use in factoring $\mathcal{A}(X_{k+1} \otimes X_{k+1})\mathcal{A}^T$.

3. Karmarkar in [36] gives a nice amortized method for updating factors of ADA^T. He develops a technique where $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$ differ only in j_k entries where $\sum j_k$ over all iterations is bounded by $O(\sqrt{n})$. From this observation he manages to reduce the overall number of operations by a factor of $\sqrt{n}$. It is not clear how one extends Karmarkar's amortized scheme to SDP interior point algorithms. (See [49] for some progress in this direction.)

4. Eigenvalues as semidefinite programs. In many cases semidefinite programs arise in the form of minimizing or maximizing an appropriate linear combination of eigenvalues of a symmetric matrix subject to linear constraints on the matrix. In this section, we study problems of this form and show that, under proper assumptions, they are indeed special case semidefinite programs. We give primal and dual characterization of each problem and examine the complementary slackness theorem as specialized to that problem.

4.1. Minimizing sum of the first few eigenvalues. First we consider minimizing sum of the first k eigenvalues of a symmetric matrix subject to linear constraints on the matrix. We consider two variations, namely,

$$\min\{\lambda_1(X) + \cdots + \lambda_k(X) : \mathcal{A}\mathbf{vec}X = \mathbf{b}\} \tag{4.1}$$

and

$$\min \sum_{i=1}^{k} \lambda_i(A(\mathbf{x})) \text{ where } A(\mathbf{x}) = A_0 + \sum_{i=1}^{m} x_i A_i. \tag{4.2}$$

To show that these problems are indeed semidefinite programs, we use the following elegant characterization by Overton and Womersley [53], [54].

THEOREM 4.1. *For the sum of the first k eigenvalues of a symmetric matrix A the following* SDP *characterization holds:*

$$\begin{aligned} \lambda_1(A) + \cdots + \lambda_k(A) = \max \quad & A \bullet U \\ \text{s.t.} \quad & \text{trace } U = k, \\ & 0 \preceq U \preceq I. \end{aligned} \tag{4.3}$$

It is worth mentioning that this result is based on a beautiful convex hull characterization that was known at least as early as 1971 (see [22]) but unfortunately has remained somewhat obscure. Here is the statement of this result.

LEMMA 4.2. *Let*

$$S_1 := \{YY^T : Y \in \Re^{n\times k},\ Y^TY = I\}$$

and

$$S_2 := \{W : W = W^T,\ \text{trace } W = k,\ 0 \preceq W \preceq I\}.$$

Then

$$\text{conv } S_1 = S_2,$$

and S_1 is exactly the set of extreme points of S_2.

For a historical account of this result, its connection to the well-known, but computationally less useful theorem of K. Fan, and interesting connections to the theorem of Birkhoff and Von Neumann concerning the convex hull of doubly stochastic matrices, refer to Overton and Womersley [54].

To express (4.1) as a semidefinite program we first derive a dual characterization of sum of the first k eigenvalues of A, by finding a dual version of Theorem 4.1. Let us see how applying the rules of Table 2.1 to (4.3) can aid us in finding such dual characterization. The constraint $U \preceq I$ gives rise to dual variable V, which by the third line of Table 2.1, must satisfy $V \succeq 0$. The variable $U \succeq 0$, by the eighth line of Table 2.1, gives rise to the constraint $zI + V \succeq A$. Thus we have the following result.

THEOREM 4.3. *For the sum of the first k eigenvalues of a symmetric matrix A the following* SDP *characterization holds:*

$$\begin{aligned} \lambda_1(A) + \cdots + \lambda_k(A) = \min \quad & kz + \text{trace } V \\ \text{s.t.} \quad & zI + V \succeq A, \\ & V \succeq 0. \end{aligned} \tag{4.4}$$

Now, it is easy to incorporate the equality constraints into (4.4) by replacing A with X. So the optimization problem (4.1) is equivalent to

$$\begin{array}{ll} \min & kz + \text{trace } V \\ \text{s.t.} & \mathcal{A}\mathbf{vec}X = \mathbf{b}, \\ & zI + V - X \succeq 0, \\ & V \succeq 0, \end{array} \tag{4.5}$$

and taking the dual again we have the following dual characterization:

$$\begin{array}{ll} \max & \mathbf{b}^T\mathbf{y} \\ \text{s.t.} & U = \text{Mat}(\mathcal{A}^T\mathbf{y}), \\ & \text{trace } U = k, \\ & 0 \preceq U \preceq I. \end{array} \tag{4.6}$$

The complementary slackness result for primal feasible z^*, X^*, and V^*, and dual feasible U^* states that these are optimal if and only if

$$(z^*I + V^* - X^*)U^* = (I - U^*)V^* = 0.$$

Similarly (4.2) may be expressed by the following primal and dual pair:

$$\begin{array}{ll} \min & kz + \text{trace } V \\ \text{s.t.} & zI + V - \sum x_i A_i \succeq A_0, \\ & V \succeq 0, \end{array} \qquad \begin{array}{ll} \max & A_0 \bullet Y \\ \text{s.t.} & \text{trace } Y = k, \\ & A_i \bullet Y = 0 \quad \text{for } i = 1, \ldots, m, \\ & 0 \preceq Y \preceq I. \end{array} \tag{4.7}$$

When $k = 1$, these characterizations become simpler, because in that case the constraint $Y \preceq I$ (and thus variable V) are redundant. Therefore, the problem

$$\min\{\lambda_1(X) : \mathcal{A}\mathbf{vec}X = \mathbf{b}\}$$

may be expressed as the solution of the primal and dual SDP pair

$$\begin{array}{ll} \min & z \\ \text{s.t.} & zI - X \succeq 0, \\ & \mathcal{A}\mathbf{vec}X = \mathbf{b}, \end{array} \qquad \begin{array}{ll} \max & \mathbf{b}^T\mathbf{y} \\ \text{s.t.} & \text{trace } \text{Mat}(\mathcal{A}^T\mathbf{y}) = 1, \\ & \text{Mat}(\mathcal{A}^T\mathbf{y}) \succeq 0, \end{array} \tag{4.8}$$

and the complementary slackness theorem indicates that for X^* and $\mathbf{y}^*$ to be primal and dual optimum solution for (4.8), in addition to being primal and dual feasible they must satisfy

$$\text{Mat}(\mathcal{A}^T\mathbf{y}^*)(\lambda_1(X^*)I - X^*) = 0.$$

4.2. Minimizing weighted sums of eigenvalues. In this section we consider the weighted sum of eigenvalues of a matrix. Let $m_1 \geq m_2 \geq \cdots \geq m_k > m_{k+1} = 0$ be a set of fixed real numbers. We are interested in the following problem:

$$\min\{m_1\lambda_1(X) + \cdots + m_k\lambda_k(X) : \mathcal{A}\mathbf{vec}X = \mathbf{b}\}. \tag{4.9}$$

Note that without the condition $m_1 \geq m_2 \geq \cdots \geq m_k > 0$, (4.9) is not necessarily a convex program. To formulate this problem as a semidefinite program, we use a

technique originally employed by Donath and Hoffman in [18]. They rewrote the sum as follows:

$$\begin{aligned} m_1\lambda_1(A) + m_2\lambda_2(A) + \cdots + m_k\lambda_k(A) = {} & (m_1 - m_2)\lambda_1(A) \\ & + (m_2 - m_3)[\lambda_1(A) + \lambda_2(A)] + \cdots \\ & + (m_{k-1} - m_k)[\lambda_1(A) + \cdots + \lambda_{k-1}(A)] \\ & + m_k[\lambda_1(A) + \cdots + \lambda_k(A)] \end{aligned} \tag{4.10}$$

and observed that the right-hand side of (4.10) is a nonnegative combination of convex functions and, therefore, is itself convex. This formulation also allows us to write (4.9) as an SDP. For each of the partial sums of eigenvalues in (4.10) we may use the SDP formulation of the last subsection and obtain the primal

$$\begin{array}{ll} \min & \sum_{i=1}^k iz_i + \sum_{i=1}^k \operatorname{trace} V_i \\ \text{s.t.} & z_i I + V_i - (m_i - m_{i+1})X \succeq 0 \quad \text{for } i = 1, \ldots, k, \\ & \mathcal{A}\mathbf{vec}X = \mathbf{b}, \\ & V_i \succeq 0 \quad \text{for } i = 1, \ldots, k \end{array} \tag{4.11}$$

and the dual

$$\begin{array}{ll} \max & \mathbf{b}^T\mathbf{y} \\ \text{s.t.} & \mathcal{A}^T\mathbf{y} - \sum_{i=1}^k (m_i - m_{i+1})U_i = 0, \\ & \operatorname{trace} U_i = i \quad \text{for } i = 1, \ldots, k, \\ & 0 \preceq U_i \preceq I \quad \text{for } i = 1, \ldots, k \end{array} \tag{4.12}$$

equivalents of (4.9).

The complementary slackness condition for feasible X^*, z_i^*, V_i^*, $\mathbf{y}^*$, and U_i^* for $i = 1, \ldots, k$ to be optimal may be stated as

$$(z_i^* I + V_i - (m_i - m_{i+1})X^*)U_i^* = (I - U_i^*)V_i^* = 0 \quad \text{for } i = 1, \ldots, k.$$

Notice that the primal and dual characterizations (4.11) and (4.12) contain $2k$ semidefinite constraints each involving $n \times n$ matrices, and therefore, the interior point methods discussed earlier require $O(\sqrt{kn})$ iterations for each new significant digit of accuracy. It will be interesting to improve this complexity to $O(\sqrt{n})$.

4.3. Minimizing sums of absolute-valuewise largest eigenvalues. The results of the two preceding subsections may be extended to the sum of the k *absolute-valuewise* largest eigenvalues as well. Overton and Womersley derived the max characterization similar to (4.3); applying duality to their result we obtain the following theorem.

THEOREM 4.4. *For a symmetric matrix A the sum $|\lambda^1(A)| + \cdots + |\lambda^k(A)|$ is equal to the optimal solution of the pair of primal and dual semidefinite programs:*

$$\begin{array}{lll} \max & A \bullet Y - A \bullet W, & \\ \text{s.t.} & \operatorname{trace}(Y + W) = k, & \min \quad kz + \operatorname{trace} V + \operatorname{trace} U \\ & 0 \preceq Y \preceq I, & \text{s.t.} \quad zI + V - A \succeq 0, \\ & 0 \preceq W \preceq I, & \qquad zI + U + A \succeq 0, \\ & & \qquad U \succeq 0, \\ & & \qquad V \succeq 0. \end{array} \tag{4.13}$$

(Recall that $\lambda^i(X)$ is the ith largest eigenvalue of X in the absolute-value sense.)

Now to solve the optimization problem

$$\min\{|\lambda^1(X)| + \cdots + |\lambda^k(X)| : \mathcal{A}\mathbf{vec}X = \mathbf{b}\}, \tag{4.14}$$

we may simply add the equality constraints to the min formulation in (4.13). Then taking its dual we get the following pair of primal and dual semidefinite programs:

$$
\begin{array}{lll}
\min & kz + \text{trace } V + \text{trace } U & \\
\text{s.t.} & \mathcal{A}\mathbf{vec}X = \mathbf{b}, & \\
& zI + V - X \succeq 0, & \\
& zI + U + X \succeq 0, & \\
& U \succeq 0, & \\
& V \succeq 0, &
\end{array}
\qquad
\begin{array}{lll}
\max & \mathbf{b}^T\mathbf{y} & \\
\text{s.t.} & \mathcal{A}^T\mathbf{y} = Y - W, & \\
& \text{trace } (Y + W) = k, & \\
& 0 \preceq Y \preceq I, & \\
& 0 \preceq W \preceq I. &
\end{array}
\tag{4.15}
$$

The complementary slackness theorem indicates that primal feasible z^*, V^*, and U^*, and dual feasible Y^*, and W^* are optimal if and only if

$$(z^*I + V^* - X^*)Y^* = (z^*I + U^* + X^*)W^* = (I - Y^*)U^* = (I - W^*)V^* = 0.$$

Again these results may be generalized to the weighted sums of absolute-value-wise largest eigenvalues. In other words, the problem

$$\min\{m_1|\lambda^1(X)| + \cdots + m_k|\lambda^k(X)| : \mathcal{A}\mathbf{vec}X = \mathbf{b}\} \tag{4.16}$$

may be expressed by a primal and dual pair of semidefinite programs. First, let us ignore the equality constraints $\mathcal{A}\mathbf{vec}X = \mathbf{b}$, and assume that X is a fixed matrix A. Then, we have the following result.

THEOREM 4.5. *The sum $m_1|\lambda^1(A)| + \cdots + m_k|\lambda^k(A)|$, where A is a symmetric matrix equals the optimal solution of the primal program*

$$
\begin{array}{lll}
\min & \sum_{i=1}^k iz_i + \sum_{i=1}^k \text{trace } (U_i + V_i) & \\
\text{s.t.} & z_iI + U_i - (m_i - m_{i+1})A \succeq 0 & \text{for } i = 1, \ldots, k, \\
& z_iI + V_i + (m_i - m_{i+1})A \succeq 0 & \text{for } i = 1, \ldots, k, \\
& U_i \succeq 0 & \text{for } i = 1, \ldots, k, \\
& V_i \succeq 0 & \text{for } i = 1, \ldots, k
\end{array}
\tag{4.17}
$$

and the dual program

$$
\begin{array}{lll}
\max & \sum_{i=1}^k (m_i - m_{i+1})(A \bullet Y_i - A \bullet W_i) & \\
\text{s.t.} & \text{trace } (Y_i + W_i) = i & \text{for } i = 1, \ldots, k, \\
& 0 \preceq Y_i \preceq I & \text{for } i = 1, \ldots, k, \\
& 0 \preceq W_i \preceq I & \text{for } i = 1, \ldots, k.
\end{array}
\tag{4.18}
$$

Now we may replace A by X and impose the equality constraints on the min characterization in (4.17). After taking the dual we have the following pair of primal and dual SDP equivalents of (4.16):

$$
\begin{array}{lll}
\min & \sum_{i=1}^k iz_i + \sum_{i=1}^k \text{trace } (U_i + V_i) & \\
\text{s.t.} & \mathcal{A}\mathbf{vec}X = \mathbf{b}, & \\
& z_iI + U_i - (m_i - m_{i+1})X \succeq 0 & \text{for } i = 1, \ldots, k, \\
& z_iI + V_i + (m_i - m_{i+1})X \succeq 0 & \text{for } i = 1, \ldots, k, \\
& U_i \succeq 0 & \text{for } i = 1, \ldots, k, \\
& V_i \succeq 0 & \text{for } i = 1, \ldots, k
\end{array}
\tag{4.19}
$$

and

$$\begin{array}{ll} \max & \mathbf{b}^T\mathbf{y} \\ \text{s.t.} & \mathcal{A}^T\mathbf{y} = \sum_{i=1}^{k}(m_i - m_{i+1})(Y_i - W_i), \\ & \text{trace}\,(Y_i + W_i) = i \quad \text{for } i = 1, \ldots, k, \\ & 0 \preceq Y_i \preceq I \quad \text{for } i = 1, \ldots, k, \\ & 0 \preceq W_i \preceq I \quad \text{for } i = 1, \ldots, k. \end{array} \tag{4.20}$$

Finally, the complementary slackness theorem for problem (4.16) states that primal (4.19) feasible z_i^*, V_i^*, and U_i^*, and dual (4.20) feasible Y_i^*, and W_i^*, for $i = 1, \ldots, k$ are optimal if and only if

$$\begin{aligned}(z_i^* I + V_i^* - (m_i - m_{i+1})X_i^*)Y_i^* &= (z_i^* I + U_i^* + (m_i - m_{i+1})X_i^*)W_i^* \\ &= (I - Y_i^*)U_i^* = (I - W_i^*)V_i^* = 0 \quad \text{for } i = 1, \ldots, k.\end{aligned}$$

The characterization (4.3), and the max part of (4.13) were given in Overton and Womersley [53]. Also, Fletcher in [23] derives a closely related result to (4.3) but the result was incorrect (Fletcher had $0 \preceq S$ rather than $0 \preceq S \preceq I$.) The min characterizations as well as the primal and dual formulation of the variants with equality constraints, we believe are new.

In a similar manner, primal and dual SDP formulations can be derived for maximizing (weighted) sums of the smallest eigenvalues of symmetric matrices or (weighted) sums of the largest singular values of arbitrary matrices; we omit these straightforward formulations here (see [62] for the study of singular values). However, maximizing the last few smallest eigenvalues of a symmetric matrix *absolute-valuewise*, or sum of the last few smallest singular values of an arbitrary matrix, cannot be formulated as SDP because these problems are not convex programs.

5. Applications in combinatorial optimization. The SDP problem studied in the previous sections has applications in combinatorial optimization, especially in graph theory. The connection usually is the spectral properties of graphs. Semidefinite programs may arise in two different roles. Their more common role is to provide an approximation—an upper or lower bound—on an NP-hard combinatorial optimization problem. In such role one hopes that the SDP bound gives rise to much sharper bounds than the more common LP bounds. Remarkably SDP relaxations have been shown to give rise to approximation algorithms whose guaranteed performance is superior to any known combinatorial or LP approximation technique; see for instance [25]. The second role is to give *exact* characterization to some special cases of combinatorial optimization problems. An example of such application is the SDP formulation of maximum clique and maximum stable set problem in perfect graphs.

In the following sections we first examine a general approach of Lovász and Schrijver which applies semidefinite programming to zero-one integer programming problems. Then we study other applications such as the maximum stable set, the maximum induced k-partite subgraph, and graph partitioning (in particular, graph bisection) and the maximum cut problems.

5.1. Nonlinear relaxations of 0-1 programming. Consider the integer programming problem

$$\max\{\bar{\mathbf{c}}^T\bar{\mathbf{x}} : \bar{A}\bar{\mathbf{x}} \geq \mathbf{b} \text{ and } \bar{x}_i \in \{0, 1\}\}. \tag{5.1}$$

The LP relaxation of (5.1) results from replacing $\bar{x}_i \in \{0, 1\}$ with $0 \leq \bar{x}_i \leq 1$. This relaxation serves as a first approximation of the solution of (5.1). In general, this first

approximation may be nonintegral and far from the actual solutions. Most effective methods of integer programming consist of adding new "cutting planes" to the LP relaxation and then using some branch and bound technique to the resulting problem. It seems, however, that little work has been done in generating "nonlinear" but convex cuts in the feasible region of the LP relaxation. Generally such cuts may produce far better approximations than linear cuts. An ingenious approach for creating a class of nonlinear cuts has been proposed by Lovász and Schrijver in [42]. The idea is to "lift" the space from vectors in $\Re^n$ to $n \times n$ symmetric matrices.[6] In essence, they provide a convex set which contains the feasible region of (5.1) and is contained in the feasible region of its LP relaxation. Furthermore, this convex relaxation may be expressed as a projection of the feasible set of a semidefinite program, and therefore itself may be represented as an SDP. Here is a summary of the Lovász and Schrijver technique.

First, it is convenient to homogenize the integer program (5.1) by introducing a new variable x_0 as a multiple of $\mathbf{b}$ and then imposing the constraint $x_0 = 1$. After this transformation the homogenized integer programming problem and its LP relaxation can be written as

IP			**LP**		
max	$\mathbf{c}^T\mathbf{x}$		max	$\mathbf{c}^T\mathbf{x}$	
s.t.	$\mathbf{a}_i^T\mathbf{x} \geq 0$	for $i = 1, \ldots, m$,	s.t.	$\mathbf{a}_i^T\mathbf{x} \geq 0$	for $i = 1, \ldots, m$,
	$x_i \in \{0, 1\}$	for $i = 0, \ldots, n$,		$0 \leq x_i \leq x_0$	for $i = 0, \ldots, n$,
	$x_0 = 1$,			$x_0 = 1$.	

Let P be the convex cone which is the feasible region of the LP relaxation *without* the constraint $x_0 = 1$, and $\Im(P)$ its integer hull (that is, $\Im(P)$ is the convex cone generated by 0-1 vectors in **LP** with $x_0 = 1$.) First, we decompose the set of constraints into two sets (with possible overlap); then we multiply each inequality in the first set by each inequality in the second set to obtain quadratic constraints, then replace each occurrence of $x_i x_j$ by a new variable x_{ij} to get linear constraints again; finally we impose on the matrix $X = (x_{ij})$ the positive semidefinite constraint. If P_1 and P_2 are the cones defined by the first and second sets of constraints, then $P = P_1 \cap P_2$, and the space of matrices just defined is denoted by $M_+(P_1, P_2)$. More formally, let J_1 and J_2 be two subsets that cover the index set of the inequality constraints in **LP**. Define $A_1{:=}A_{J_1}$, and $A_2{:=}A_{J_2}$, and P_i the set $\{\mathbf{x} : A_i\mathbf{x} \geq \mathbf{0}\}$ for $i = 1, 2$. We require that constraints $0 \leq x_i \leq x_0$ be in both subsets. Then

$$M_+(P_1, P_2){:=}\{X \in \Re^{\frac{n \times n}{2}} : X \succeq 0,\ X\mathbf{e}_0 = \mathbf{diag}(X), \text{ and } (A_1 \otimes A_2)\mathbf{vec}(X) \geq \mathbf{0}\},$$

where $\mathbf{e}_0 = (1, 0, \ldots, 0)^T$. Also, let $N_+(P_1, P_2)$ be the set of n-vectors made up of diagonals of matrices in $M_+(P_1, P_2)$, that is,

$$N_+(P_1, P_2){:=}\{\mathbf{diag}(X) : X \in M_+(P_1, P_2)\}.$$

The main result of Lovász and Schrijver, for the purposes of our discussion, is that

$$\Im(P) \subseteq N_+(P_1, P_2) \subseteq P.$$

It is clear that optimizing a linear function over $N_+(P_1, P_2)$ is an SDP problem, and interior point techniques may be applied (as long as P is given by an explicit

[6] The presentation here is more restrictive than that given in [42]. Lovász and Schrijver consider optimization problems over a cone $\mathcal{K}$ endowed with a separation oracle and derive nonliner cuts for the subcone generated by 0-1 vectors in $\mathcal{K}$.

system of inequalities.) The process just described may be quite powerful in certain combinatorial optimization problems. For instance in a general branch and bound algorithm, one may use interior point algorithms to solve the optimization problem

$$\max\{\mathbf{c}^T\mathbf{x} : \mathbf{x} \in N_+(P_1, P_2)\}.$$

The solution then may be used as a bound and the resulting $\mathbf{x}$ necessarily satisfies $\mathbf{0} \leq \mathbf{x} \leq \mathbf{1}$. Now if for some coordinate i we have $0 < x_i < 1$, then we branch by solving the two SDP subproblems with additional constraints, respectively, $x_i = 0$ and $x_i = 1$. From a practical point of view such subproblems are all polynomial time solvable by the interior point methods, though they are computationally more expensive than the classical branch and bound approach based on LP relaxations. The advantage however is that the bounds are sharper (hopefully much sharper) than the corresponding LP bounds, and therefore the total number of subproblems solved may be considerably smaller.

Lovász and Schrijver show that applying the N_+ operator to the LP relaxation of the stable set polytope of a graph $G = (V, E)$ gives bounds that are already stronger than a combination of several well-known classes of linear cuts. Recall that a stable set in a graph $G = (V, E)$ is a subset of vertices S where each pair of vertices i and j in S are nonadjacent. Let $\mathbf{w}$ be a weight vector on the vertices of G, such that w_i is the weight of vertex i. The weighted maximum stable set problem in graphs can now be formulated as the following 0-1 program:

$$\begin{array}{lll} \max & \mathbf{w}^T\mathbf{x} & \\ \text{s.t.} & x_i + x_j \leq 1 & \text{for all } \{i,j\} \in E, \\ & x_i \in \{0,1\} & \text{for all } i \in V. \end{array} \tag{5.2}$$

Now we homogenize (5.2) by adding a new variable x_0, and applying the N_+ operator to a decomposition of (5.2), where P_1 is given by the entire feasible set of (5.2) and P_2 is induced by $0 \leq x_i \leq x_0$. In other words

$$P_1{:=}P = \{\mathbf{x} :\ x_i + x_j \leq x_0 \text{ for all } i,j \in E, \text{ and } 0 \leq x_i \leq x_0 \text{ for all } i \in V\}$$

and

$$P_2{:=}\{\mathbf{x} :\ x_0 - x_i \geq 0, \text{ and } x_i \geq 0\}.$$

Let the resulting set be $N_+(\text{STAB } G)$. Optimization over this set is a semidefinite program and can be done in polynomial time using interior point methods (Lovász and Schrijver use the ellipsoid method to establish polynomiality). Furthermore, it is clear that

$$\text{STAB } G \subseteq N_+(\text{STAB } G) \subseteq E\text{–STAB } G,$$

where STAB G is the convex hull of all 0-1 vectors that characterize some stable set of G, and E–STAB G is the polytope associated with the LP relaxation of (5.2) (that is the polytope obtained by replacing constraints $x_i \in \{0,1\}$ by $0 \leq x_i \leq 1$.) The set $N_+(\text{STAB } G)$ is convex, but generally nonpolyhedral. However, Lovász and Schrijver show that the set of points in $N_+(\text{STAB } G)$ already satisfy the following classes of well-known valid inequalities for STAB G.

1. *Clique constraints.* Let K be a clique in G, that is, a subset of vertices every pair of which is adjacent. Let S be a stable set in G. Then clearly $|S \cap K| \leq 1$. This observation implies that for all cliques in G the inequality

$$\mathbf{1}_K^T\mathbf{x} \leq 1 \tag{5.3}$$

(where $\mathbf{1}_K$ is the characteristic vector of clique K) is valid for STAB G. Define

$$Q-\text{STAB } G := \{\mathbf{x}: \ x_i \geq 0 \text{ and } \mathbf{1}_K^T \mathbf{x} \leq 1 \text{ for each clique } K\}.$$

2. *Odd hole constraints.* For every cycle (hole) C with $2k+1$ edges and every stable set S we know that $|C \cap S| \leq k$. Thus, for all odd cycles C in G, the constraint

$$\mathbf{1}_C^T \mathbf{x} \leq k \tag{5.4}$$

is valid for STAB G. Define

$$C-\text{STAB } G := \{\mathbf{x}: \ x_i \geq 0 \text{ and } \mathbf{1}_C^T \mathbf{x} \leq k \text{ for each odd cycle } C\}.$$

3. *Odd antihole constraints.* Let $\bar{C}$ be a graph whose edge complement set is an odd cycle. Then the maximum stable set in G has two vertices and therefore, $|\bar{C} \cap S| \leq 2$ for all stable sets S. Therefore, for all odd antiholes $\bar{C}$ in G, every inequality

$$\mathbf{1}_{\bar{C}}^T \mathbf{x} \leq 2 \tag{5.5}$$

is valid for STAB G. Define

$$\bar{C}-\text{STAB } G := \{\mathbf{x}: \ x_i \geq 0 \text{ and } \mathbf{1}_{\bar{C}}^T \mathbf{x} \leq 2 \text{ for each odd antihole } \bar{C}\}.$$

4. *Odd wheel constraints.* Let W be a graph with $2k$ vertices such that vertices $v_1, v_2, \ldots, v_{2k-1}$ induce a cycle and vertex v_{2k} is adjacent to all other vertices. Then W is called an odd wheel. It can be shown (see [32]) that for all wheels W in G, the inequality

$$\sum_{i=1}^{2k-1} x_{v_i} + (k-1)x_{v_{2k}} \leq k-1 \tag{5.6}$$

is valid for STAB G. Define

$$W\text{-STAB } G := \left\{\mathbf{x}: \ x_i \geq 0 \text{ and } \sum_{i=1}^{2k-1} x_{v_i} + (k-1)x_{v_{2k}} \leq k-1 \text{ for each odd wheel } W\right\}.$$

It turns out that (see [42])

$$\begin{aligned}&\text{STAB } G \subseteq N_+(\text{STAB } G)\\ &\subseteq Q-\text{STAB } G \cap C-\text{STAB } G \cap \bar{C}-\text{STAB } G \cap W-\text{STAB } G \subseteq E-\text{STAB } G\end{aligned}$$

and $N_+(\text{STAB } G)$ already provides a sharper relaxation of STAB G than any of the polytopes defined above. Yet optimization over $N_+(\text{STAB } G)$ is an SDP problem, and the interior point methods developed in this paper may yield practical ways of achieving strong bounds on the maximum stable set problem.

Remark. Barriers for polytopes with exponentially many facets. A strong property of the ellipsoid method for combinatorial optimization problems is that generally one does not need to have the LP formulation of the problem *explicitly.* All that is required is existence of a separation oracle and an initial ellipsoid to start the process. For instance, for certain classes of graphs, the stable set polytope may be characterized completely by $C-\text{STAB } G$ (such graphs are called t-perfect).

Other classes may have their stable set polytope characterized by $Q-\text{STAB } G$ (perfect graphs), or by $C-\text{STAB } G \cap Q-\text{STAB } G$ (h-perfect graphs), or generally any combination of the polytopes mentioned in items 1–4 above. The stable set polytopes of such graphs have generally exponentially many facets. However, in [32], [42] it is shown that one can construct polynomial time computable separation oracles for these polytopes and thus find the maximum stable set for the corresponding graphs in polynomial time.

It is common belief that in contrast to the ellipsoid method, interior point methods require explicit knowledge of the facets of the polytope on which we wish to optimize; see, for instance, [32] and the quotation from [27] in the introduction. However, we can use polynomial time interior point methods to optimize over STAB G in the special cases mentioned above, *even though* the number of facets in such polytopes may be exponentially large. In fact, the ground breaking work of Nesterov and Nemirovskii implies that, at least in principle, a listing of all inequality constraints in the LP formulation is not necessary. Instead of a separation oracle as is required in the ellipsoid method, one needs a polynomial time computable *barrier oracle* with a polynomially bounded *self-concordance* parameter. For instance, as indicated, we can optimize over $N_+(\text{STAB } G)$ in polynomial time and $N_+(\text{STAB } G) = \text{STAB } G$ for the classes of graphs mentioned above. The general results of Nesterov and Nemirovskii imply that one can directly compute a barrier function for $N_+(\text{STAB } G)$.

THEOREM 5.1. *Let* $b : \text{Int} N_+(\text{STAB } G) \to \Re$ *be the function defined by*

$$b(\mathbf{x}) := \min\{-\ln\det X : \mathbf{diag}(X) = \mathbf{x}, X \in M_+(\text{STAB } G)\}. \tag{5.7}$$

Then there is an interior point algorithm that uses $b(\mathbf{x})$ *as its barrier and finds* $\max\{\mathbf{w}^T\mathbf{x} : \mathbf{x} \in N_+(\text{STAB } G)\}$ *in* $O(\sqrt{n}\max(\|\mathbf{w}\|, |\log\epsilon|))$ *iterations and error at most* ϵ.

Proof. Nesterov and Nemirovskii prove that $\ln\det X$ is *n-selfconcordant* for the cone of positive semidefinite $n \times n$ matrices. (See [48] for definitions.) They also show that existence of an n-self-concordant barrier for a convex set generally implies that one can optimize a linear function over that set with every $O(\sqrt{n})$ iterations yielding a significant bit. Furthermore, in Proposition 1.5, [50, p. 121] they show that if a convex set $K \subseteq \Re^n$ is endowed with an n-self-concordant barrier b, and $\mathcal{A} : \Re^n \to \Re^m$ is an affine transformation mapping K on to $\mathcal{A}(K)$, then the following function is n-self-concordant for $\mathcal{A}(K)$:

$$b^+(\mathbf{y}) := \inf\{b(\mathbf{x}) : \mathbf{x} \in \mathcal{A}^{-1}(\mathbf{y}) \cap \text{Int } K\}.$$

Now the theorem follows immediately from the definition of $N_+(\text{STAB } G)$ as given in [42] with the affine transformation $\mathcal{A}$ replaced by projection of elements of $M_+(\text{STAB } G)$ onto their diagonals. □

Notice that each iteration itself requires evaluating the function $b(\mathbf{x})$, which involves another optimization problem. Nevertheless the result above shows that if a convex set K in $\Re^n$ can be represented as a projection of another convex set $K' \in \Re^N$ with $N > n$, such that K' is endowed with a polynomial time computable p-self-concordant barrier, then there is a polynomial time computable p-self-concordant barrier for K. In combinatorial optimization there are many examples of polytopes with exponentially many facets that can be represented as a projection of polytopes in higher dimensions but with fewer (polynomially many) facets. For all such polytopes one can apply interior point methods and optimize over them in polynomial time. For a thorough discussion of liftings of polyhedra associated with combinatorial optimization problems, consult [65], [42] and the references cited in them.

It is an interesting problem to look for easily computable (for instance NC-computable or at least polynomial time computable) barriers for combinatorial optimization problems whose LP formulation contains exponentially many inequalities. A concrete open problem is to find an easily computable barrier for the matching polytope with the property that a suitable interior point algorithm with such barrier requires $O(\sqrt{m})$ iterations where m is the number of edges in the graph. This problem is especially interesting because Yannakakis has shown that under certain symmetry-preserving conditions on the lift operator it is impossible to lift the matching polytope to a higher dimensional polytope with polynomially many facets [65]. Whether the matching polytope can be represented as a projection of a convex set endowed with an $O(m)$-self-concordant barrier function remains an interesting open problem.

5.2. Maximum cliques in perfect graphs. A particularly nice application of semidefinite programming is to the solution of the maximum clique problem in perfect graphs. A graph $G(V, E)$ is called perfect if for all induced subgraphs G' of G, the size of the maximum clique, $\omega(G')$, equals the size of minimum proper coloring, $\chi(G')$. (A proper coloring of vertices of a graph is an assignment of colors to each vertex such that no two adjacent vertices have the same color.) It is clear that $\omega(G) \leq \chi(G)$ for all graphs, as one needs at least $\omega(G)$ colors just to cover the vertices of the maximum clique. Several interesting properties of perfect graphs should be noted. First, the perfect graph theorem of Lovász indicates that a graph is perfect if and only if its complement is perfect [40]. This statement is equivalent to saying that for all induced subgraphs G' of G, $\alpha(G') = \rho(G')$, where $\alpha(G')$ is the size of the largest stable set in G', and $\rho(G')$ is the size of the smallest number of cliques that cover all vertices of G'. Thus, in effect, studying cliques in perfect graphs is equivalent to studying stable sets and any algorithm for one is valid for the other one (by simply applying it to the complementary graph.) As a consequence of the perfect graph theorem, one can show that equality of maximum cliques and minimum coloring extends to the weighted graphs. More precisely, let $\mathbf{w} \in N^n$ be an integral weight vector defined on the vertices of G. A proper $\mathbf{w}$-coloring of G is an assignment of colors to the vertices of G such that each vertex has at least w_i colors and for two adjacent vertices, their color sets are disjoint. $\chi(G, \mathbf{w})$ is the minimum number of colors over all proper $\mathbf{w}$-colorings of G. A maximum weighted clique in G is the clique whose sum of weights of vertices is maximum; this sum is denoted by $\omega(G, \mathbf{w})$. A graph is perfect if and only if for all weight vectors $\mathbf{w} \in N^n$, $\omega(G, \mathbf{w}) = \chi(G, \mathbf{w})$. Restating this for the complements of graphs, we have that a graph is perfect if and only if $\alpha(G, \mathbf{w}) = \rho(G, \mathbf{w})$, where, $\alpha(G, \mathbf{w})$ is the weight of the maximum weighted stable set in G, and $\rho(G, \mathbf{w})$ is the minimum number of cliques required to cover vertices of G such that each vertex i is in at least w_i cliques. These results are equivalent to the following statement; see [32].

THEOREM 5.2. *A graph $G = (V, E)$ is perfect if and only if* STAB $G = Q -$ STAB G.

Therefore, already the discussion in the preceding subsection implies that computing maximum cliques and maximum independent sets in perfect graphs can be accomplished in polynomial time by interior point methods. However, in this case one can derive a slightly stronger result.

Lovász in [41] discovered an invariant of graphs, $\theta(G, \mathbf{w})$ that has two desirable properties: first it is polynomial time computable, and second it is simultaneously an upper bound for $\omega(G, \mathbf{w})$ and a lower bound for $\chi(G, \mathbf{w})$. This invariant can be

defined by a pair of primal and dual semidefinite programs. Let

$$\mathcal{M}:=\{X \in \Re^{\frac{n\times n}{2}} : \ X_{ij} = 0 \text{ for all } i,j \in E \text{ or } i = j\}$$

and

$$\mathcal{M}^{\perp}:=\{Y \in \Re^{\frac{n\times n}{2}} : \ Y_{ij} = 0 \text{ for all } i,j \notin E\}.$$

Then the *weighted Lovász number* of G is defined by the following primal-dual SDP pair:

$$\begin{aligned}\theta(G,\mathbf{w}) &:= \min\{\lambda_1(X+W) : \ X \in \mathcal{M}\} \\ &= \max\{W \bullet Y : \ Y \in \mathcal{M}^{\perp},\ Y \succeq 0 \text{ and trace } Y = 1\},\end{aligned} \tag{5.8}$$

where $W:=\sqrt{\mathbf{w}}\sqrt{\mathbf{w}}^T$ and $\sqrt{\mathbf{w}}$ is an n-vector whose ith component is $\sqrt{w_i}$. This min-max equality is proved directly in [32], and also follows easily from the duality theory stated earlier; see (4.8).

LEMMA 5.3. *For every vertex weighted graph* $G = (V,E)$,

$$\omega(G,\mathbf{w}) \leq \theta(G,\mathbf{w}) \leq \chi(G,\mathbf{w})$$

and

$$\alpha(G,\mathbf{w}) \leq \vartheta(G,\mathbf{w}):=\theta(\bar{G},\mathbf{w}) \leq \rho(G,\mathbf{w}).$$

See [32, Chap. 9] for a thorough treatment of the Lovász number of graphs including several other characterizations and many interesting properties. We just mention here that $\theta(G,\mathbf{w})$ is a relaxation of $\max\{\mathbf{w}^{\mathbf{x}} : \mathbf{x} \in N_+(\text{STAB } G)\}$. Since in case of perfect graphs we have

$$\omega(G,\mathbf{w}) = \theta(G,\mathbf{w}) = \chi(G,\mathbf{w})$$

and

$$\alpha(G,\mathbf{w}) = \vartheta(G,\mathbf{w}) = \rho(G,\mathbf{w}),$$

we can actually compute the maximum clique and maximum stable set in polynomial time for this class of graphs. In [32] the ellipsoid method was used to establish the polynomial time computability of maximum cliques in perfect graphs. We now show that interior point methods give us a slightly stronger result than the ellipsoid method. More precisely, we show that computing maximum cliques (and maximum stable sets) in perfect graphs can be accomplished in $\tilde{O}(\sqrt{n})$ randomized parallel time using the P-RAM model of computation if $\|\mathbf{w}\|_\infty = O(n^c)$ for some constant c.[7] This is straightforward. First recall that we showed that a standard SDP problem can be solved in $O(\sqrt{n}\max(L,L',|\log\epsilon|))$ iterations, if L is the number of bits in the input SDP, L' is an a priori bound on the norm of the solution, and ϵ is the accuracy required on the size of the duality gap. In case of perfect graphs we only need to set $\epsilon = 1/3$; in fact, if z_k and Y_k are our current primal and dual estimates where there is only one integer between z_k and $W \bullet Y_k$, then we can stop and declare $\theta(G,\mathbf{w}) = \lceil z_k \rceil = \lfloor W \bullet Y_k \rfloor$. Furthermore, $L = \tilde{O}(1)$ since all coefficients in the primal-dual characterization of $\theta(G,\mathbf{w})$ in (5.8) are either zero or one or $w_i w_j$. Similarly, $L' =$

[7] $\tilde{O}(\sqrt{n})$ means $O(\sqrt{n}\log^k n)$ for some constant k.

$\tilde{O}(1)$ because the weight of the maximum clique cannot exceed $\sum w_i$. Thus computing $\theta(G, \mathbf{w})$ requires $\tilde{O}(\sqrt{n})$ iterations. Each iteration essentially involves solving a system of linear equations that is already known to be in complexity class NC, that is, it requires $\tilde{O}(1)$ time with polynomial number of processors. Therefore, computing $\theta(G, \mathbf{w})$ for polynomially bounded $\mathbf{w}$ requires $\tilde{O}(\sqrt{n})$ operations on a P-RAM model of computation.

It remains to show that computing the maximum clique itself can be accomplished in $\tilde{O}(\sqrt{n})$. We cannot use the self-reducibility process here since it may require $O(n)$ time even on a P-RAM machine. However, observe that if the maximum clique is unique then we can compute it in $\tilde{O}(\sqrt{n})$ parallel time. One could remove one vertex i of the graph and compute $\theta(G\backslash i, \mathbf{w})$ for the remaining graph. The vertex i is in the unique maximum clique if and only if $\omega(G\backslash i, \mathbf{w}) < \omega(G, \mathbf{w})$. Therefore, testing this simultaneously for all vertices we get the set of vertices in the maximum clique. In general we do not have uniqueness, but we could use the randomized perturbation scheme of Mulmuley, Vazirani, and Vazirani [46]. First recall their isolating lemma.

LEMMA 5.4. *Let $S = \{x_1, \ldots, x_n\}$ and F a family of subsets of S, that is $F = \{S_1, \ldots, S_N\}$. Furthermore, let elements of S be assigned integer weights chosen uniformly and independently at random from $[1, 2n]$. Then,*

$$\Pr[\text{There is a unique maximum weight set in } F] \geq \frac{1}{2}.$$

See [46] for the proof.

To get a maximum clique in a perfect graph we follow a procedure similar to the one adopted by Mulmuley, Vazirani, and Vazirani for constructing the minimum weighted perfect matching in graphs. The idea is to assign weights to vertices randomly so that with high probability the maximum clique with the new weights is unique, but at the same time, this clique is among the maximum cliques with the original weights.

Let $C := \sum_i w_i$. First give a weight of $2C^2 w_i$ to each vertex i so that the weight of maximum weighted cliques is at least $2C^2$ more than the next largest clique weight. Then perturb weight of each vertex i by adding integer u_i uniformly and independently chosen from integers in $[1, 2C]$. So now each vertex has weight $\mathbf{w}_i = 2C^2 w_i + u_i$. Notice that if a clique was not maximum before, then it is impossible for it to become maximum after assigning new weights. Therefore, the maximum clique with respect to new weights is among one of the maximum cliques with respect to the original weights. The isolating lemma implies that this clique is unique with a probability at least one-half and we may use the scheme mentioned earlier in this section to find it in parallel.

We should mention that this scheme, in fact, results in a Las Vegas type randomized algorithm. No randomization is involved in computing $\omega(G, \mathbf{w})$; only constructing a maximum clique involves probabilistic choices. If the weights generated do not result in a unique maximum weighted clique, our method may return a set that is not even a clique. This can be checked in parallel and the algorithm returns a message of failure; any set returned by the algorithm is a genuine maximum clique with no possibility of error. We summarize these results in the following theorem.

THEOREM 5.5. *Let $G = (V, E)$ be a perfect graph with an integral weight vector $\mathbf{w}$ on its vertices. Let also that $\|\mathbf{w}\|_\infty = O(n^c)$ for some constant c. Then one can compute the maximum weighted clique and the maximum weighted stable set of G in $\tilde{O}(\sqrt{n})$ Las Vegas randomized parallel time using a* P-RAM *model of computation.*

Finally, we remark that presently no representation of the stable set polytope of perfect graphs as projection of a higher dimensional polytope with polynomially many

facets is known. Therefore, STAB G for a perfect graph G serves as an example of a polytope with exponentially many facets on which one can optimize a linear function in polynomial time using interior point methods. In fact, as mentioned in §5.1, one can compute an n-self-concordant barrier for this polytope in polynomial time.

5.3. The maximum induced k-partite subgraph problem. In [47] Narasimhan and Manber generalized the concept of the Lovász number of graphs as follows: Let $\alpha_k(G)$ be the size of the largest induced k-partite subgraph in G. Recall that $\rho(G)$ is the minimum number of cliques that can cover all vertices of G. Then Narasimhan and Manber show that

$$\alpha_k(G) \leq \vartheta_k(G) := \min_{X \in \mathcal{M}^\perp} \sum_{i=1}^{k} \lambda_i(X + J) \leq k\rho(G), \tag{5.9}$$

where J is the matrix of all 1's. For $k = 1$, ϑ_k reduces to the Lovász number ϑ. It is clear now that computing $\vartheta_k(G)$ is an SDP problem and may be solved by interior point methods. Taking the dual of (5.9) we get

$$\begin{aligned} \vartheta_k(G) = \max \quad & J \bullet Y \\ \text{s.t.} \quad & \operatorname{trace} Y = k, \\ & Y \in \mathcal{M}, \\ & 0 \preceq Y \preceq I. \end{aligned} \tag{5.10}$$

It is not difficult to extend the bound of Narasimhan and Manber to the weighted case. Let $\mathbf{w}$ be a weight vector over the vertices of G and $\alpha_k(G, \mathbf{w})$ the maximum weight k-colorable induced subgraph of G.

THEOREM 5.6. *Let* $W = (\sqrt{\mathbf{w}})(\sqrt{\mathbf{w}})^T$. *Then* $\alpha_k(G, \mathbf{w}) \leq \vartheta_k(G, \mathbf{w})$, *where* $\vartheta_k(G, \mathbf{w})$ *is defined as*

$$\begin{aligned} \vartheta_k(G, \mathbf{w}) &:= \min\{\textstyle\sum_1^k \lambda_i(X + W) : X \in \mathcal{M}^\perp\} \\ &= \max\{W \bullet Y : Y \in \mathcal{M} \text{ and } \operatorname{trace} Y = k,\ 0 \preceq Y \preceq I\}. \end{aligned} \tag{5.11}$$

Proof. (This proof is essentially the same as the one given in [31] for the case $k = 1$.) One can transform a weighted graph G into an unweighted one $G_{\mathbf{w}}$ by replacing each vertex i with w_i mutually nonadjacent vertices and then connecting all w_i vertices arising from vertex i to all w_j vertices arising from vertex j if and only if i and j are adjacent in G. Clearly the size of the unweighted maximum k-partite subgraph of $G_{\mathbf{w}}$ equals $\alpha_k(G, \mathbf{w})$. It suffices to show that $\vartheta_k(G, \mathbf{w}) = \vartheta_k(G_{\mathbf{w}})$. Now, in $G_{\mathbf{w}}$ two vertices i and j (respectively, edges uv and kl) are *equivalent* if there is an automorphism of $G_{\mathbf{w}}$ mapping i to j (respectively, uv to kl). In particular all w_i vertices arising from vertex i in G are equivalent; so are the all edges arising from uv. It is clear that if two vertices i and j (respectively, two edges uv and kl) are equivalent, then in (5.10) the corresponding variables Y_{ii} and Y_{jj} (respectively, Y_{uv} and Y_{kl}) are equivalent in the sense that by exchanging these variables (5.10) does not change at all. This in turn implies that among all optimal solutions of (5.10) for graph $G_{\mathbf{w}}$, there are solutions where equivalent vertices (respectively, edges) have identical optimal values for their corresponding variables. In other words, among all optimal solutions of (5.10) for $G_{\mathbf{w}}$, there is one solution $Y^*_{\mathbf{w}}$ with the following property: $Y^*_{\mathbf{w}}$ can be partitioned into an $n \times n$ block matrix, such that the i, j block is a $w_i \times w_j$ matrix with all its entries equal to, say, y^*_{ij}. Now, matrix Y^* whose i, j entry is $y^*_{ij}/\sqrt{w_i w_j}$ is feasible for the max problem in the theorem and it is easy to

verify that $W \bullet Y^* = J \bullet Y^*_{\mathbf{w}} = \vartheta_k(G_{\mathbf{w}})$ and thus, $\vartheta_k(G_{\mathbf{w}}) \leq \vartheta_k(G, \mathbf{w})$. The other direction inequality is also easily verified by reversing the construction given. □

Let $\mho(k)$ be the class of graphs for which $\alpha_k(G') = \vartheta_k(G')$ for all induced subgraphs G'. Then the sublinear parallel time algorithm of Theorem 5.5 may be extended to solve the largest induced k-partite subgraph problem for graphs in class $\mho(k)$. It remains an interesting open problem to fully characterize $\mho(k)$.

5.4. The graph partitioning problem. An important class of combinatorial NP-hard optimization problems which lend themselves to SDP methods for finding upper or lower bounds arise from graph partitioning and cut problems. In many cases such problems result in semidefinite programs with only $O(n)$ variables. Therefore, the interior point methods may be especially efficient as each iteration requires only solving $n \times n$ systems of equations.

The general graph partitioning problem is defined as follows. Suppose we are given a set of integers $m_1 \geq m_2 \geq \cdots \geq m_k$, with $\sum_j m_j = n$. Denote by $\mathbf{m}$ the k-vector made up of m_j's. Also let $G = (V, E)$ be a complete edge-weighted graph with n vertices and each edge $\{i, j\}$ with weight w_{ij}. We want to partition the vertices of G into k subsets such that the jth subset has cardinality m_j, and that the sum of the weights of those edges whose endpoints are in different subsets is minimized. Let us denote this minimum number by $\pi_{\mathbf{m}}(G)$. Computing $\pi_{\mathbf{m}}(G)$ is of course NP-hard. Donath and Hoffman in [17] and [18] derive a lower bound on the size of the minimum partition (see, also, Barnes and Hoffman [5]). Let A be a matrix with $A_{ij} = w_{ij}$ ($A_{ii} = 0$). Then Donath and Hoffman prove the following relation [18]:

$$\pi_{\mathbf{m}}(G) \geq -\frac{1}{2} \min_{\mathbf{1}^T\mathbf{x}=a} \sum_{j=1}^{k} m_j \lambda_j(A + \operatorname{Diag} \mathbf{x}), \tag{5.12}$$

where $a := -\sum w_{ij}$. Again it is clear that computing this bound is an SDP problem. Using the results from §4 and after some simplification we get the following pair of primal and dual SDP programs:

$$\begin{array}{lll} \min & \sum_{i=1}^k i z_i + \mathbf{1}^T\mathbf{x} + \sum_{i=1}^k \operatorname{trace} V_i & \\ \text{s.t.} & z_i I + V_i + (m_i - m_{i+1})\operatorname{Diag}\mathbf{x} \succeq (m_i - m_{i+1})A & \text{for } i = 1, \ldots, k \\ & V_i \succeq 0 \quad \text{for } i = 1, \ldots, k & \end{array} \tag{5.13a}$$

and

$$\begin{array}{lll} \max & A \bullet \left(\sum_{i=1}^k (m_i - m_{i+1}) U_i\right) & \\ \text{s.t.} & \operatorname{trace} U_i = i & \text{for } i = 1, \ldots, k, \\ & \sum_{i=1}^k (m_i - m_{i+1})(U_i)_{jj} = 1 & \text{for } j = 1, \ldots, n, \\ & 0 \preceq U_i \preceq I & \text{for } i = 1, \ldots, k. \end{array} \tag{5.13b}$$

Barnes and Hoffman in [5] describe a method that uses eigenvectors associated with the k largest eigenvalues of the optimal matrix $A + \operatorname{Diag}\mathbf{x}^*$ to generate a partition of the nodes of the graph. See, also, Barnes [6], [7].

An important special case of the graph partitioning problem is the case when all m_i's are equal. In that case the graph partitioning problem simplifies to

$$\begin{array}{llll} \min & (k/n)\mathbf{1}^T\mathbf{x} + \operatorname{trace} V & \max & A \bullet Y \\ \text{s.t.} & V + \operatorname{Diag}\mathbf{x} \succeq A, & \text{s.t.} & Y_{ii} = \frac{k}{n} \quad \text{for } i = 1, \ldots, n, \\ & V \succeq 0, & & 0 \preceq Y \preceq I. \end{array} \tag{5.14}$$

Boppana in [10] considers the graph bisection problem (that is when $k = 2$ and $m_1 = m_2 = n/2$) and derives the following bound on the *bisection width* $\beta(G)$, which is always sharper than (5.14):

$$\beta(G) \geq \frac{1}{4} \max \left[J \bullet (A + \mathrm{Diag}(\mathbf{x})) - n\lambda_1(P(A + \mathrm{Diag}(\mathbf{x})P)) \right],$$

where $P{:=}(I - J/n)$ is the projection operator on the linear space $S{:=}\{\mathbf{x} : \mathbf{1}^T\mathbf{x} = 0\}$. This characterization is equivalent to the primal and dual SDP pair

$$\begin{array}{lll} & \min & nz + \mathbf{1}^T\mathbf{x} \\ (5.15a) & \text{s.t.} & zI - \mathrm{Diag}(\mathbf{x}) - \frac{\mathbf{1}\mathbf{x}^T + \mathbf{x}\mathbf{1}^T}{2n} \succeq A + \frac{JA + AJ}{2n} \end{array}$$

and

$$\begin{array}{lll} & \max & A(I + J/n) \bullet Y \\ (5.15b) & \text{s.t.} & Y_{ii} + (1/n)\sum_{j=1}^{n} Y_{ij} = 1 \quad \text{for } i = 1, \ldots, n, \\ & & Y \succeq 0. \end{array}$$

(Boppana had the min characterization only, the max characterization results by simply taking the dual.) To find an actual bisection Boppana uses an eigenvector corresponding to the largest eigenvalue of $\lambda_1(P(A + \mathrm{Diag}(\mathbf{x}^*)P)$ and outputs the bisection that has the $n/2$ largest component of the eigenvector on one side. Using the primal characterization, Boppana shows that in the unweighted case (i.e., the matrix A is simply the 0-1 adjacency matrix of graph G) one may get the optimal bisection with high probability. The graph bisection problem has important applications in the very large scale integration (VLSI) routing problem. Combining the SDP formulation of Hoffman and Donath, the favorable average case analysis of Boppana, and the interior point technique developed in this paper may result in an effective and practical method for solving this problem. For generalizations of these ideas see [58].

Related to the graph bisection problem is the maximum cut problem: partition the nodes of the graph into two sets such that the number of edges with endpoints on different sets is maximum. Of course one obvious way for finding bounds for this problem is to solve the graph partitioning problem with $k = 2$, $m_1 = i$, and $m_2 = n - i$ for all $i = 1, \ldots, \lfloor n/2 \rfloor$ (notice that in graph partitioning problem max and min characterizations are essentially equivalent by simply changing the weights w_i with $\sum w_j - w_i$). In [16], [56] the following SDP bound is proposed:

$$\min \left\{ \frac{n}{4} \lambda_1(A + \mathrm{Diag}(\mathbf{x})) : \mathbf{1}^T\mathbf{x} = a \right\} \geq \mathrm{MC}(G), \tag{5.16}$$

where $\mathrm{MC}(G)$ is the size of maximum cut in G. Equation (5.16) is equivalent to

$$\begin{array}{llll} & \min & z + (1/n)\mathbf{1}^T\mathbf{x} & \max \quad A \bullet Y \\ (5.17) & \text{s.t.} & zI - \mathrm{Diag}(\mathbf{x}) \succeq -A, & \text{s.t.} \quad Y_{ii} = 1/4, \\ & & & \quad\;\; Y \succeq 0, \end{array}$$

and may be solved by interior point methods. Recently, Goemans and Williamson [25] have shown that the solution of (5.16) yields a cut whose size is guaranteed to be to within 0.87 of the optimum; the previous best result only guaranteed 0.5 of the optimum. Thus far, the best approximation algorithm for the maximum cut problem (as well as for the maximum satisfiability problem) is based on SDP relaxations. For related treatment of maximum cut and graph bisection problem, see [57].

Acknowledgments. Discussions with Stephen Boyd, Uriel Feige, Don Knuth, László Lovász, Yuri Nesterov, Arkadii Nemirovskii, Motakuri Ramana, Rob Womersley, and particularly Michael Overton and Yinyu Ye, have been most useful in preparing this article and removing several errors in the preliminary versions. Also comments of an anonymous referee were instrumental in removing errors in the statements of the Farkas lemma and duality theory in the earlier versions of this paper.

REFERENCES

[1] F. ALIZADEH, *Combinatorial Optimization with Interior Point Methods and Semi-Definite Matrices*, Ph.D. thesis, Computer Science Department, University of Minnesota, Minneapolis, 1991.

[2] ———, *Optimization over positive semi-definite cone; interior-point methods and combinatorial applications*, in Advances in Optimization and Parallel Computing, P. Pardalos, ed., North-Holland, Amsterdam, 1992.

[3] E. ANDERSON AND P. NASH, *Linear Programming in Infinite Dimensional Spaces*, John Wiley & Sons, New York, 1987.

[4] K. M. ANSTREICHER, *A monotone projective algorithm for fractional linear programming*, Algorithmica, 1 (1986), pp. 483–498.

[5] E. BARNES AND A. J. HOFFMAN, *Partitioning, spectra, and linear programming*, in Progress in Combinatorial Optimization, W. E. Pulleyblank, ed., Academic Press, New York, 1984.

[6] E. R. BARNES, *An algorithm for partitioning the nodes of a graph*, SIAM J. Alg. Disc. Meth., 3 (1982), pp. 541–550.

[7] ———, *Partitioning the nodes of a graph*, in Graph Theory with Applications to Algorithms and Computer Science, L. Lesniak, D. R. Lick, and C. E. Wall, eds., John Wiley, New York, 1982.

[8] M. BAZARAA, J. JARVIS, AND H. SHERALI, *Linear Programming and Network Flows*, John Wiley & Sons, New York, 1990.

[9] A. BEN-ISRAEL, A. CHARNES, AND K. O. KORTANEK, *Duality and asymptotic solvability over cones*, Bull. Amer. Math. Soc., 75 (1969), pp. 318–324.

[10] R. B. BOPPANA, *Eigenvalues and graph bisection: an average case analysis*, in Proc. 28th IEEE Annual Symposium on Foundations of Computer Science, 1987.

[11] J. M. BORWEIN AND H. WOLKOWICZ, *Characterization of optimality for the abstract convex program with finite dimensional range*, J. Austral. Math. Soc. Ser. A, 30 (1981), pp. 390–411.

[12] ———, *Facial reduction for a cone-convex programming problem*, J. Austral. Math. Soc. Ser. A, 30 (1981), pp. 369–380.

[13] B. D. CRAVEN AND J. J. KOLIHA, *Generalizations of Farkas' theorem*, SIAM J. Math. Anal., 8 (1977), pp. 983–997.

[14] B. D. CRAVEN AND B. MOND, *Linear programming with matrix variables*, Linear Algebra Appl., 38 (1981), pp. 73–80.

[15] J. CULLUM, W. E. DONATH, AND P. WOLFE, *The minimization of certain nondifferentiable sums of eigenvalue problems*, Math. Programming Stud., 3 (1975), pp. 35–55.

[16] C. DELORME AND S. POLJAK, *Laplacian eigenvalues and the maximum cut problem*, Math. Programming, 62 (1993), pp. 557,574.

[17] W. E. DONATH AND A. J. HOFFMAN, *Algorithms for partitioning graphs and computer logic based on eigenvectors of connection matrices*, IBM Technical Disclosure Bulletin, 15, 1972.

[18] ———, *Lower bounds for the partitioning of graphs*, IBM J. Res. and Devel., 17 (1973).

[19] R. J. DUFFIN, *Infinite programs*, in Linear Inequalities and Related Systems, H. W. Kuhn and A. W. Tucker, eds., Princeton University Press, Princeton, NJ, 1956, pp. 157–170.

[20] M. FAN, *A quadratically convergent local algorithm on minimizing the largest eigenvalue of a symmetric matrix*, Linear Algebra Appl., 188,189 (1993), pp. 207–230.

[21] U. FEIGE AND L. LOVÁSZ, *Two-prover one-round proof systems: their power and their problems*, in Proc. 24th Annual ACM Symposium on Theory of Computing, 1992, pp. 733–744.

[22] P. A. FILLMORE AND J. P. WILLIAMS, *Some convexity theorems for matrices*, Glasgow Math. J., 12 (1971), pp. 110–117.

[23] R. FLETCHER, *Semi-definite matrix constraints in optimization*, SIAM J. Control Optim., 23 (1985), pp. 493–513.

[24] S. FRIEDLAND, J. NOCEDAL, AND M. L. OVERTON, *The formulation and analysis of numerical methods for inverse eigenvalue problems*, SIAM J. Numer. Anal., 24 (1987), pp. 634–667.

[25] M. X. GOEMANS AND D. P. WILLIAMSON, *Improved approximation algorithms for maximum*

cut and satisfiability problems using semidefinite programming, J. Assoc. Comput. Mach., 1994, to appear; A preliminary version appeared in Proc. 26th Annual ACM Symposium on Theory of Computing.

[26] A. V. GOLDBERG, S. A. PLOTKIN, D. SHMOYS, AND E. TARDOS, *Interior-point methods in parallel computation*, SIAM J. Comput., 21 (1991), pp. 140–150.

[27] D. GOLDFARB AND M. J. TODD, *Linear programming*, in Optimization, G. L. Nemhauser, A. H. G. Rinnooy Kan, and M. J. Todd, eds., Handbooks in Operations Research and Management Sciences, North–Holland, Amsterdam, 1989.

[28] C. C. GONZAGA, *An algorithm for solving linear programming in $O(n^3L)$ operations*, in Progress in Mathematical Programming, N. Megiddo, ed., Springer-Verlag, Berlin, 1989, pp. 1–28.

[29] A. GRAHAM, *Kronecker Products and Matrix Calculus: with Applications*, Ellis Horwood, London, 1981.

[30] M. GRÖTSCHEL, L. LOVÁSZ, AND A. SCHRIJVER, *The ellipsoid method and its consequences in combinatorial optimization*, Combinatorica, 1 (1981), pp. 169–197.

[31] ———, *Polynomial algorithms for perfect graphs*, in Perfect Graphs, C. Berge and V. Chvatal, eds., North Holland, Amsterdam, 1984; also Ann. Discrete Math., 21.

[32] ———, *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, New York, 1988.

[33] R. HORN AND C. JOHNSON, *Matrix Analysis*, Cambridge University Press, Cambridge, 1985.

[34] L. HURWICZ, *Programming in linear spaces*, in Studies in Linear and Nonlinear Programming, II, K. J. Arrow, L. Hurwicz, and H. Uzawa, eds., Stanford University Press, Stanford, CA, 1958, pp. 4–102.

[35] 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.

[36] N. KARMARKAR, *A new polynomial-time algorithm for linear programming*, Combinatorica, 4 (1984), pp. 373–395.

[37] M. KOJIMA, N. MEGIDDO, T. NOMA, AND A. YOSHISE, *A unified approach to interior point algorithms for linear complementarity problems*, No. 538 in Lecture Notes in Computer Science, Springer–Verlag, New York, 1991.

[38] M. KOJIMA, S. MIZUNI, AND A. YOSHISE, *A primal-dual interior-point algorithm for linear programming*, in Progress in Mathematical Programming, N. Megiddo, ed., Springer-Verlag, Berlin, 1989.

[39] P. LANCASTER AND M. TISMENETSKY, *The Theory of Matrices*, Academic Press, New York, 1985.

[40] L. LOVÁSZ, *Normal hypergraphs and the weak perfect graph conjecture*, Discrete Math., 2 (1972), pp. 253–267.

[41] ———, *On the Shannon capacity of a graph*, IEEE Trans. Inform. Theory, 25 (1979), pp. 1–7.

[42] L. LOVÁSZ AND A. SCHRIJVER, *Cones of matrices and setfunctions, and 0-1 optimization*, SIAM J. Optim., 1 (1991), pp. 166–190.

[43] N. MEGIDDO, *Pathways to the optimal set in linear programming*, in Progress in Mathematical Programming, N. Megiddo, ed., Springer-Verlag, Berlin, New York, 1989.

[44] R. C. MONTEIRO AND I. ADLER, *Interior path following primal–dual algorithms. Part* I*: Linear programming*, Math. Programming, 44 (1989), pp. 27–41.

[45] ———, *Interior path following primal–dual algorithms. Part* II*: Convex quadratic programming*, Math. Programming, 44 (1989), pp. 43–66.

[46] K. MULMULEY, U. V. VAZIRANI, AND V. V. VAZIRANI, *Matching is as easy as matrix inversion*, Combinatorica, (1987), pp. 105–131.

[47] G. NARASIMHAN AND R. MANBER, *A generalization of Lovász's ϑ function*, in Polyhedral Combinatorics, W. Cook, and P. D. Seymour, eds., Vol. 1, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Amer. Math. Soc. Assoc. for Computing Machinery, 1990, pp. 19–27.

[48] Y. NESTEROV AND A. NEMIROVSKII, *Self-concordant functions and polynomial time methods in convex programming*, Central Economical and Mathematical Institute, USSR Academy of Science, Moscow, 1990.

[49] ———, *Acceleration of the path–following method for optimization over the cone of positive semidefinite matrices*, manuscript, 1992.

[50] ———, *Interior Point Polynomial Methods in Convex Programming: Theory and Applications*, Society for Industrial and Applied Mathematics, Philadelphia, 1994.

[51] M. L. OVERTON, *On minimizing the maximum eigenvalue of a symmetric matrix*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 256–268.

[52] ———, *Large-scale optimization of eigenvalues*, SIAM J. Optim., 2 (1992), pp. 88–120.

[53] M. L. OVERTON AND R. S. WOMERSLEY, *Optimality conditions and duality theory for min-*

imizing sums of the largest eigenvalues of symmetric matrices, Math. Programming, 62 (1993), pp. 321–357.

[54] ———, *On the sum of the largest eigenvalues of a symmetric matrix*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 41–45.

[55] G. PATAKI, *Algorithms for linear programs over cones and semi-definite programming*, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, 1993, manuscript.

[56] S. POLJAK AND F. RENDL, *Solving the max-cut problem using eigenvalues*, Tech. Report 91735-OR, Forschungsinstitut Für Diskrete Mathematik, Institut Für ökonometrie und Operations Research, Rheinische Friedrich–Wilhelms-Universität, Bonn, November 1991.

[57] ———, *Nonpolyhedral relaxations of graph-bisection problems*, Tech. Report 92-55, DIMACS, November 1992.

[58] F. RENDL AND H. WOLKOWICZ, *A projection technique for partitioning the nodes of a graph*, CORR Report 90-20, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, 1993.

[59] R. T. ROCKAFELLAR, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[60] A. SCHRIJVER, *Theory of Linear and Integer Programming*, J. Wiley & Sons, New York, 1986.

[61] A. SHAPIRO, *Extremal problems on the set of nonnegative definite matrices*, Linear Algebra Appl., 67 (1985), pp. 7–18.

[62] S. SUBRAMANI, *Sums of Singular Values*, Master's thesis, School of Mathematics, University of New South Wales, Kensington, Australia, 1993.

[63] L. VANDENBERGHE AND S. BOYD, *Primal–dual potential reduction method for problems involving matrix inequalities*, Tech. Report, Information Systems Laboratory, Department of Electrical Engineering, Stanford University, Stanford, CA, January 1993; Math. Programming, to appear.

[64] H. WOLKOWICZ, *Some applications of optimization in matrix theory*, Linear Algebra Appl., 40 (1981), pp. 101–118.

[65] M. YANNAKAKIS, *Expressing combinatorial optimization problems by linear programs*, J. Comput. Syst. Sci., 43 (1991), pp. 441–466.

[66] Y. YE, *A class of projective transformations for linear programming*, SIAM J. Comput., 19 (1990), pp. 457–466.

[67] ———, *An $O(n^3L)$ potential reduction algorithm for linear programming*, Math. Programming, 50 (1991), pp. 239–258.

SIAM J. OPTIMIZATION
Vol. 5, No. 1, pp. 52–67, February 1995

INFEASIBLE-INTERIOR-POINT PRIMAL-DUAL POTENTIAL-REDUCTION ALGORITHMS FOR LINEAR PROGRAMMING*

SHINJI MIZUNO†, MASAKAZU KOJIMA‡, AND MICHAEL J. TODD§

Abstract. In this paper, primal-dual potential-reduction algorithms are proposed that can start from an infeasible interior point. The authors first describe two such algorithms and show that both are polynomial-time bounded. One of the algorithms decreases the Tanabe–Todd–Ye primal-dual potential function by a constant at each iteration under the condition that the duality gap decreases by at most the same ratio as the infeasibility. The other algorithm reduces a new potential function, which has one more term in the Tanabe–Todd–Ye potential function, by a fixed constant at each iteration without any other conditions on the step size. Finally, modifications of these methods are described (incorporating centering steps) that dramatically decrease their computational complexity. The algorithms also extend to the case of monotone linear complementarity problems.

Key words. polynomial time, linear programming, primal-dual, infeasible-interior-point algorithm, potential function

AMS subject classifications. 90C05, 65K05

1. Introduction.

The primal-dual infeasible-interior-point algorithm for linear programming is a simple variant of the primal-dual (feasible-)interior-point algorithms developed by Megiddo [11], Kojima, Mizuno, and Yoshise [4], [5], Monteiro and Adler [15], [16], and Tanabe [18]. The algorithm can start from an infeasible point, while interior-point algorithms must start from a feasible point. When we solve a given problem by an interior-point algorithm, we need to construct an artificial problem to get an initial feasible point. The advantage of the infeasible-interior-point algorithm over the interior-point algorithm is in solving the given problem directly. (This is a very significant advantage in practice. In theory, the complexity analysis of most of these methods still requires initial solutions that may need to have very large (big M) components.) The algorithm has been studied by Lustig [8], Lustig, Marsten, and Shanno [9], Marsten et al. [10], and Tanabe [19] and is known to be one of the most efficient interior-point algorithms (see for example [9], [10]). Kojima, Megiddo, and Mizuno [3] demonstrated the global convergence of an infeasible-interior-point algorithm. Then Zhang [22], Mizuno [12], and Potra [17] proved polynomial-time convergence of certain infeasible-interior-point algorithms. Those algorithms generate a sequence of points in a neighborhood of the path of centers and they are classified as path-following algorithms.

In the framework of interior-point algorithms, potential functions have played important roles in determining a step size at each iteration and in obtaining a theoretical upper bound on the number of iterations (see Karmarkar [2], Ye [20], Kojima, Mizuno,

* Received by the editors September 2, 1992; accepted for publication (in revised form) September 23, 1993.

† The Institute of Statistical Mathematics, 4-6-7 Minami-Azabu, Minato-ku, Tokyo 106, Japan. This research was performed while the author was visiting Cornell University, April 1992–January 1993, as an overseas research scholar of The Ministry of Science, Education and Culture of Japan.

‡ Department of Information Sciences, Tokyo Institute of Technology, 2-12-1 Oh-Okayama, Meguro, Tokyo 152, Japan. This research was partially performed while the author was visiting Cornell University in July 1992.

§ School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York. This research was supported in part by the National Science Foundation, Air Force Office of Scientific Research, and Office of Naval Research through a National Science Foundation grant DMS-8920550.

and Yoshise [6]). Kojima, Noma, and Yoshise [7] investigated the global convergence of infeasible-interior-point (both potential-reduction and path-following) algorithms for monotone complementarity problems. In this paper, we propose polynomial-time primal-dual potential-reduction algorithms that start from an infeasible interior point.

Let $\boldsymbol{A}$ be an $m \times n$ matrix, $\boldsymbol{b} \in R^m$, and $\boldsymbol{c} \in R^n$. Consider the standard form linear program

$$\begin{array}{lll} \text{(P)} & \text{Minimize} & \boldsymbol{c}^T\boldsymbol{x} \\ & \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \geq \boldsymbol{0} \end{array}$$

and its dual

$$\begin{array}{lll} \text{(D)} & \text{Maximize} & \boldsymbol{b}^T\boldsymbol{y} \\ & \text{subject to} & \boldsymbol{A}^T\boldsymbol{y} + \boldsymbol{z} = \boldsymbol{c},\ \boldsymbol{z} \geq \boldsymbol{0}. \end{array}$$

We assume that the matrix $\boldsymbol{A}$ has full row rank, i.e., rank $\boldsymbol{A} = m$. We call $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ an (infeasible) interior point if $\boldsymbol{x} > \boldsymbol{0}$ and $\boldsymbol{z} > \boldsymbol{0}$, and a feasible interior point if in addition $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ and $\boldsymbol{A}^T\boldsymbol{y} + \boldsymbol{z} = \boldsymbol{c}$.

Let $(\boldsymbol{x}^1, \boldsymbol{y}^1, \boldsymbol{z}^1)$ be an interior point and $\sigma > 0$ be such that

$$(\boldsymbol{x}^1)^T\boldsymbol{z}^1 > \sigma\|(\boldsymbol{A}\boldsymbol{x}^1 - \boldsymbol{b}, \boldsymbol{A}^T\boldsymbol{y}^1 + \boldsymbol{z}^1 - \boldsymbol{c})\|,$$

where $\|\cdot\|$ denotes the ℓ_2-norm. For a constant $\nu \geq 0$, we define two primal-dual potential functions:

$$\phi(\boldsymbol{x}, \boldsymbol{z}) := (n + \nu)\ln(\boldsymbol{x}^T\boldsymbol{z}) - \sum_{i=1}^{n} \ln(x_i z_i) - n \ln n,$$

$$\begin{aligned} \psi(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) := (n + \nu + 1)\ln(\boldsymbol{x}^T\boldsymbol{z}) - \sum_{i=1}^{n} \ln(x_i z_i) - n \ln n \\ - \ln(\boldsymbol{x}^T\boldsymbol{z} - \sigma\|(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}, \boldsymbol{A}^T\boldsymbol{y} + \boldsymbol{z} - \boldsymbol{c})\|). \end{aligned}$$

The first is known as the Tanabe–Todd–Ye primal-dual potential function (used for feasible-interior-point algorithms) and the second is defined here for an infeasible-interior-point algorithm. If $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ is feasible, $\psi(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = \phi(\boldsymbol{x}, \boldsymbol{z})$. Note that ψ involves the norm of a vector formed from the primal and dual infeasibilities. It appears that this would be very sensitive to different scalings of the original problem. However, we see in (6) below that each component of this vector decreases at the same rate during the algorithms. Hence the norm measures how much each infeasibility has been reduced.

Sections 2–4 of this paper construct two infeasible-interior-point algorithms, namely, Algorithms I and II, which start from the initial point $(\boldsymbol{x}^1, \boldsymbol{y}^1, \boldsymbol{z}^1)$ and generate a sequence $\{(\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{z}^k)\}$ of interior points. Algorithms I and II decrease the potential functions ϕ and ψ at each iteration, respectively. The step size α at the kth iterate $(\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{z}^k)$ of Algorithm I is determined such that ϕ decreases at least a constant value and an extra condition holds, while Algorithm II does not need any such condition. So Algorithm I is a constrained potential-reduction algorithm, while Algorithm II is a pure potential-reduction algorithm. In the worst case, the decrease in the potential functions at each iteration is only $\Omega(n^{-2})$, and this leads to a complexity bound of $O(n^{2.5}L)$ iterations, where L is related to the initialization and the

termination criterion of the algorithms. Then §5 describes variants that require only $O(nL)$ iterations by adding centering steps when the current iterate lies outside a wide neighborhood of the path of centers. The centering steps keep the "duality gap" and the infeasibilities fixed while decreasing the potential functions ϕ and ψ. Finally, §6 contains a discussion of why the complexity bounds of these infeasible-interior-point methods are so much higher than those for feasible-interior-point algorithms, and shows how the algorithms also extend to monotone linear complementarity problems (LCPs). We chose to confine ourselves to the more familiar setting of linear programming for the main development.

2. A constrained potential-reduction algorithm. The path of centers consists of the solutions $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ to the system of equations

$$\begin{pmatrix} \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \\ \boldsymbol{A}^T\boldsymbol{y} + \boldsymbol{z} - \boldsymbol{c} \\ \boldsymbol{X}\boldsymbol{z} - \mu\boldsymbol{e} \end{pmatrix} = \boldsymbol{0} \tag{1}$$

for all $\mu > 0$. Here $\boldsymbol{X} := \mathrm{diag}(\boldsymbol{x})$ denotes the $n \times n$ diagonal matrix containing the coordinates of a vector $\boldsymbol{x} \in R^n$ and $\boldsymbol{e} := (1, \ldots, 1)^T \in R^n$. At each iteration, we assign the value $(\boldsymbol{x}^k)^T\boldsymbol{z}^k/(n+\nu)$ to the parameter μ, and then compute the Newton direction $(\boldsymbol{\Delta x}, \boldsymbol{\Delta y}, \boldsymbol{\Delta z})$ at $(\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{z}^k)$ for the system (1) of equations; that is, $(\boldsymbol{\Delta x}, \boldsymbol{\Delta y}, \boldsymbol{\Delta z})$ is the unique solution of the system of linear equations

$$\begin{pmatrix} \boldsymbol{A} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}^T & \boldsymbol{I} \\ \boldsymbol{Z}^k & \boldsymbol{0} & \boldsymbol{X}^k \end{pmatrix} \begin{pmatrix} \boldsymbol{\Delta x} \\ \boldsymbol{\Delta y} \\ \boldsymbol{\Delta z} \end{pmatrix} = - \begin{pmatrix} \boldsymbol{A}\boldsymbol{x}^k - \boldsymbol{b} \\ \boldsymbol{A}^T\boldsymbol{y}^k + \boldsymbol{z}^k - \boldsymbol{c} \\ \boldsymbol{X}^k\boldsymbol{z}^k - \mu\boldsymbol{e} \end{pmatrix}, \tag{2}$$

where $\boldsymbol{X}^k := \mathrm{diag}(\boldsymbol{x}^k)$ and $\boldsymbol{Z}^k := \mathrm{diag}(\boldsymbol{z}^k)$.

Let ρ be a positive constant for which we want to find the optimal solutions $\boldsymbol{x}^*$ of (P) and $(\boldsymbol{y}^*, \boldsymbol{z}^*)$ of (D), if they exist, such that

$$\|(\boldsymbol{x}^*, \boldsymbol{z}^*)\|_\infty \leq \rho.$$

ALGORITHM I

Step 1. Choose $\gamma_0 \in (0, 1]$ and a positive constant δ (which may depend on n and ν). Set $(\boldsymbol{x}^1, \boldsymbol{y}^1, \boldsymbol{z}^1) := \gamma_0\rho(\boldsymbol{e}, \boldsymbol{0}, \boldsymbol{e})$. Let $k := 1$.

Step 2. If $(\boldsymbol{x}^k)^T\boldsymbol{z}^k \leq \epsilon$ then stop.

Step 3. Let $\mu := (\boldsymbol{x}^k)^T\boldsymbol{z}^k/(n+\nu)$. Compute the solution $(\boldsymbol{\Delta x}, \boldsymbol{\Delta y}, \boldsymbol{\Delta z})$ at $(\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{z}^k)$ of the system (2) of equations.

Step 4. Find a step size α such that

$$\phi(\boldsymbol{x}^k + \alpha\boldsymbol{\Delta x}, \boldsymbol{z}^k + \alpha\boldsymbol{\Delta z}) \leq \phi(\boldsymbol{x}^k, \boldsymbol{z}^k) - \delta, \tag{3}$$

$$(\boldsymbol{x}^k + \alpha\boldsymbol{\Delta x})^T(\boldsymbol{z}^k + \alpha\boldsymbol{\Delta z}) \geq (1-\alpha)(\boldsymbol{x}^k)^T\boldsymbol{z}^k. \tag{4}$$

If we cannot find such a step size then stop.

Step 5. Let $(\boldsymbol{x}^{k+1}, \boldsymbol{y}^{k+1}, \boldsymbol{z}^{k+1}) := (\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{z}^k) + \alpha(\boldsymbol{\Delta x}, \boldsymbol{\Delta y}, \boldsymbol{\Delta z})$. Increase k by 1 and go to Step 2.

The direction $(\boldsymbol{\Delta x}, \boldsymbol{\Delta y}, \boldsymbol{\Delta z})$ is, except for the choice of μ, the same as in the earlier primal-dual infeasible-interior-point algorithms. If the current iterate is feasible, our

choice of μ yields the direction that is the projected scaled steepest descent direction for the potential function ϕ (or ψ) [6].

Since $\phi(\boldsymbol{x},\boldsymbol{z}) \geq \nu \ln(\boldsymbol{x}^T\boldsymbol{z})$ and the potential function decreases by a constant δ at each iteration, Algorithm I terminates in $O(\nu L/\delta)$ iterations provided that $\phi(\boldsymbol{x}^1,\boldsymbol{z}^1) = O(\nu L)$ and $\ln(1/\epsilon) = O(L)$. If $L \geq \ln n$ and $\ln\rho = O(L)$ then $\phi(\boldsymbol{x}^1,\boldsymbol{z}^1) = O(\nu L)$. In the next section, we show that if there are optimal solutions $\boldsymbol{x}^*$ of (P) and $(\boldsymbol{y}^*,\boldsymbol{z}^*)$ of (D) such that $\|(\boldsymbol{x}^*,\boldsymbol{z}^*)\|_\infty \leq \rho$, then there exists a step size α, which satisfies (3) and (4) for $\delta := \gamma_0^4/(300(n+\nu)^2)$. (Condition (4) is what makes this a constrained potential-reduction algorithm.) Hence we have the following result.

THEOREM 1. *Let $L \geq \ln n$ (L may be the input size of problem* (P)*) and $\gamma_0 \in (0,1]$. Suppose that $\ln\rho = O(L)$, $\ln(1/\epsilon) = O(L)$, $\nu \geq \sqrt{n}$, and $\delta := \gamma_0^4/(300(n+\nu)^2)$. Then Algorithm* I *terminates in $O(\nu(n+\nu)^2 L)$ iterations. If the algorithm stops in Step* 2*, we get an approximate solution; otherwise (if it stops in Step* 4*) there are no optimal solutions $\boldsymbol{x}^*$ of* (P) *and $(\boldsymbol{y}^*,\boldsymbol{z}^*)$ of* (D) *such that $\|(\boldsymbol{x}^*,\boldsymbol{z}^*)\|_\infty \leq \rho$.*

3. Analysis of Algorithm I. Theorem 1 follows from the following four lemmas. The first lemma gives a bound on the decrease in ϕ.

LEMMA 2 (Kojima, Mizuno, and Yoshise [6]). *For any n-vectors $\boldsymbol{x} > \boldsymbol{0}$, $\boldsymbol{z} > \boldsymbol{0}$, $\boldsymbol{\Delta x}$, $\boldsymbol{\Delta z}$, and $\alpha > 0$ such that $\|\alpha \boldsymbol{X}^{-1}\boldsymbol{\Delta x}\|_\infty \leq \tau$ and $\|\alpha \boldsymbol{Z}^{-1}\boldsymbol{\Delta z}\|_\infty \leq \tau$ for a constant $\tau \in (0,1)$, we have*

$$\phi(\boldsymbol{x}+\alpha\boldsymbol{\Delta x},\boldsymbol{z}+\alpha\boldsymbol{\Delta z}) \leq \phi(\boldsymbol{x},\boldsymbol{z}) + \left(\frac{n+\nu}{\boldsymbol{x}^T\boldsymbol{z}}\boldsymbol{e} - (\boldsymbol{X}\boldsymbol{Z})^{-1}\boldsymbol{e}\right)^T (\boldsymbol{Z}\boldsymbol{\Delta x} + \boldsymbol{X}\boldsymbol{\Delta z})\alpha$$
$$+ \left((n+\nu)\frac{\boldsymbol{\Delta x}^T\boldsymbol{\Delta z}}{\boldsymbol{x}^T\boldsymbol{z}} + \frac{\|\boldsymbol{X}^{-1}\boldsymbol{\Delta x}\|^2 + \|\boldsymbol{Z}^{-1}\boldsymbol{\Delta z}\|^2}{2(1-\tau)}\right)\alpha^2. \tag{5}$$

The next result is important in analyzing the linear term above, with $\boldsymbol{v} := \boldsymbol{X}^{1/2}\boldsymbol{Z}^{1/2}\boldsymbol{e}$.

LEMMA 3 (Lemma 2.5 in Kojima, Mizuno, and Yoshise [6]). *For any n-vector $\boldsymbol{v} > \boldsymbol{0}$ and $\nu \geq \sqrt{n}$,*

$$\left\|\boldsymbol{V}^{-1}\boldsymbol{e} - \frac{n+\nu}{\boldsymbol{v}^T\boldsymbol{v}}\boldsymbol{v}\right\| \geq \frac{\sqrt{3}}{2v_{\min}},$$

where $\boldsymbol{V} := \operatorname{diag}(\boldsymbol{v})$ and $v_{\min} := \min_i v_i$.

Note that $\nu = \sqrt{n}$ is fixed in [6], but the proof is valid for any $\nu \geq \sqrt{n}$.

Let α^k be the step size at the kth iteration of Algorithm I. We define a sequence $\{\theta^k\}$ by

$$\theta^1 := 1 \quad \text{and} \quad \theta^{k+1} := (1-\alpha^k)\theta^k \quad \text{for } k = 1,2,3,\ldots.$$

As shown in [3], we have

$$(\boldsymbol{A}\boldsymbol{x}^k - \boldsymbol{b}, \boldsymbol{A}^T\boldsymbol{y}^k + \boldsymbol{z}^k - \boldsymbol{c}) = \theta^k(\boldsymbol{A}\boldsymbol{x}^1 - \boldsymbol{b}, \boldsymbol{A}^T\boldsymbol{y}^1 + \boldsymbol{z}^1 - \boldsymbol{c}). \tag{6}$$

The following result is used to bound the second-order term in (5). The parameter γ_1 is introduced to allow this lemma to be used in the analysis of Algorithm II also.

LEMMA 4 (based on Mizuno [12]). *Let $\gamma_0 \in (0,1]$, $\gamma_1 \in (0,1]$, and $\rho > 0$. Suppose that*

$$\begin{aligned} (\boldsymbol{x}^1,\boldsymbol{y}^1,\boldsymbol{z}^1) &= \gamma_0\rho(\boldsymbol{e},\boldsymbol{0},\boldsymbol{e}), \\ (\boldsymbol{A}\boldsymbol{x}^k - \boldsymbol{b}, \boldsymbol{A}^T\boldsymbol{y}^k + \boldsymbol{z}^k - \boldsymbol{c}) &= \theta^k(\boldsymbol{A}\boldsymbol{x}^1 - \boldsymbol{b}, \boldsymbol{A}^T\boldsymbol{y}^1 + \boldsymbol{z}^1 - \boldsymbol{c}), \\ (\boldsymbol{x}^k)^T\boldsymbol{z}^k &\geq \theta^k\gamma_1(\boldsymbol{x}^1)^T\boldsymbol{z}^1. \end{aligned} \tag{7}$$

If there exist optimal solutions x^* *of* (P) *and* (y^*, z^*) *of* (D) *such that* $\|(x^*, z^*)\|_\infty \leq \rho$, *then we have*

$$\|D^{-1}\Delta x\| \leq \frac{5}{\gamma_0^2\gamma_1}\frac{(x^k)^T z^k}{v_{\min}},$$
$$\|D\Delta z\| \leq \frac{5}{\gamma_0^2\gamma_1}\frac{(x^k)^T z^k}{v_{\min}},$$

where $D := (X^k)^{1/2}(Z^k)^{-1/2}$ *and* $v_{\min} := \min_i \sqrt{x_i^k z_i^k}$.

Proof. Assume that there exist optimal solutions x^* of (P) and (y^*, z^*) of (D) such that $\|(x^*, z^*)\|_\infty \leq \rho$. Then we have

$$(\theta^k x^1 + (1-\theta^k)x^* - x^k)^T(\theta^k z^1 + (1-\theta^k)z^* - z^k) = 0, \tag{8}$$

which implies

$$\begin{aligned}&(\theta^k x^1 + (1-\theta^k)x^*)^T z^k + (\theta^k z^1 + (1-\theta^k)z^*)^T x^k \\ &\quad = (\theta^k x^1 + (1-\theta^k)x^*)^T(\theta^k z^1 + (1-\theta^k)z^*) + (x^k)^T z^k.\end{aligned}$$

By using this equality, $x^1 = z^1 = \gamma_0\rho e$, $x^* \leq \rho e$, $z^* \leq \rho e$, and $x_i^* z_i^* = 0$ for each i, we have

$$\begin{aligned}\theta^k(\gamma_0\rho)\|(x^k, z^k)\|_1 &= \theta^k((z^1)^T x^k + (x^1)^T z^k) \\ &\leq (\theta^k x^1 + (1-\theta^k)x^*)^T z^k + (\theta^k z^1 + (1-\theta^k)z^*)^T x^k \\ &= (\theta^k x^1 + (1-\theta^k)x^*)^T(\theta^k z^1 + (1-\theta^k)z^*) + (x^k)^T z^k \\ &\leq n\theta^k\gamma_0\rho^2 + (x^k)^T z^k,\end{aligned}$$

where the last inequality follows from the fact that for each i one of $\theta^k x_i^1 + (1-\theta^k)x_i^*$ and $\theta^k z_i^1 + (1-\theta^k)z_i^*$ is at most $\theta^k\gamma_0\rho$ and the other is at most ρ. From (7), $(x^k)^T z^k \geq \theta^k\gamma_1(x^1)^T z^1 = n\theta^k\gamma_0^2\gamma_1\rho^2$. Hence we have

$$\theta^k\gamma_0\rho\|(x^k, z^k)\|_1 \leq \frac{2}{\gamma_0\gamma_1}(x^k)^T z^k. \tag{9}$$

From (2) and (6), we get

$$\begin{aligned}&\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z^k & 0 & X^k \end{pmatrix}\begin{pmatrix} \Delta x + \theta^k(x^1 - x^*) \\ \Delta y + \theta^k(y^1 - y^*) \\ \Delta z + \theta^k(z^1 - z^*) \end{pmatrix} \\ &= -\begin{pmatrix} 0 \\ 0 \\ X^k z^k - \mu e - \theta^k Z^k(x^1 - x^*) - \theta^k X^k(z^1 - z^*) \end{pmatrix}.\end{aligned} \tag{10}$$

Then we have via a straightforward computation (see also Mizuno [12])

$$\begin{aligned}D^{-1}\Delta x = &-\theta^k Q D^{-1}(x^1 - x^*) + \theta^k(I - Q)D(z^1 - z^*) \\ &-(I - Q)(X^k Z^k)^{-1/2}(X^k z^k - \mu e),\end{aligned} \tag{11}$$

where $Q := DA^T(AD^2A^T)^{-1}AD$. Since Q and $I - Q$ are orthogonal projections, we have

$$\|D^{-1}\Delta x\| \leq \theta^k\|D^{-1}(x^1 - x^*)\| + \theta^k\|D(z^1 - z^*)\| + \|(X^k Z^k)^{-1/2}(X^k z^k - \mu e)\|.$$

From the definition of ρ, $-\rho e \le x^1 - x^* \le \rho e$ and $-\rho e \le z^1 - z^* \le \rho e$. Thus we have

$$\begin{aligned}\|D^{-1}\Delta x\| &\le \theta^k\rho\|D^{-1}e\| + \theta^k\rho\|De\| + \|(X^kZ^k)^{-1/2}(X^kz^k - \mu e)\| \\ &\le \theta^k\rho\|(X^kZ^k)^{-1/2}\|(\|z^k\| + \|x^k\|) + \sqrt{\sum_{i=1}^n((x_i^kz_i^k)^{1/2} - \mu(x_i^kz_i^k)^{-1/2})^2} \\ &\le 2\theta^k\rho\|(X^kZ^k)^{-1/2}\|\|(x^k, z^k)\| + \sqrt{(x^k)^Tz^k - 2n\mu + \sum_{i=1}^n \mu^2(x_i^kz_i^k)^{-1}}. \qquad (12)\end{aligned}$$

By using $v_{\min} = \min_i \sqrt{x_i^kz_i^k}$, $\mu = (x^k)^Tz^k/(n+\nu)$, and (from (9))

$$\|(x^k, z^k)\| \le \|(x^k, z^k)\|_1 \le \frac{2}{\gamma_0^2\gamma_1\theta^k\rho}(x^k)^Tz^k, \qquad (13)$$

we see

$$\begin{aligned}\|D^{-1}\Delta x\| &\le 2\theta^k\rho\frac{1}{v_{\min}}\frac{2}{\gamma_0^2\gamma_1\theta^k\rho}(x^k)^Tz^k \\ &\quad + \sqrt{(x^k)^Tz^k - 2\frac{n}{n+\nu}(x^k)^Tz^k + \frac{n((x^k)^Tz^k)^2}{(n+\nu)^2(v_{\min})^2}} \\ &= \left(\frac{4}{\gamma_0^2\gamma_1} + \sqrt{\left(1 - 2\frac{n}{n+\nu}\right)\frac{(v_{\min})^2}{(x^k)^Tz^k} + \frac{n}{(n+\nu)^2}}\right)\frac{(x^k)^Tz^k}{v_{\min}} \\ &\le \frac{5}{\gamma_0^2\gamma_1}\frac{(x^k)^Tz^k}{v_{\min}}. \qquad (14)\end{aligned}$$

The other inequality follows from a similar analysis of

$$\begin{aligned}D\Delta z = &-\theta^k(I - Q)D(z^1 - z^*) + \theta^kQD^{-1}(x^1 - x^*) \\ &-Q(X^kZ^k)^{-1/2}(X^kz^k - \mu e). \qquad (15)\end{aligned}$$

□

Note that if (4) holds until the $(k-1)$th iteration, then we have (7) for $\gamma_1 = 1$ by the definition of θ^k. This also shows that if Algorithm I stops in Step 2, the infeasibility of x and (y, z) has been reduced at least as much as the duality gap, so we do have approximate solutions. Indeed, we have almost optimal solutions to a nearby linear programming problem and its dual.

Finally, the lemma below completes the proof of Theorem 1.

LEMMA 5. *Let* $\nu \ge \sqrt{n}$, $\gamma_0 \in (0, 1]$, $\gamma_1 \in (0, 1]$, *and*

$$\delta := \frac{\gamma_0^4\gamma_1^2}{300(n+\nu)^2},$$

and suppose (7) holds. If there exist optimal solutions x^* *of* (P) *and* (y^*, z^*) *of* (D) *such that* $\|(x^*, z^*)\|_\infty \le \rho$ *then we have* (3) *and* (4) *for*

$$\alpha := \frac{\gamma_0^4\gamma_1^2v_{\min}^2}{100(n+\nu)(x^k)^Tz^k}$$

at the kth iteration.

Proof. Let $\boldsymbol{v} := (\boldsymbol{X}^k\boldsymbol{Z}^k)^{1/2}\boldsymbol{e}$ and $\boldsymbol{D} := (\boldsymbol{X}^k)^{1/2}(\boldsymbol{Z}^k)^{-1/2}$. Then if τ satisfies the hypotheses of Lemma 2, we have

$$\begin{aligned}
&\phi(\boldsymbol{x}^k + \alpha\Delta\boldsymbol{x}, \boldsymbol{z}^k + \alpha\Delta\boldsymbol{z}) - \phi(\boldsymbol{x}^k, \boldsymbol{z}^k) \\
&\leq \left(\frac{n+\nu}{\boldsymbol{v}^T\boldsymbol{v}}\boldsymbol{e} - \boldsymbol{V}^{-2}\boldsymbol{e}\right)^T \boldsymbol{V}(\boldsymbol{D}^{-1}\Delta\boldsymbol{x} + \boldsymbol{D}\Delta\boldsymbol{z})\alpha \\
&\quad + \left((n+\nu)\frac{\Delta\boldsymbol{x}^T\Delta\boldsymbol{z}}{\boldsymbol{v}^T\boldsymbol{v}} + \frac{\|\boldsymbol{V}^{-1}\boldsymbol{D}^{-1}\Delta\boldsymbol{x}\|^2 + \|\boldsymbol{V}^{-1}\boldsymbol{D}\Delta\boldsymbol{z}\|^2}{2(1-\tau)}\right)\alpha^2. \qquad (16)
\end{aligned}$$

By the third equality in (2) with $\mu = \boldsymbol{v}^T\boldsymbol{v}/(n+\nu)$, we have

$$\boldsymbol{D}^{-1}\Delta\boldsymbol{x} + \boldsymbol{D}\Delta\boldsymbol{z} = -\left(\boldsymbol{v} - \frac{\boldsymbol{v}^T\boldsymbol{v}}{n+\nu}\boldsymbol{V}^{-1}\boldsymbol{e}\right).$$

From this equality, we also have, from

$$(\boldsymbol{D}^{-1}\Delta\boldsymbol{x})^T(\boldsymbol{D}\Delta\boldsymbol{z}) = \frac{1}{4}\{\|\boldsymbol{D}^{-1}\Delta\boldsymbol{x} + \boldsymbol{D}\Delta\boldsymbol{z}\|^2 - \|\boldsymbol{D}^{-1}\Delta\boldsymbol{x} - \boldsymbol{D}\Delta\boldsymbol{z}\|^2\}, \qquad (17)$$

$$\Delta\boldsymbol{x}^T\Delta\boldsymbol{z} \leq \frac{1}{4}\left\|\boldsymbol{v} - \frac{\boldsymbol{v}^T\boldsymbol{v}}{n+\nu}\boldsymbol{V}^{-1}\boldsymbol{e}\right\|^2. \qquad (18)$$

By Lemma 4, we see that

$$\begin{aligned}
\|\alpha\boldsymbol{X}^{-1}\Delta\boldsymbol{x}\| &\leq \alpha\|\boldsymbol{V}^{-1}\|\|\boldsymbol{D}^{-1}\Delta\boldsymbol{x}\| \\
&\leq \frac{\gamma_0^4\gamma_1^2 v_{\min}^2}{100(n+\nu)\boldsymbol{v}^T\boldsymbol{v}}\frac{1}{v_{\min}}\frac{5}{\gamma_0^2\gamma_1}\frac{\boldsymbol{v}^T\boldsymbol{v}}{v_{\min}} \\
&\leq 1/20, \text{ and similarly} \\
\|\alpha\boldsymbol{Z}^{-1}\Delta\boldsymbol{z}\| &\leq 1/20.
\end{aligned}$$

These inequalities imply that we have (16) for $\tau := 1/20$. Using the above results and Lemmas 3 and 4 in (16), we obtain

(19)

$$\begin{aligned}
&\phi(\boldsymbol{x}^k + \alpha\Delta\boldsymbol{x}, \boldsymbol{z}^k + \alpha\Delta\boldsymbol{z}) - \phi(\boldsymbol{x}^k, \boldsymbol{z}^k) \\
&\leq -\frac{n+\nu}{\boldsymbol{v}^T\boldsymbol{v}}\left\|\boldsymbol{v} - \frac{\boldsymbol{v}^T\boldsymbol{v}}{n+\nu}\boldsymbol{V}^{-1}\boldsymbol{e}\right\|^2\alpha \\
&\quad + \left(\frac{1}{4}\frac{n+\nu}{\boldsymbol{v}^T\boldsymbol{v}}\left\|\boldsymbol{v} - \frac{\boldsymbol{v}^T\boldsymbol{v}}{n+\nu}\boldsymbol{V}^{-1}\boldsymbol{e}\right\|^2 + \frac{10}{19}\left(\|\boldsymbol{V}^{-1}\boldsymbol{D}^{-1}\Delta\boldsymbol{x}\|^2 + \|\boldsymbol{V}^{-1}\boldsymbol{D}\Delta\boldsymbol{z}\|^2\right)\right)\alpha^2 \\
&\leq -\frac{\boldsymbol{v}^T\boldsymbol{v}}{n+\nu}\left\|\boldsymbol{V}^{-1}\boldsymbol{e} - \frac{n+\nu}{\boldsymbol{v}^T\boldsymbol{v}}\boldsymbol{v}\right\|^2\left(1 - \frac{\alpha}{4}\right)\alpha + \frac{10}{19}\|\boldsymbol{V}^{-2}\|(\|\boldsymbol{D}^{-1}\Delta\boldsymbol{x}\|^2 + \|\boldsymbol{D}\Delta\boldsymbol{z}\|^2)\alpha^2 \\
&\leq -\frac{\boldsymbol{v}^T\boldsymbol{v}}{n+\nu}\frac{3}{4v_{\min}^2}\frac{399}{400}\alpha + \frac{20}{19}\frac{1}{v_{\min}^2}\frac{25}{\gamma_0^4\gamma_1^2}\left(\frac{\boldsymbol{v}^T\boldsymbol{v}}{v_{\min}}\right)^2\alpha^2 \\
&\leq -\frac{2}{300}\frac{\gamma_0^4\gamma_1^2}{(n+\nu)^2} + \frac{1}{380}\frac{\gamma_0^4\gamma_1^2}{(n+\nu)^2} \\
&\leq -\delta.
\end{aligned}$$

Hence we have (3). The inequality (4) follows from

$$\begin{aligned}(\boldsymbol{x}^k+\alpha\boldsymbol{\Delta x})^T(\boldsymbol{z}^k+\alpha\boldsymbol{\Delta z}) &= (\boldsymbol{x}^k)^T\boldsymbol{z}^k-((\boldsymbol{x}^k)^T\boldsymbol{z}^k-n(\boldsymbol{x}^k)^T\boldsymbol{z}^k/(n+\nu))\alpha+\boldsymbol{\Delta x}^T\boldsymbol{\Delta z}\alpha^2\\ &\geq (1-\alpha)(\boldsymbol{x}^k)^T\boldsymbol{z}^k+\frac{n(\boldsymbol{x}^k)^T\boldsymbol{z}^k}{n+\nu}\alpha-\frac{25}{\gamma_0^4\gamma_1^2}\frac{((\boldsymbol{x}^k)^T\boldsymbol{z}^k)^2}{v_{min}^2}\alpha^2\\ &\geq (1-\alpha)(\boldsymbol{x}^k)^T\boldsymbol{z}^k. \qquad \square\end{aligned}$$

4. A pure potential-reduction algorithm. We now consider a potential-reduction algorithm that does not impose the explicit constraint (4) on the step size.

ALGORITHM II

Step 1. Choose $\gamma_0 \in (0,1]$ and a positive constant δ (which may depend on n and ν). Set $(\boldsymbol{x}^1,\boldsymbol{y}^1,\boldsymbol{z}^1) := \gamma_0\rho(\boldsymbol{e},\boldsymbol{0},\boldsymbol{e})$. Let $k := 1$.

Step 2. If $(\boldsymbol{x}^k)^T\boldsymbol{z}^k \leq \epsilon$ then stop.

Step 3. Let $\mu := (\boldsymbol{x}^k)^T\boldsymbol{z}^k/(n+\nu)$. Compute the unique solution $(\boldsymbol{\Delta x},\boldsymbol{\Delta y},\boldsymbol{\Delta z})$ at $(\boldsymbol{x}^k,\boldsymbol{y}^k,\boldsymbol{z}^k)$ of the system (2) of equations.

Step 4. Find a step size α such that

$$\psi(\boldsymbol{x}^k+\alpha\boldsymbol{\Delta x},\boldsymbol{y}^k+\alpha\boldsymbol{\Delta y},\boldsymbol{z}^k+\alpha\boldsymbol{\Delta z}) \leq \psi(\boldsymbol{x}^k,\boldsymbol{y}^k,\boldsymbol{z}^k)-\delta. \tag{20}$$

If we cannot find such a step size then stop.

Step 5. Let $(\boldsymbol{x}^{k+1},\boldsymbol{y}^{k+1},\boldsymbol{z}^{k+1}) := (\boldsymbol{x}^k,\boldsymbol{y}^k,\boldsymbol{z}^k)+\alpha(\boldsymbol{\Delta x},\boldsymbol{\Delta y},\boldsymbol{\Delta z})$. Increase k by 1 and go to Step 2.

The performance of this method is summarized in the following result.

THEOREM 6. *Let* $L \geq \ln n$, $\gamma_0 \in (0,1]$ *and* $\gamma_1 \in (0,1)$. *Suppose that* $\ln\rho = O(L)$, $\ln(1/\epsilon) = O(L)$, $\nu \geq \sqrt{n}$, $\sigma := \gamma_1(\boldsymbol{x}^1)^T\boldsymbol{z}^1/\|(\boldsymbol{A}\boldsymbol{x}^1-\boldsymbol{b},\boldsymbol{z}^1-\boldsymbol{c})\|$, *and* $\delta := \gamma_0^4\gamma_1^2/(300(n+\nu)^2)$. *Then Algorithm* II *terminates in* $O(\nu(n+\nu)^2L)$ *iterations. If the algorithm stops in Step* 2, *we get an approximate solution*; *otherwise* (*if it stops in Step* 4) *there are no optimal solutions* $\boldsymbol{x}^*$ *of* (P) *and* $(\boldsymbol{y}^*,\boldsymbol{z}^*)$ *of* (D) *such that* $\|(\boldsymbol{x}^*,\boldsymbol{z}^*)\|_\infty \leq \rho$.

The proof of this result is like that of Theorem 1. The lemma below shows that it will stop in the required number of iterations.

LEMMA 7. *Under the assumptions of Theorem* 6, $\psi(\boldsymbol{x}^1,\boldsymbol{y}^1,\boldsymbol{z}^1) = O(\nu L)$. *If* $\psi(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) \leq \nu\ln\epsilon$ *then* $\boldsymbol{x}^T\boldsymbol{z} \leq \epsilon$.

Proof. It follows from

$$\begin{aligned}\psi(\boldsymbol{x}^1,\boldsymbol{y}^1,\boldsymbol{z}^1) &= \nu\ln(n\gamma_0^2\rho^2)-\ln(1-\gamma_1),\\ \psi(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) &= \nu\ln(\boldsymbol{x}^T\boldsymbol{z})-\sum_{i=1}^n\ln\left(\frac{x_iz_i}{\boldsymbol{x}^T\boldsymbol{z}/n}\right)-\ln\left(1-\frac{\sigma\|(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b},\boldsymbol{A}^T\boldsymbol{y}+\boldsymbol{z}-\boldsymbol{c})\|}{\boldsymbol{x}^T\boldsymbol{z}}\right)\\ &\geq \nu\ln(\boldsymbol{x}^T\boldsymbol{z}). \qquad \square\end{aligned}$$

To complete the proof of the theorem, we need to show that if there are optimal solutions $\boldsymbol{x}^*$ of (P) and $(\boldsymbol{y}^*,\boldsymbol{z}^*)$ of (D) such that $\|(\boldsymbol{x}^*,\boldsymbol{z}^*)\|_\infty \leq \rho$, then there exists a step size α which satisfies (20) for $\delta = \gamma_0^4\gamma_1^2/(300(n+\nu)^2)$. We use Lemma 5. Note that (7) holds automatically since $\psi(\boldsymbol{x}^k,\boldsymbol{y}^k,\boldsymbol{z}^k)$ is finite, using (6) and the definition of σ. Hence we only need the following result.

LEMMA 8. *If*

$$\begin{aligned}\phi(\boldsymbol{x}^k+\alpha\boldsymbol{\Delta x},\boldsymbol{z}^k+\alpha\boldsymbol{\Delta z}) &\leq \phi(\boldsymbol{x}^k,\boldsymbol{z}^k)-\delta,\\ (\boldsymbol{x}^k+\alpha\boldsymbol{\Delta x})^T(\boldsymbol{z}^k+\alpha\boldsymbol{\Delta z}) &\geq (1-\alpha)(\boldsymbol{x}^k)^T\boldsymbol{z}^k,\end{aligned}$$

then

$$\psi(\boldsymbol{x}^k + \alpha\Delta\boldsymbol{x}, \boldsymbol{y}^k + \alpha\Delta\boldsymbol{y}, \boldsymbol{z}^k + \alpha\Delta\boldsymbol{z}) \leq \psi(\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{z}^k) - \delta.$$

Proof. By (2) we have that

$$(\boldsymbol{A}(\boldsymbol{x}^k+\alpha\Delta\boldsymbol{x})-\boldsymbol{b}, \boldsymbol{A}^T(\boldsymbol{y}^k+\alpha\Delta\boldsymbol{y})+(\boldsymbol{z}^k+\alpha\Delta\boldsymbol{z})-\boldsymbol{c}) = (1-\alpha)(\boldsymbol{A}\boldsymbol{x}^k-\boldsymbol{b}, \boldsymbol{A}^T\boldsymbol{y}^k+\boldsymbol{z}^k-\boldsymbol{c}).$$

Thus we get

$$\begin{aligned}
&\psi(\boldsymbol{x}^k + \alpha\Delta\boldsymbol{x}, \boldsymbol{y}^k + \alpha\Delta\boldsymbol{y}, \boldsymbol{z}^k + \alpha\Delta\boldsymbol{z}) \\
&= \phi(\boldsymbol{x}^k + \alpha\Delta\boldsymbol{x}, \boldsymbol{z}^k + \alpha\Delta\boldsymbol{z}) - \ln\left(1 - \frac{(1-\alpha)\sigma\|(\boldsymbol{A}\boldsymbol{x}^k - \boldsymbol{b}, \boldsymbol{A}^T\boldsymbol{y}^k + \boldsymbol{z}^k - \boldsymbol{c})\|}{(\boldsymbol{x}^k + \alpha\Delta\boldsymbol{x})^T(\boldsymbol{z}^k + \alpha\Delta\boldsymbol{z})}\right) \\
&\leq \phi(\boldsymbol{x}^k, \boldsymbol{z}^k) - \delta - \ln\left(1 - \frac{\sigma\|(\boldsymbol{A}\boldsymbol{x}^k - \boldsymbol{b}, \boldsymbol{A}^T\boldsymbol{y}^k + \boldsymbol{z}^k - \boldsymbol{c})\|}{(\boldsymbol{x}^k)^T\boldsymbol{z}^k}\right) \\
&= \psi(\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{z}^k) - \delta. \qquad \square
\end{aligned}$$

5. An $O(nL)$-iteration variant. Algorithms I and II require $O(n^{2.5}L)$ iterations. Mizuno [12] proposed an $O(nL)$-iteration variant of the infeasible-interior-point path following algorithm. We can also construct $O(nL)$-iteration variants of Algorithms I and II. In this section, we only show the variant of Algorithm II. Although the $O(nL)$-iteration variant in [12] generates a sequence of infeasible interior points in a neighborhood of the path of centers, our variant does not confine the sequence to such a neighborhood.

ALGORITHM III

Step 1. Choose γ_0 and λ in $(0, 1]$ and positive constants δ_1 and δ_2. Set $(\boldsymbol{x}^1, \boldsymbol{y}^1, \boldsymbol{z}^1) := \gamma_0\rho(\boldsymbol{e}, \boldsymbol{0}, \boldsymbol{e})$. Let $k := 1$.

Step 2. If $(\boldsymbol{x}^k)^T\boldsymbol{z}^k \leq \epsilon$ then stop.

Step 3. If

$$v_{min}^2 := \min_i x_i^k z_i^k \geq \lambda(\boldsymbol{x}^k)^T\boldsymbol{z}^k/n \tag{21}$$

then

Step A. Let $\mu := (\boldsymbol{x}^k)^T\boldsymbol{z}^k/(n+\nu)$. Compute the unique solution $(\Delta\boldsymbol{x}, \Delta\boldsymbol{y}, \Delta\boldsymbol{z})$ at $(\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{z}^k)$ of the system (2) of equations. Find a step size α such that

$$\psi(\boldsymbol{x}^k + \alpha\Delta\boldsymbol{x}, \boldsymbol{y}^k + \alpha\Delta\boldsymbol{y}, \boldsymbol{z}^k + \alpha\Delta\boldsymbol{z}) \leq \psi(\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{z}^k) - \delta_1. \tag{22}$$

If we cannot find such a step size then stop.

else

Step B. Compute the unique solution $(\Delta\boldsymbol{x}, \Delta\boldsymbol{y}, \Delta\boldsymbol{z})$ of the system of equations

$$\begin{pmatrix} \boldsymbol{A} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}^T & \boldsymbol{I} \\ \boldsymbol{Z}^k & \boldsymbol{0} & \boldsymbol{X}^k \end{pmatrix} \begin{pmatrix} \Delta\boldsymbol{x} \\ \Delta\boldsymbol{y} \\ \Delta\boldsymbol{z} \end{pmatrix} = - \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{X}^k\boldsymbol{z}^k - ((\boldsymbol{x}^k)^T\boldsymbol{z}^k/n)\boldsymbol{e} \end{pmatrix}. \tag{23}$$

Find a step size α such that

$$\psi(\boldsymbol{x}^k + \alpha\Delta\boldsymbol{x}, \boldsymbol{y}^k + \alpha\Delta\boldsymbol{y}, \boldsymbol{z}^k + \alpha\Delta\boldsymbol{z}) \leq \psi(\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{z}^k) - \delta_2. \tag{24}$$

Step 4. Let $(\boldsymbol{x}^{k+1}, \boldsymbol{y}^{k+1}, \boldsymbol{z}^{k+1}) := (\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{z}^k) + \alpha(\boldsymbol{\Delta x}, \boldsymbol{\Delta y}, \boldsymbol{\Delta z})$. Increase k by 1 and go to Step 2.

Note that $(\boldsymbol{\Delta x}, \boldsymbol{\Delta y}, \boldsymbol{\Delta z})$ in Step B is a centering step as in [14]; because of the zeros in the right-hand side vector in (23), this step maintains the current infeasibilities as well as the "duality gap" $\boldsymbol{x}^T\boldsymbol{z}$.

The performance of this method is summarized in the following result.

THEOREM 9. *Let* $L \geq \ln n$, $\gamma_0 \in (0,1]$ *and* γ_1, $\lambda \in (0,1)$. *Suppose that* $\ln \rho = O(L)$, $\ln(1/\epsilon) = O(L)$, $\nu \geq n$, $\sigma := \gamma_1(\boldsymbol{x}^1)^T\boldsymbol{z}^1/\|(\boldsymbol{A}\boldsymbol{x}^1 - \boldsymbol{b}, \boldsymbol{z}^1 - \boldsymbol{c})\|$, $\delta_1 := .001\lambda^2\gamma_0^4\gamma_1^2$, *and* $\delta_2 := (1-\lambda)^2/4$. *Then Algorithm* III *terminates in* $O(\nu L)$ *iterations in Step* 2 *or* A. *If the algorithm stops in Step* 2, *we get an approximate solution*; *otherwise* (*if it stops in Step* A) *there are no optimal solutions* $\boldsymbol{x}^*$ *of* (P) *and* $(\boldsymbol{y}^*, \boldsymbol{z}^*)$ *of* (D) *such that* $\|(\boldsymbol{x}^*, \boldsymbol{z}^*)\|_\infty \leq \rho$.

Since δ_1 and δ_2 are constants independent of the input data, the number of iterations is bounded by $O(\nu L)$ (see Lemma 7). As shown in §3, we can get an approximate solution if the algorithm stops in Step 2. To complete the proof, we need to show that

(i) if (21) holds, there is a step size α that satisfies (22), or there are no optimal solutions $\boldsymbol{x}^*$ of (P) and $(\boldsymbol{y}^*, \boldsymbol{z}^*)$ of (D) such that $\|(\boldsymbol{x}^*, \boldsymbol{z}^*)\|_\infty \leq \rho$,

(ii) if (21) does not hold, there is a step size α which satisfies (24).

From Lemmas 8 and 11 below (i) follows. Lemma 10 is used in the proof of Lemma 11. From Lemma 12 (ii) follows.

LEMMA 10. *For any n-vector* $\boldsymbol{v} > \boldsymbol{0}$ *and any* $\nu \geq 0$,

$$\left\| \boldsymbol{V}^{-1}\boldsymbol{e} - \frac{n+\nu}{\boldsymbol{v}^T\boldsymbol{v}}\boldsymbol{v} \right\|^2 \geq \frac{\nu^2}{\boldsymbol{v}^T\boldsymbol{v}}.$$

Proof. It follows from

$$\begin{aligned} \|\boldsymbol{V}^{-1}\boldsymbol{e} - \frac{n+\nu}{\boldsymbol{v}^T\boldsymbol{v}}\boldsymbol{v}\|^2 &= \|\boldsymbol{V}^{-1}\boldsymbol{e} - \frac{n}{\boldsymbol{v}^T\boldsymbol{v}}\boldsymbol{v}\|^2 + \|\frac{\nu}{\boldsymbol{v}^T\boldsymbol{v}}\boldsymbol{v}\|^2 \\ &\geq \nu^2/\boldsymbol{v}^T\boldsymbol{v}. \qquad \square \end{aligned}$$

LEMMA 11. *Let* $\nu \geq n$, $\gamma_0 \in (0,1]$, $\gamma_1 \in (0,1]$, *and* $\delta_1 := .001\lambda^2\gamma_0^4\gamma_1^2$, *and suppose* (7) *and* (21) *hold. If there exist optimal solutions* $\boldsymbol{x}^*$ *of* (P) *and* $(\boldsymbol{y}^*, \boldsymbol{z}^*)$ *of* (D) *such that* $\|(\boldsymbol{x}^*, \boldsymbol{z}^*)\|_\infty \leq \rho$, *then we have* (3) *and* (4) *for*

$$\alpha := \frac{\lambda^2\gamma_0^4\gamma_1^2}{100(n+\nu)}$$

at the kth iteration.

Proof. As in the proof of Lemma 5, we have (19) for $\alpha = \lambda^2\gamma_0^4\gamma_1^2/100(n+\nu)$. Using Lemma 10, $\nu \geq n$, and $v_{\min}^2 \geq \lambda\boldsymbol{v}^T\boldsymbol{v}/n$, we see that

$$\begin{aligned} &\phi(\boldsymbol{x}^k + \alpha\boldsymbol{\Delta x}, \boldsymbol{z}^k + \alpha\boldsymbol{\Delta z}) - \phi(\boldsymbol{x}^k, \boldsymbol{z}^k) \\ &\leq -\frac{\boldsymbol{v}^T\boldsymbol{v}}{n+\nu}\frac{\nu^2}{\boldsymbol{v}^T\boldsymbol{v}}\left(1 - \frac{\alpha}{4}\right)\alpha + \frac{10}{19}\|\boldsymbol{V}^{-2}\|(\|\boldsymbol{D}^{-1}\boldsymbol{\Delta x}\|^2 + \|\boldsymbol{D}\boldsymbol{\Delta z}\|^2)\alpha^2 \\ &\leq -\frac{\nu^2}{n+\nu}\frac{399}{400}\alpha + \frac{20}{19}\frac{1}{v_{\min}^2}\frac{25}{\gamma_0^4\gamma_1^2}\left(\frac{\boldsymbol{v}^T\boldsymbol{v}}{v_{\min}}\right)^2\alpha^2 \\ &\leq -.9\frac{\nu^2}{n+\nu}\alpha + \frac{20}{19}\frac{25}{\gamma_0^4\gamma_1^2}\left(\frac{n}{\lambda}\right)^2\alpha^2 \end{aligned}$$

$$\leq -.009\lambda^2\gamma_0^4\gamma_1^2\frac{\nu^2}{(n+\nu)^2} + \frac{1}{380}\lambda^2\gamma_0^4\gamma_1^2\frac{n^2}{(n+\nu)^2}$$
$$\leq -\delta_1.$$

Inequality (4) follows from the same analysis as in the proof of Lemma 5. □

LEMMA 12 (Theorem 6 in Mizuno and Nagasawa [13]). *Let* $\nu \geq 0$ *and* $\delta_2 := (1-\lambda)^2/4$, *and suppose that* (21) *does not hold. Then we have* (24) *for*

$$\alpha := \frac{\lambda_k}{1+\sqrt{\lambda_k \rho_k}}$$

at the kth iteration, where

$$\lambda_k := \min_i \frac{x_i^k z_i^k}{(\boldsymbol{x}^k)^T \boldsymbol{z}^k/n} \quad \text{and} \quad \rho_k := \sum_{i=1}^n \left(\frac{(\boldsymbol{x}^k)^T \boldsymbol{z}^k/n}{x_i^k z_i^k} - 1 \right).$$

The result above is proved in [13] for the potential function ϕ with $\nu = \sqrt{n}$, but it is valid for the potential function ψ with any $\nu \geq 0$ since the duality gap and infeasibility do not change in Step B.

6. Concluding remarks. In this final section we contrast Algorithms I and II, discuss the results obtained, and briefly consider other possible primal-dual potential functions for the infeasible case. We also describe an extension to monotone linear complementarity problems.

Remark A. *Getting information on infeasibility.* We note that γ_0 and ρ appear in the algorithms only through their product and the dependence of δ on γ_0. Suppose that we start Algorithm I, II, or III with $(\boldsymbol{x}^1, \boldsymbol{y}^1, \boldsymbol{z}^1) = \rho_0(\boldsymbol{e}, \boldsymbol{0}, \boldsymbol{e})$ for some $\rho_0 > 0$, and that at each iteration we perform a line search to achieve the largest decrease in ϕ subject to satisfying (4) (largest decrease in ψ).

If there are optimal solutions $\boldsymbol{x}^*$ of (P) and $(\boldsymbol{y}^*, \boldsymbol{z}^*)$ of (D) such that $\|(\boldsymbol{x}^*, \boldsymbol{z}^*)\|_\infty \leq \rho_0/\gamma_0$ for some $\gamma_0 \in (0,1]$, it follows from the inequality (9) in the proof of Lemma 4 that

$$\theta^k \rho_0 \|(\boldsymbol{x}^k, \boldsymbol{z}^k)\|_1 \leq \frac{2(\boldsymbol{x}^k)^T \boldsymbol{z}^k}{\gamma_0 \gamma_1}.$$

Hence if this inequality is violated, then we can conclude that there are no optimal solutions $\boldsymbol{x}^*$ of (P) and $(\boldsymbol{y}^*, \boldsymbol{z}^*)$ of (D) such that $\|(\boldsymbol{x}^*, \boldsymbol{z}^*)\|_\infty \leq \rho_0/\gamma_0$. Here γ_0 can be varied during one run of the algorithm.

Remark B. *Comparison between Algorithms* I *and* II. In Algorithm I, we put an explicit bound on α via (4) to ensure that

$$(\boldsymbol{x}^k)^T \boldsymbol{z}^k \geq \frac{\|(\boldsymbol{A}\boldsymbol{x}^k - \boldsymbol{b}, \boldsymbol{A}^T\boldsymbol{y}^k + \boldsymbol{z}^k - \boldsymbol{c})\|}{\|(\boldsymbol{A}\boldsymbol{x}^1 - \boldsymbol{b}, \boldsymbol{A}^T\boldsymbol{y}^1 + \boldsymbol{z}^1 - \boldsymbol{c})\|} (\boldsymbol{x}^1)^T \boldsymbol{z}^1 \tag{25}$$

for all k. Inequalities of this kind were first used by Kojima, Megiddo, and Mizuno [3]. In fact, (4) can be relaxed as long as (25) holds at each iteration; if (4) held strictly at some previous iteration, (25) may hold even if (4) does not. Algorithms II and III dispense with this explicit constraint by adding a barrier term to ϕ. If σ is as in Theorem 6, then ψ is only finite if

$$(\boldsymbol{x}^k)^T \boldsymbol{z}^k \geq \gamma_1 \frac{\|(\boldsymbol{A}\boldsymbol{x}^k - \boldsymbol{b}, \boldsymbol{A}^T\boldsymbol{y}^k + \boldsymbol{z}^k - \boldsymbol{c})\|}{\|(\boldsymbol{A}\boldsymbol{x}^1 - \boldsymbol{b}, \boldsymbol{A}^T\boldsymbol{y}^1 + \boldsymbol{z}^1 - \boldsymbol{c})\|} (\boldsymbol{x}^1)^T \boldsymbol{z}^1. \tag{26}$$

For $\gamma_1 < 1$, this is a weaker condition than (25), but the complexity bounds include a factor γ_1^{-2}.

In contrast, the homogeneous self-dual infeasible-interior-point algorithm of Ye, Todd, and Mizuno [21] maintains

$$\boldsymbol{c}^T\boldsymbol{x}^k - \boldsymbol{b}^T\boldsymbol{y}^k \le \frac{\|(\boldsymbol{A}\boldsymbol{x}^k - \boldsymbol{b}\tau^k)\|}{\|(\boldsymbol{A}\boldsymbol{x}^1 - \boldsymbol{b})\|}(\boldsymbol{c}^T\boldsymbol{x}^1 - \boldsymbol{b}^T\boldsymbol{y}^1 + 1)$$

in the present notation. Here, τ^k is the value of the homogenizing variable at the kth iteration; $\tau^1 = 1$. Hence in [21] the "duality gap" decreases *faster* than the infeasibility, whereas in the current paper as well as in [3], [22], and [12], the "total complementarity" decreases *at most as fast* as the infeasibility. (Note that, with infeasible iterates, $\boldsymbol{c}^T\boldsymbol{x}^k - \boldsymbol{b}^T\boldsymbol{y}^k$ may not equal $(\boldsymbol{x}^k)^T\boldsymbol{z}^k$, and may even be negative.)

Remark C. *Complexity bounds.*

Our bound on the number of iterations for Algorithms I and II is $O(n^{2.5}L)$ (when $\nu = \sqrt{n}$), while Zhang [22] and Mizuno [12] obtain $O(n^2L)$ (Mizuno has a variant with $O(nL)$) and Potra [17] achieves $O(n^{1.5}L)$ (the revised version has $O(nL)$); in contrast, feasible-interior-point algorithms typically have bounds of $O(n^{.5}L)$ iterations [5], [6], [15], [16]. Let us examine why the complexity is so much larger in our case, and why it decreases for Algorithm III.

Since the analysis for Algorithm II is based on that for Algorithm I, we consider only the latter. We also assume $\nu \le 3n$. Using the arguments of Lemmas 2 and 5, we have for any $0 < \alpha < 1$ satisfying

$$\|\alpha(\boldsymbol{X}^k)^{-1}\boldsymbol{\Delta x}\| \le \tau, \quad \|\alpha(\boldsymbol{Z}^k)^{-1}\boldsymbol{\Delta z}\| \le \tau,$$

for some $\tau \in (0,1)$,

$$\begin{aligned}\Delta\phi &:= \phi(\boldsymbol{x}^k + \alpha\boldsymbol{\Delta x}, \boldsymbol{z}^k + \alpha\boldsymbol{\Delta z}) - \phi(\boldsymbol{x}^k, \boldsymbol{z}^k)\\ &\le -\frac{\boldsymbol{v}^T\boldsymbol{v}}{n+\nu}\|\boldsymbol{V}^{-1}\boldsymbol{e} - \frac{n+\nu}{\boldsymbol{v}^T\boldsymbol{v}}\boldsymbol{v}\|^2\frac{3}{4}\alpha\\ &\quad + \frac{1}{2(1-\tau)}\|\boldsymbol{V}^{-2}\|(\|\boldsymbol{D}^{-1}\boldsymbol{\Delta x}\|^2 + \|\boldsymbol{D}\boldsymbol{\Delta z}\|^2)\alpha^2 \qquad (27)\end{aligned}$$

(cf. (19)). In our analysis, we bounded the second-order term above using Lemma 4; also using Lemma 3 to bound the first-order term, we get

$$\begin{aligned}\Delta\phi &\le -\frac{3}{4}\left(\frac{\sqrt{3}}{2}\right)^2\frac{\boldsymbol{v}^T\boldsymbol{v}}{n+\nu}v_{\min}^{-2}\alpha\\ &\quad + \frac{1}{2(1-\tau)}v_{\min}^{-2}\frac{50}{\gamma_0^4\gamma_1^2}\frac{(\boldsymbol{v}^T\boldsymbol{v})^2}{v_{\min}^2}\alpha^2. \qquad (28)\end{aligned}$$

The linear term in (28) would allow a constant decrease in ϕ by choosing a constant α, but unfortunately the quadratic term is much too large. Indeed, the right-hand side of (28) is minimized by

$$\alpha = \frac{9(1-\tau)}{800}\frac{\gamma_0^4\gamma_1^2 v_{\min}^2 n}{\boldsymbol{v}^T\boldsymbol{v}}\frac{1}{n(n+\nu)}.$$

Notice that $nv_{\min}^2 = \boldsymbol{v}^T\boldsymbol{v}$ for $\boldsymbol{v}$ a multiple of $\boldsymbol{e}$, i.e., when $(\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{z}^k)$ is on the central path, and then $\alpha = O(n^{-2})$ and hence $\Delta\phi = O(n^{-2})$. Our choice for α in Lemma 5 approximates this "optimal" value.

Now let us see how this analysis changes when the iterate is feasible. Note that

$$\begin{aligned}\|\boldsymbol{D}^{-1}\boldsymbol{\Delta x}\|^2 + \|\boldsymbol{D}\boldsymbol{\Delta z}\|^2 &= \|\boldsymbol{D}^{-1}\boldsymbol{\Delta x} + \boldsymbol{D}\boldsymbol{\Delta z}\|^2 - 2\boldsymbol{\Delta x}^T\boldsymbol{\Delta z} \\ &= \left(\frac{\boldsymbol{v}^T\boldsymbol{v}}{n+\nu}\right)^2 \|\boldsymbol{V}^{-1}\boldsymbol{e} - \frac{n+\nu}{\boldsymbol{v}^T\boldsymbol{v}}\boldsymbol{v}\|^2 - 2\boldsymbol{\Delta x}^T\boldsymbol{\Delta z}. \end{aligned} \tag{29}$$

The last equality follows from the final equation of (2). In the feasible case, $\boldsymbol{\Delta x}^T\boldsymbol{\Delta z}$ is zero, and, for

$$\|\boldsymbol{V}^{-1}\boldsymbol{e} - \frac{n+\nu}{\boldsymbol{v}^T\boldsymbol{v}}\boldsymbol{v}\| = O(v_{\min}^{-1}),$$

the resulting second-order term is smaller than in (28) by a factor of about n^2. We can thus choose a much larger value for α; indeed,

$$\alpha := \frac{\tau(n+\nu)v_{\min}}{\boldsymbol{v}^T\boldsymbol{v}\|\boldsymbol{V}^{-1}\boldsymbol{e} - \frac{n+\nu}{\boldsymbol{v}^T\boldsymbol{v}}\boldsymbol{v}\|}$$

for $\tau := \sqrt{3}/9$ satisfies all our requirements and yields $\Delta\phi \leq -1/16$.

The equations in (29) show that $\|\boldsymbol{D}^{-1}\boldsymbol{\Delta x}\|^2 + \|\boldsymbol{D}\boldsymbol{\Delta z}\|^2$ is much larger than $\|\boldsymbol{D}^{-1}\boldsymbol{\Delta x} + \boldsymbol{D}\boldsymbol{\Delta z}\|^2$ when $\boldsymbol{\Delta x}^T\boldsymbol{\Delta z}$ is large and negative. (It cannot be large and positive by (18), but this does not help us; (17) provides no lower bound.) But from (11) and (15), we obtain

$$\begin{aligned}\boldsymbol{\Delta x}^T\boldsymbol{\Delta z} &= (\boldsymbol{D}^{-1}\boldsymbol{\Delta x})^T(\boldsymbol{D}\boldsymbol{\Delta z}) \\ &= -(\theta^k)^2(\boldsymbol{x}^1 - \boldsymbol{x}^*)^T\boldsymbol{D}^{-1}\boldsymbol{Q}\boldsymbol{D}^{-1}(\boldsymbol{x}^1 - \boldsymbol{x}^*) \\ &\quad -(\theta^k)^2(\boldsymbol{z}^1 - \boldsymbol{z}^*)^T\boldsymbol{D}(\boldsymbol{I} - \boldsymbol{Q})\boldsymbol{D}(\boldsymbol{z}^1 - \boldsymbol{z}^*) \\ &\quad +\theta^k(\boldsymbol{x}^1 - \boldsymbol{x}^*)^T\boldsymbol{D}^{-1}\boldsymbol{Q}(\boldsymbol{v} - \mu\boldsymbol{V}^{-1}\boldsymbol{e}) + \theta^k(\boldsymbol{z}^1 - \boldsymbol{z}^*)^T\boldsymbol{D}(\boldsymbol{I} - \boldsymbol{Q})(\boldsymbol{v} - \mu\boldsymbol{V}^{-1}\boldsymbol{e}).\end{aligned}$$

The first two terms are negative, while the last two are of indeterminate sign. It is not hard to see that

$$\begin{aligned}(\theta^k)^2(\boldsymbol{x}^1 - \boldsymbol{x}^*)^T\boldsymbol{D}^{-1}\boldsymbol{Q}\boldsymbol{D}^{-1}(\boldsymbol{x}^1 - \boldsymbol{x}^*) &= (\boldsymbol{x}^k - \boldsymbol{x}^*)^T\boldsymbol{D}^{-1}\boldsymbol{Q}\boldsymbol{D}^{-1}(\boldsymbol{x}^k - \boldsymbol{x}^*) \\ &= \|\boldsymbol{Q}\boldsymbol{D}^{-1}(\boldsymbol{x}^k - \boldsymbol{x}^*)\|^2,\end{aligned}$$

and this can be seen to be the square of the distance from $\boldsymbol{D}^{-1}\boldsymbol{x}^k$ to affine set $\{\boldsymbol{x} : \boldsymbol{A}\boldsymbol{D}\boldsymbol{x} = \boldsymbol{b}\}$. Similarly,

$$\begin{aligned}(\theta^k)^2(\boldsymbol{z}^1 - \boldsymbol{z}^*)^T\boldsymbol{D}(\boldsymbol{I} - \boldsymbol{Q})\boldsymbol{D}(\boldsymbol{z}^1 - \boldsymbol{z}^*) &= (\boldsymbol{z}^k - \boldsymbol{z}^*)^T\boldsymbol{D}(\boldsymbol{I} - \boldsymbol{Q})\boldsymbol{D}(\boldsymbol{z}^k - \boldsymbol{z}^*) \\ &= \|(\boldsymbol{I} - \boldsymbol{Q})\boldsymbol{D}(\boldsymbol{z}^k - \boldsymbol{z}^*)\|^2\end{aligned}$$

is the square of the distance from $\boldsymbol{D}\boldsymbol{z}^k$ to affine set $\{\boldsymbol{z} : \boldsymbol{D}\boldsymbol{A}^T\boldsymbol{y} + \boldsymbol{z} = \boldsymbol{D}\boldsymbol{c}$ for some $\boldsymbol{y}\}$. The last two terms are bounded by $\|\boldsymbol{Q}\boldsymbol{D}^{-1}(\boldsymbol{x}^k - \boldsymbol{x}^*)\|\|\boldsymbol{v} - \mu\boldsymbol{V}^{-1}\boldsymbol{e}\|$ and $\|(\boldsymbol{I} - \boldsymbol{Q})\boldsymbol{D}(\boldsymbol{z}^k - \boldsymbol{z}^*)\|\|\boldsymbol{v} - \mu\boldsymbol{V}^{-1}\boldsymbol{e}\|$. If $\boldsymbol{v}$ is a multiple of $\boldsymbol{e}$ ($(\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{z}^k)$ is on the central path), then $\|\boldsymbol{v} - \mu\boldsymbol{V}^{-1}\boldsymbol{e}\|$ is bounded by $\sqrt{(\boldsymbol{x}^k)^T\boldsymbol{z}^k}$. Hence if the infeasibility is large compared to the duality gap, $\boldsymbol{\Delta x}^T\boldsymbol{\Delta z}$ will be large and negative, the second-order term in (27) will be large, and only a small decrease in ϕ can be guaranteed. This also explains why we need to carefully balance the infeasibility and the duality gap, as in (25)–(26).

We remark that, if the 2-norm of $(\boldsymbol{x}^k, \boldsymbol{z}^k)$ is much smaller than its 1-norm (as when, for instance, it is close to a multiple of $(\boldsymbol{e}, \boldsymbol{e})$), then the first inequality in (13),

and hence the bound (14) on $\|\boldsymbol{D}^{-1}\boldsymbol{\Delta x}\|$ and similarly that on $\|\boldsymbol{D}\boldsymbol{\Delta z}\|$, could be improved by a factor close to $\sqrt{n}$. Then α could be chosen to give a reduction in ϕ of order n^{-1} rather than n^{-2}.

Finally, Lemma 10 allows us to obtain a better bound on the first term in (27) when the current iterate is approximately centered, and in this case the second-order term is smaller and thus a greater decrease in ϕ (and ψ) can be achieved by choosing a larger value for α. This is the basis for Step A in Algorithm III. Lemma 12 proves that a constant decrease in ϕ (and ψ) can also be achieved when the current iterate is far from centered, by using a simple centering step.

Remark D. *Some other potential functions.* There are two other primal-dual potential functions that could be used in the infeasible case. The first is

$$\psi'(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) := (n+\nu)\ln(\boldsymbol{x}^T\boldsymbol{z} + \kappa\|(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}, \boldsymbol{A}^T\boldsymbol{y}+\boldsymbol{z}-\boldsymbol{c})\|) - \sum_{i=1}^{n}\ln(x_i z_i) - n\ln n,$$

and was suggested for a pure potential-reduction method by Kojima, Noma, and Yoshise [7] in the context of the monotone complementarity problem. The second is

$$\psi''(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) := (n+\nu)\ln(\max\{\boldsymbol{x}^T\boldsymbol{z}, \kappa\|(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}, \boldsymbol{A}^T\boldsymbol{y}+\boldsymbol{z}-\boldsymbol{c})\|\}) - \sum_{i=1}^{n}\ln(x_i z_i) - n\ln n.$$

Both can be reduced by an amount sufficient to establish a polynomial time bound if we add a restriction like (4) on the step size, so that (25) holds for all k. However, in this case there seems to be no reason to choose the more complicated functions over the simpler ϕ. If we relax the constraint, Kojima, Noma, and Yoshise [7] show that ψ' can always be reduced by some amount, but provide no bound (indeed, it seems hard to do so, even in the case of linear programming). Similar difficulties arise with ψ''. It seems to be very hard to obtain a guaranteed decrease in such a potential function when the duality gap $\boldsymbol{x}^T\boldsymbol{z}$ is much smaller than the infeasibility $\|(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}, \boldsymbol{A}^T\boldsymbol{y}+\boldsymbol{z}-\boldsymbol{c})\|$. We also mention a modified primal-dual potential function given by Kaliski and Ye [1] for a monotone linear complementarity problem with a restriction that some prescribed variables are zero. Their algorithm with the use of the modified potential function solves a combined Phase I-Phase II primal-dual linear program in $O(nL)$ iterations.

Remark E. *Extension to monotone linear complementarity problems.* Consider a linear complementarity problem with a positive semi-definite matrix $\boldsymbol{M}$ and a vector $\boldsymbol{q}$: Find a pair $(\boldsymbol{x},\boldsymbol{z}) \geq \boldsymbol{0}$ such that $\boldsymbol{z} = \boldsymbol{M}\boldsymbol{x} + \boldsymbol{q}$ and $\boldsymbol{x}^T\boldsymbol{z} = \boldsymbol{0}$. We can easily adapt Algorithms I, II, and III to the problem. Major changes are as follows.

- Eliminate $\boldsymbol{y}^1$, $\boldsymbol{y}^k$, $\boldsymbol{y}^{k+1}$ and $\boldsymbol{\Delta y}$.
- Replace the system (2) of equations by

$$\begin{pmatrix} -\boldsymbol{M} & \boldsymbol{I} \\ \boldsymbol{Z}^k & \boldsymbol{X}^k \end{pmatrix}\begin{pmatrix} \boldsymbol{\Delta x} \\ \boldsymbol{\Delta z} \end{pmatrix} = -\begin{pmatrix} \boldsymbol{z}^k - \boldsymbol{M}\boldsymbol{x}^k - \boldsymbol{q} \\ \boldsymbol{X}^k\boldsymbol{z}^k - \mu\boldsymbol{e} \end{pmatrix}.$$

- Replace the system (23) of equations by

$$\begin{pmatrix} -\boldsymbol{M} & \boldsymbol{I} \\ \boldsymbol{Z}^k & \boldsymbol{X}^k \end{pmatrix}\begin{pmatrix} \boldsymbol{\Delta x} \\ \boldsymbol{\Delta z} \end{pmatrix} = -\begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{X}^k\boldsymbol{z}^k - ((\boldsymbol{x}^k)^T\boldsymbol{z}^k/n)\boldsymbol{e} \end{pmatrix}.$$

- Replace the potential function ψ by

$$\psi(\boldsymbol{x},\boldsymbol{z}) := (n+\nu+1)\ln(\boldsymbol{x}^T\boldsymbol{z}) - \sum_{i=1}^{n}\ln(x_i z_i) - n\ln n \\ - \ln(\boldsymbol{x}^T\boldsymbol{z} - \sigma\|\boldsymbol{z} - \boldsymbol{M}\boldsymbol{x} - \boldsymbol{q}\|).$$

Then we have results similar to Theorems 1, 6, and 9, whose proofs are basically the same as in the linear programming case except for Lemma 4. In the proof of Lemma 4, we have

$$(\theta^k\boldsymbol{x}^1 + (1-\theta^k)\boldsymbol{x}^* - \boldsymbol{x}^k)^T(\theta^k\boldsymbol{z}^1 + (1-\theta^k)\boldsymbol{z}^* - \boldsymbol{z}^k) \geq 0,$$

$$\begin{pmatrix} -\boldsymbol{M} & \boldsymbol{I} \\ \boldsymbol{Z}^k & \boldsymbol{X}^k \end{pmatrix} \begin{pmatrix} \Delta\boldsymbol{x} + \theta^k(\boldsymbol{x}^1 - \boldsymbol{x}^*) \\ \Delta\boldsymbol{z} + \theta^k(\boldsymbol{z}^1 - \boldsymbol{z}^*) \end{pmatrix} \\ = -\begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{X}^k\boldsymbol{z}^k - \mu\boldsymbol{e} - \theta^k\boldsymbol{Z}^k(\boldsymbol{x}^1 - \boldsymbol{x}^*) - \theta^k\boldsymbol{X}^k(\boldsymbol{z}^1 - \boldsymbol{z}^*) \end{pmatrix},$$

instead of (8) and (10), respectively. It is well known in interior-point methods for LCP (and easy to show) that for any n-dimensional vector $\boldsymbol{p}$, the solution of the system

$$\begin{pmatrix} -\boldsymbol{M} & \boldsymbol{I} \\ \boldsymbol{Z}^k & \boldsymbol{X}^k \end{pmatrix} \begin{pmatrix} \Delta\boldsymbol{x}' \\ \Delta\boldsymbol{z}' \end{pmatrix} = \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{p} \end{pmatrix}$$

satisfies

$$\|\boldsymbol{D}^{-1}\Delta\boldsymbol{x}'\| = \|\boldsymbol{D}(\Delta\boldsymbol{z}' - (\boldsymbol{X}^k)^{-1}\boldsymbol{p})\| \leq \|(\boldsymbol{X}^k\boldsymbol{Z}^k)^{-1/2}\boldsymbol{p}\|, \\ \|\boldsymbol{D}\Delta\boldsymbol{z}'\| = \|\boldsymbol{D}^{-1}(\Delta\boldsymbol{x}' - (\boldsymbol{Z}^k)^{-1}\boldsymbol{p})\| \leq \|(\boldsymbol{X}^k\boldsymbol{Z}^k)^{-1/2}\boldsymbol{p}\|,$$

where $\boldsymbol{D} = (\boldsymbol{X}^k)^{1/2}(\boldsymbol{Z}^k)^{-1/2}$. Let $(\Delta\boldsymbol{x}'_1, \Delta\boldsymbol{z}'_1)$, $(\Delta\boldsymbol{x}'_2, \Delta\boldsymbol{z}'_2)$, and $(\Delta\boldsymbol{x}'_3, \Delta\boldsymbol{z}'_3)$ be the solution of the system above when $\boldsymbol{p}$ is $-(\boldsymbol{X}^k\boldsymbol{z}^k - \mu\boldsymbol{e})$, $\theta^k\boldsymbol{Z}^k(\boldsymbol{x}^1 - \boldsymbol{x}^*)$, and $\theta^k\boldsymbol{X}^k(\boldsymbol{z}^1 - \boldsymbol{z}^*)$ respectively. Then we have

$$\begin{aligned} \|\boldsymbol{D}^{-1}\Delta\boldsymbol{x}\| &= \|\boldsymbol{D}^{-1}(\Delta\boldsymbol{x}'_1 + \Delta\boldsymbol{x}'_2 + \Delta\boldsymbol{x}'_3 - \theta^k(\boldsymbol{x}^1 - \boldsymbol{x}^*))\| \\ &\leq \|\boldsymbol{D}^{-1}(\Delta\boldsymbol{x}'_1 + \Delta\boldsymbol{x}'_3)\| + \|\boldsymbol{D}\Delta\boldsymbol{z}'_2\| \\ &\leq \theta^k\|\boldsymbol{D}^{-1}(\boldsymbol{x}^1 - \boldsymbol{x}^*)\| + \theta^k\|\boldsymbol{D}(\boldsymbol{z}^1 - \boldsymbol{z}^*)\| + \|(\boldsymbol{X}^k\boldsymbol{Z}^k)^{-1/2}(\boldsymbol{X}^k\boldsymbol{z}^k - \mu\boldsymbol{e})\| \end{aligned}$$

and similarly

$$\|\boldsymbol{D}\Delta\boldsymbol{z}\| \leq \theta^k\|\boldsymbol{D}^{-1}(\boldsymbol{x}^1 - \boldsymbol{x}^*)\| + \theta^k\|\boldsymbol{D}(\boldsymbol{z}^1 - \boldsymbol{z}^*)\| + \|(\boldsymbol{X}^k\boldsymbol{Z}^k)^{-1/2}(\boldsymbol{X}^k\boldsymbol{z}^k - \mu\boldsymbol{e})\|.$$

Thus we can prove the lemma following the same arguments as before.

REFERENCES

[1] J. A. KALISKI AND Y. YE, *An extension of the potential reduction algorithm for solving the linear complentary problem with priority goals*, Linear Algebra Appl., 193 (1993), pp. 35–50.

[2] N. K. KARMARKAR, *A new polynomial-time algorithm for linear programming*, Combinatorica, 4 (1984), pp. 373–395.

[3] M. KOJIMA, N. MEGIDDO, AND S. MIZUNO, *A primal-dual infeasible-interior-point algorithm for linear programming*, Math. Programming, 61 (1993), pp. 263–280.

[4] 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.

[5] ———, *A polynomial-time algorithm for a class of linear complementarity problems*, Math. Programming, 44 (1989), pp. 1–26.

[6] ———, *An $O(\sqrt{n}L)$ iteration potential reduction algorithm for linear complementarity problems*, Math. Programming, 50 (1991), pp. 331–342.

[7] M. KOJIMA, T. NOMA, AND A. YOSHISE, *Global convergence in infeasible-interior-point algorithms*, Math. Programming, 65 (1994), pp. 43–72.

[8] I. J. LUSTIG, *Feasibility issues in a primal-dual interior point method for linear programming*, Math. Programming, 49 (1990/91), pp. 145–162.

[9] 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.

[10] R. E. MARSTEN, R. SUBRAMANIAM, M. J. SALTZMAN, I. J. LUSTIG, AND D. F. SHANNO, *Interior point methods for linear programming: Just call Newton, Lagrange, and Fiacco and McCormick!*, Interfaces, 20 (1990), pp. 105–116.

[11] 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; Also, Proc. 6th Mathematical Programming Symposium of Japan, Nagoya, Japan, pp. 1–35, 1986.

[12] S. MIZUNO, *Polynomiality of the Kojima–Megiddo–Mizuno infeasible interior point algorithm for linear programming*, Tech. Report 1006, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, May 1992.

[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] 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. D. C. MONTEIRO AND I. ADLER, *Interior path following primal-dual algorithms: Part* I: *Linear programming*, Math. Programming, 44 (1989), pp. 27–41.

[16] ———, *Interior path following primal-dual algorithms: Part* II: *Convex quadratic programming*, Math. Programming, 44 (1989), pp. 43–66.

[17] F. A. POTRA, *An infeasible interior-point predictor-corrector algorithm for linear programming*, Reports on Computational Mathematics 26, Dept. of Mathematics, The University of Iowa, Iowa City, June 1992.

[18] K. TANABE, *Centered Newton method for mathematical programming*, in System Modelling and Optimization: Proc. 13th IFIP Conference, Tokyo, Japan, Aug./Sept. 1987, M. Iri and K. Yajima, eds., Vol. 113, Lecture Notes in Control and Information Sciences, Springer-Verlag, Berlin, West-Germany, 1988, pp. 197–206.

[19] ———, *Centered Newton method for linear programming: Interior and "exterior" point method*, in New Methods for Linear Programming 3, K. Tone, ed., The Institute of Statistical Mathematics, Tokyo, Japan, 1990, pp. 98–100. (In Japanese.)

[20] Y. YE, *An $O(n^3L)$ potential reduction algorithm for linear programming*, Math. Programming, 50 (1991), pp. 239–258.

[21] 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.

[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. OPTIMIZATION
Vol. 5, No. 1, pp. 68–99, February 1995

004

A FAST HEURISTIC METHOD FOR POLYNOMIAL MOMENT PROBLEMS WITH BOLTZMANN–SHANNON ENTROPY*

J. M. BORWEIN† AND W. Z. HUANG‡

Abstract. The authors consider the best entropy estimate to a nonnegative density $\bar{x}$ on $\mathbb{R}^m$, given some of its algebraic or trigonometric moments. Using the special structure of this kind of problem, a useful linear relationship among the moments is derived. A simple algorithm then provides a fairly good estimate of $\bar{x}$ by just solving a couple of linear systems. Numerical computations make the algorithm seem reasonable although the theoretical convergence is still an open problem. Some notes about the error bounds are given at the end of the paper.

Key words. convex programming, constrained optimization, moment problems, entropy, heuristic algorithms

AMS subject classifications. 49M39, 65K05, 90C30

1. Introduction.

The problem we discuss is the following Boltzmann–Shannon entropy moment problem:

$$(P_n)\qquad \begin{aligned} \inf\quad & \textstyle\int_T [x(t)\log(x(t)) - x(t)]dt \\ \text{s.t.}\quad & \textstyle\int_T a_i(t)x(t)dt = b_i,\quad i \in I_n \\ & 0 \le x \in L_1(T, dt), \end{aligned}$$

where $T \subset \mathbb{R}^m$ is compact, and in this paper we assume that $T = [0,1]^m$ or $[-\pi, \pi]^m$, while dt is the Lebesgue measure on T, and $\{a_i, i \in I_n\}$ are (algebraic or trigonometric) polynomials of order at most n. The I_n's are finite index sets such that

$$I_n \subset I_{n+1}, \qquad n = 1, 2, \ldots.$$

Let $k(n)$ denote the number of the elements in the set I_n. The limit problem is

$$(P_\infty)\qquad \begin{aligned} \inf\quad & \textstyle\int_T [x(t)\log(x(t)) - x(t)]dt \\ \text{s.t.}\quad & \textstyle\int_T a_i(t)x(t)dt = b_i,\quad i \in \bigcup_{n=1}^{\infty} I_n \\ & 0 \le x \in L_1(T, dt). \end{aligned}$$

If $\{a_i, i \in \bigcup_{n=1}^{\infty} I_n\}$ is weak*- dense in $L_\infty(T, dt)$, then the problem (P_∞) has unique solution $\bar{x}$. The convergence of the solutions of (P_n) to the solution of (P_∞) has been dealt with in many papers [1], [3]. From duality considerations [2], under some constraint qualification (CQ) condition, the solution x_n of (P_n) can be expressed as

$$x_n(t) = \exp\left(\sum_{i \in I_n} \lambda_i a_i(t)\right), \tag{1}$$

* Received by the editors December 22, 1992; accepted for publication (in revised form) September 15, 1993.

† Department of Mathematics and Statistics, Simon Fraser University, Burnaby, British Columbia, V5A 1S6.

‡ Department of Mathematical Sciences, Lakehead University, Thunder bay, Ontario, Canada, P7B 5E1 (wzhuang@flash.lakeheadu.ca). The work of this author formed part of her doctoral dissertation at Dalhousie University.

where the $\{\lambda_i, i \in I_n\}$ can be determined by the nonlinear system

$$\int_T \exp\left(\sum_{i\in I_n} \lambda_i a_i(t)\right) a_k(t)dt = b_k, \qquad k \in I_n.$$

This system can be solved by various iterative nonlinear or optimization techniques [4]–[6]. In this note, however, we consider heuristic methods for avoiding such iterative techniques.

Indeed, the special structure of the Kuhn–Tucker conditions (1) suggests that it might be possible to obtain the multipliers, λ_i, directly without recourse to solving what is in fact quite a costly convex program. While we cannot fully realize this ambition, we make the observation that in the next three sections if the underlying signal were of form (1) exactly, we could then obtain the multipliers from a knowledge of sufficiently many moments (twice as many in one dimension, four times as many in two dimensions, etc.). This involves solving a very simple linear system of equations. Heuristically, we argue that every positive smooth function is close to the exponential of a polynomial and so the solution to our linear system should provide a good estimator to the actual maximum entropy solution.

As we see in §5, our heuristic estimate performs very well in the sense that (i) it usually provides a very respectable reconstruction of the underlying signal, and (ii) it is much quicker to compute since it has removed all nonlinearities and is performed only once. Our numerical experiments show that it is often more than 10–50 times faster.

In this paper we see precisely that when the $\{a_i(t)\}$ are algebraic or trigonometric polynomials and T is a real interval or a cube in $\mathbb{R}^n$, we can obtain such a heuristic estimate. In §2 we determine the estimate given algebraic moments on [0,1]. In §3 we extend our analysis to algebraic moments in several dimensions. Section 4 provides the corresponding trigonometric estimates. In §5 we produce substantial numerical support for our heuristic. Finally, in §6 we make some modest attempts at an error analysis of the one-dimensional case.

2. Algebraic polynomial case on [0,1]. To explain the simple idea of our algorithms, we first consider $T = [0, 1]$, $a_i(t) = t^i$, and $I_n = \{0, 1, \ldots, n\}$. As we have observed, the optimal solution of (P_n) is "usually" of the form (1). Suppose that the underlying density $\bar{x}$ is exactly of the form

$$\bar{x}(t) = \exp\left(\sum_{i=0}^{n} \lambda_i a_i(t)\right) \tag{2}$$

for some n and we need to find out the arguments λ_i, $i = 0, 1, \ldots$. If we know $2n + 1$ moments given by

$$b_k = \int_0^1 \exp\left(\sum_{i=0}^{n} \lambda_i a_i(t)\right) t^k dt, \qquad k = 0, 1, \ldots, 2n,$$

integrating by parts, we have for $k = 0, 1, \ldots,$

$$b_k \triangleq \int_0^1 \exp\left(\sum_{i=0}^{n} \lambda_i a_i(t)\right) t^k dt$$

$$= \frac{1}{k+1} \exp\left(\sum_{i=0}^{n} \lambda_i a_i(t)\right) t^{k+1}|_0^1$$
$$-\frac{1}{k+1} \int_0^1 t^{k+1} \exp\left(\sum_{i=0}^{n} \lambda_i a_i(t)\right) \sum_{i=1}^{n} \lambda_i i t^{i-1} dt$$
$$= \frac{1}{k+1} \exp\left(\sum_{i=0}^{n} \lambda_i\right) - \frac{1}{k+1} \sum_{i=1}^{n} \lambda_i i b_{k+i},$$

or

$$(k+1)b_k = \exp\left(\sum_{i=0}^{n} \lambda_i\right) - \sum_{i=1}^{n} i\lambda_i b_{k+i}. \tag{3}$$

Thus $\lambda_0, \lambda_1, \ldots, \lambda_n$ in (2) can be obtained by solving the linear system

$$b = Br, \tag{4}$$

where

$$b = \begin{bmatrix} b_0 \\ 2b_1 \\ \vdots \\ (n+1)b_n \end{bmatrix}, \quad B = \begin{bmatrix} 1 & b_1 & b_2 & \cdots & b_n \\ 1 & b_2 & b_3 & \cdots & b_{n+1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & b_{n+1} & b_{n+2} & \cdots & b_{2n} \end{bmatrix}, \quad r = \begin{bmatrix} r_0 \\ r_1 \\ \vdots \\ r_n \end{bmatrix},$$

and

$$\begin{aligned} r_0 &= \exp\left(\textstyle\sum_{i=0}^{n} \lambda_i\right), \\ r_k &= -k\lambda_k, \qquad k = 1, 2, \ldots, n. \end{aligned} \tag{5}$$

It is not difficult to show that under a mild condition, which is implied by CQ, the linear system (4) is solvable.

LEMMA 1. *If there exists a nonzero density $\hat{x}$ on $[0,1]$, such that $b_0, b_1, \ldots, b_{2n}$ are given by*

$$b_k = \int_0^1 \hat{x}(t) t^k dt, \qquad k = 0, 1, \ldots, 2n,$$

then B is nonsingular.

Proof. Since

$$|B| = \begin{vmatrix} 1 & b_1 & b_2 & \cdots & b_n \\ 1 & b_2 & b_3 & \cdots & b_{n+1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & b_{n+1} & b_{n+2} & \cdots & b_{2n} \end{vmatrix}$$
$$= (-1)^n \begin{vmatrix} b_1 - b_2 & b_2 - b_3 & \cdots & b_n - b_{n+1} \\ b_2 - b_3 & b_3 - b_4 & \cdots & b_{n+1} - b_{n+2} \\ \vdots & \vdots & \cdots & \vdots \\ b_n - b_{n+1} & b_{n+1} - b_{n+2} & \cdots & b_{2n-1} - b_{2n} \end{vmatrix} \triangleq (-1)^n |D|.$$

For any $v = (v_1, v_2, \ldots, v_n)^T \in \mathbb{R}^n$, $v \neq 0$, consider

$$\begin{aligned} v^T D v &= \sum_{j=1}^{n}\sum_{i=1}^{n} v_i v_j (b_{i+j-1} - b_{i+j}) \\ &= \sum_{j=1}^{n}\sum_{i=1}^{n} v_i v_j \int_0^1 \hat{x}(t)(t^{i+j-1} - t^{i+j})dt \\ &= \int_0^1 \hat{x}(t) \left(\sum_{j=1}^{n}\sum_{i=1}^{n} v_i v_j t^{i+j-2} \right) t(1-t)dt \\ &= \int_0^1 \hat{x}(t) \left(\sum_{i=1}^{n} v_i t^{i-1} \right)^2 t(1-t)dt \\ &> 0. \end{aligned}$$

So D is positive definite, $|D| \neq 0$, and so $|B|$ is nonzero. □

From the lemma we see that if $\{b_k\}$ are consistent then there is a unique solution for the linear system (4). And thus the parameters $\lambda_0, \lambda_1, \ldots, \lambda_n$ can be obtained from (5).

For a density $\bar{x}$ other than of the form in (2), we may use this simple method to get a heuristic estimate of $\bar{x}$. We can see that in (5), r_0 is required to be positive, which may not be true all the time. But from the first moment

$$\begin{aligned} b_0 &= \int_0^1 \exp\left(\sum_{i=0}^{n} \lambda_i t^i \right) dt \\ &= e^{\lambda_0} \int_0^1 \exp\left(\sum_{i=1}^{n} \lambda_i t^i \right) dt, \end{aligned}$$

we can still "determine" r_0 when $\lambda_1, \lambda_2, \ldots, \lambda_n$ are known.

ALGORITHM 1. Let $2n+1$ moments $b_0, b_1, \ldots, b_{2n}$ be given.

Step 1. Construct:

$$B_n = \begin{bmatrix} 1 & b_1 & b_2 & \cdots & b_n \\ 1 & b_2 & b_3 & \cdots & b_{n+1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & b_{n+1} & b_{n+2} & \cdots & b_{2n} \end{bmatrix}, \qquad b^n = \begin{bmatrix} b_0 \\ 2b_1 \\ \vdots \\ (n+1)b_n \end{bmatrix}.$$

Step 2. Compute $r^n \in \mathbb{R}^{n+1}$ which solves the linear system

$$B_n r^n = b^n.$$

Step 3. Compute $\lambda^n \in \mathbb{R}^{n+1}$ as follows:

$$\begin{aligned} \lambda_k^n &= -\frac{r_k^n}{k}, \qquad k = 1, 2, \ldots, n, \\ \lambda_0^n &= \log\left(\frac{b_0}{\int_0^1 \exp(\sum_{i=1}^{n} \lambda_i^n t^i)dt} \right). \end{aligned}$$

Step 4. Construct

$$x_n(t) = \exp\left(\sum_{i=0}^{n} \lambda_i^n t^i\right).$$

The following fact is now obvious.

THEOREM 1. *If the prior density $\bar{x}$ is of the form* (2) *for some $\lambda \in \mathbb{R}^n$, and the first $2n+1$ moments are given, then the estimate density constructed by Algorithm* 1 *is exactly $\bar{x}$ itself.*

3. Algorithm generalized to $[0,1]^m$. We first consider $T = [0,1]^2$. For $n = (n_1, n_2) \in \mathbf{Z}_+^2$, let $\{a_i,\ i \in I_n\}$ be algebraic polynomials of degree at most n_1 in t_1 and n_2 in t_2, of the form

$$t_1^i t_2^j, \quad i = 0, 1, \ldots, n_1, \quad j = 0, 1, \ldots, n_2.$$

If we assume that

$$\bar{x}(t_1, t_2) = \exp\left(\sum_{i=0}^{n_1}\sum_{j=0}^{n_2} \lambda_{i,j} t_1^i t_2^j\right), \tag{6}$$

and the moments are given by

$$b_{l_1,l_2} = \int_0^1 \int_0^1 \bar{x}(t_1, t_2) t_1^{l_1} t_2^{l_2} dt_2 dt_1 \tag{7}$$

for $l_1 = 0, 1, \ldots, 2n_1$, $l_2 = 0, 1, \ldots, 2n_2$, then the formula analogue to (3) is

$$\begin{aligned} b_{l_1,l_2} = {} & \frac{1}{l_1+1} \int_0^1 \exp\left(\sum_{i=0}^{n_1}\sum_{j=0}^{n_2} \lambda_{i,j} t_2^j\right) t_2^{l_2} dt_2 \\ & - \frac{1}{l_1+1} \sum_{i=1}^{n_1}\sum_{j=0}^{n_2} \lambda_{i,j} i b_{l_1+i,l_2+j}, \end{aligned} \tag{8}$$

or

$$\begin{aligned} (l_1+1) b_{l_1,l_2} = {} & \int_0^1 \exp\left(\sum_{i=0}^{n_1}\sum_{j=0}^{n_2} \lambda_{i,j} t_2^j\right) t_2^{l_2} dt_2 \\ & - \sum_{i=1}^{n_1}\sum_{j=0}^{n_2} i \lambda_{i,j} b_{l_1+i,l_2+j}. \end{aligned}$$

Now let

$$r_{0,l_2} = \int_0^1 \exp\left(\sum_{i=0}^{n_1}\sum_{i=0}^{n_2} \lambda_{i,j} t_2^j\right) t_2^{l_2} dt_2, \qquad l_2 = 0, 1, \ldots, n_2$$

and

$$r_{l_1,l_2} = -l_1 \lambda_{l_1,l_2}, \quad l_1 = 1, 2, \ldots, n_1, \quad l_2 = 0, 1, \ldots, n_2.$$

Then we can solve a linear system and obtain $\lambda_{i,j}$, $i \neq 0$. Switching the order of t_1 and t_2, and integrating by parts in (8), we have a linear system that can be used to find $\lambda_{i,j}$, $j \neq 0$. Finally, from the first moment $b_{0,0}$, we can determine $\lambda_{0,0}$.

ALGORITHM 2. Let $b_{i,j}$, $i = 0, 1, \ldots, 2n_1$, $j = 0, 1, \ldots, 2n_2$ be given moments in (7).

Step 1. Construct

$$d_k = \begin{bmatrix} b_{k,0} \\ b_{k,1} \\ \vdots \\ b_{k,n_2} \end{bmatrix}, \quad u_k = \begin{bmatrix} r_{k,0} \\ r_{k,1} \\ \vdots \\ r_{k,n_2} \end{bmatrix}, \quad k = 0, 1, \ldots, n_1,$$

$$D_k = \begin{bmatrix} b_{k,0} & b_{k,1} & \cdots & b_{k,n_2} \\ b_{k,1} & b_{k,2} & \cdots & b_{k,n_2+1} \\ \vdots & \vdots & \cdots & \vdots \\ b_{k,n_2} & b_{k,n_2+1} & \cdots & b_{k,2n_2} \end{bmatrix}, \qquad k = 1, 2, \ldots, 2n_1.$$

Step 2. Solve the linear system

$$d = Du,$$

where

$$d = \begin{bmatrix} d_0 \\ 2d_1 \\ \vdots \\ (n_1+1)d_{n_1} \end{bmatrix}, \qquad u = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{n_1} \end{bmatrix},$$

$$D = \begin{bmatrix} I & D_1 & D_2 & \cdots & D_{n_1} \\ I & D_2 & D_3 & \cdots & D_{n_1+1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ I & D_{n_1+1} & D_{n_1+2} & \cdots & D_{2n_1} \end{bmatrix}.$$

Step 3. Compute

$$\lambda_{i,j} = -\frac{1}{l_1} r_{l_1,l_2}, \quad l_1 = 1, 2, \ldots, n_1, \quad l_2 = 0, 1, \ldots, n_2.$$

Step 4. Compute

$$b'_{l_2} = (l_2+1)b_{0,l_2} + \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} j\lambda_{i,j} b_{i,l_2+j}, \qquad l_2 = 0, 1, \cdots, n_2$$

and solve the linear system

$$b' = B'u',$$

where

$$b' = \begin{bmatrix} b'_0 \\ b'_1 \\ \vdots \\ b_{n_2} \end{bmatrix}, \qquad u' = \begin{bmatrix} r'_0 \\ r'_2 \\ \vdots \\ r'_{n_2} \end{bmatrix},$$

and

$$B' = \begin{bmatrix} 1 & b_{0,1} & b_{0,2} & \cdots & b_{0,n_2} \\ 1 & b_{0,2} & b_{0,3} & \cdots & b_{0,n_2+1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & b_{0,n_2+1} & b_{0,n_2+2} & \cdots & b_{0,2n_2} \end{bmatrix}.$$

Step 5. Compute

$$\lambda_{0,j} = -\frac{1}{j} r'_j, \qquad j = 1, 2, \ldots, n_2.$$

Step 6. Finally we have

$$\lambda_{0,0} = \log \left[b_{0,0} \left(\int_0^1 \int_0^1 \exp \left(\sum_{(i,j)\neq(0,0)} \lambda_{i,j} t_1^i t_2^j \right) dt_2 dt_1 \right)^{-1} \right]$$

and

$$x_n(t_1, t_2) = \exp \left(\sum_{i=0}^{n_1} \sum_{j=0}^{n_2} \lambda_{i,j} t_1^i t_2^j \right)$$

is the estimate density.

Anologuously to Theorem 1, we have the following theorem.

THEOREM 2. *If the prior density $\bar{x}$ is of the form* (6) *for some $n_1, n_2 \in \mathbf{Z}_+$, and we know the first $(2n_1+1)(2n_2+1)$ moments given by* (7), *then the estimate density x_n constructed by Algorithm* 2 *is exactly $\bar{x}$ itself.*

Now we generalize the algorithm to $[0,1]^m$. Let $T = [0,1]^m$,

$$\begin{aligned} n &\triangleq (n_1, n_2, \ldots, n_m)^T \in \mathbf{Z}_+^m, \\ I_n &\triangleq \{(i_1, i_2, \ldots, i_m)^T \in \mathbf{Z}_+^m \mid i_j = 0, 1, \ldots, n_j,\ j = 1, 2, \ldots, m\} \\ &\equiv \{(i \in \mathbf{Z}_+^m \mid 1 \leq i \leq n\}, \end{aligned}$$

$\{a_i, i \in I_n\}$ be algebraic polynomials of the form

$$t_1^{i_1} t_2^{i_2} \cdots t_m^{i_m}, \quad i_j = 0, 1, \ldots, n_j, \quad j = 1, 2, \ldots, m.$$

Then we have

$$k(n) = \prod_{j=1}^{m} (n_j + 1).$$

Assume $\bar{x}$ is of the form

$$\bar{x}(t_1, t_2, \ldots, t_m) = \exp \left(\sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_2} \cdots \sum_{i_m=0}^{n_m} \lambda_{i_1, i_2, \ldots, i_m} t_1^{i_1} t_2^{i_2} \cdots t_m^{i_m} \right)$$

or

$$\bar{x}(t) = \exp \left(\sum_{i \in I_n} \lambda_i t^i \right).$$

Here we denote

$$t \triangleq (t_1, t_2, \ldots, t_m)^T \in \mathbb{R}^m,$$
$$i \triangleq (i_1, i_2, \ldots, i_m)^T \in \mathbf{Z}_+^m,$$
$$t^i \triangleq t_1^{i_1} t_2^{i_2} \cdots t_m^{i_m},$$

and

$$\lambda_i \triangleq \lambda_{i_1, i_2, \ldots, i_m}, \quad i \in I_n,$$

hence

$$\lambda \in \mathbb{R}^{k(n)}.$$

Then for $l_i = 0, 1, \ldots, n_i$, $i = 1, 2, \ldots, m$, the moments are given by

$$b_{l_1, l_2, \ldots, l_m} = \int_0^1 \int_0^1 \cdots \int_0^1 \bar{x}(t_1, t_2, \ldots, t_m) t_1^{l_1} t_2^{l_2} \cdots t_m^{l_m} dt_1 dt_2 \cdots dt_m$$

or

$$b_l = \int_{[0,1]^m} \bar{x}(t) t^l dt \qquad l \in I_n,$$

where

$$l = (l_1, l_2, \ldots, l_m) \in \mathbf{Z}_+^m,$$
$$dt = dt_1 dt_2 \cdots dt_m.$$

Note that for each $j = 1, 2, \ldots, m$, integrating by parts, we have

$$b_l = \frac{1}{l_j + 1} \int_{[0,1]^{m-1}} \exp\left(\sum_{i \in I_n(j)} \lambda_i t(j)^{i(j)} \right) t(j)^{l(j)} dt(j)$$
$$- \frac{1}{l_j + 1} \sum_{i \in I_n \setminus \{i_j = 0\}} i_j \lambda_i b_{i+l} \quad \forall l \in I_n,$$

where

$$I_n(j) \triangleq \{i \in I_n \mid i_j = 0\},$$
$$t(j) \triangleq (t_1, \ldots, t_{j-1}, t_{j+1}, \ldots, t_m)^T \in \mathbb{R}^{m-1},$$
$$i(j) \triangleq (i_1, \ldots, i_{j-1}, i_{j+1}, \ldots, i_m)^T \in \mathbf{Z}_+^{m-1},$$
$$dt(j) \triangleq dt_1 \cdots dt_{j-1} dt_{j+1} \cdots dt_m.$$

The algorithm can be stated as follows.

ALGORITHM 3. Let $b_i, i \in I_{2n}$ be $\prod_{k=1}^m (2n_k + 1)$ the moments given in (7).
Step 1. Construct linear system with $\prod_{k=1}^m (n_k + 1)$ unknowns:

$$d_{l_1, l_2, \ldots, l_m} = r_{0, l_2, \ldots, l_m} + \sum_{i_1=1}^{n_1} \sum_{i_2=0}^{n_2} \cdots \sum_{i_m=0}^{n_m} r_{i_1, \ldots, i_m} b_{i_1 + l_1, \ldots, i_m + l_m},$$

where

$$d_{l_1,l_2,\ldots,l_m} = (l_1+1)b_{l_1,l_2,\ldots,l_m},$$

and solve it. Let

$$\lambda_{i_1,\ldots,i_m} = -\frac{r_{i_1,\ldots,i_m}}{i_1},$$

for

$$i_1 = 1,\ldots,n_1, \quad i_2 = 0,\ldots,n_2,\ldots, \quad i_m = 0,\ldots,n_m.$$

We have that $j = 1$.

Step 2. If $j \le m-1$, construct the linear system with $\prod_{k=j+1}^{m}(n_k+1)$ unknowns:

$$d_{l_{j+1},\ldots,l_m} = r_{0,l_{j+2},\ldots,l_m} + \sum_{i_{j+1}=1}^{n_{j+1}} \sum_{i_{j+2}=0}^{n_{j+2}} \cdots \sum_{i_m=0}^{n_m} r_{i_{j+1},\ldots,i_m} b_{0,\ldots,0,i_{j+1}+l_{j+1},\ldots,i_m+l_m},$$

for

$$l_{j+1} = 1,\ldots,n_{j+1}, \quad l_{j+2} = 0,\ldots,n_{j+2},\ldots, \quad l_m = 0,\ldots,n_m,$$

where

$$d_{l_{j+1},\ldots,l_m} = (l_{j+1}+1)b_{0,\ldots,0,l_{j+1},\ldots,l_m} + \sum_{i_1+\cdots+i_j>0} i_{j+1}\lambda_{i_1,\ldots,i_m} b_{i_1,\ldots,i_j,i_{j+1}+l_{j+1},\ldots,i_m+l_m}.$$

Solve it and let

$$\lambda_{0,\ldots,0,l_{j+1},\ldots,l_m} = -\frac{r_{l_{j+1},\ldots,l_m}}{l_{j+1}},$$

for

$$l_{j+1} = 1,\ldots,n_{j+1},\ l_{j+2} = 0,\ldots,n_{j+2},\ldots,l_m = 0,\ldots,n_m.$$

We have that $j = j+1$. Repeat Step 2.

Step 3. When $j = m$, compute:

$$\lambda_{0,\ldots,0} = \log\left[b_{0,\ldots,0}\left(\int_0^1\int_0^1\cdots\int_0^1 \exp\left(\sum_{i_1+i_2+\cdots+i_m>0} \lambda_{i_1,\ldots,i_m} t_1^{i_1}\cdots t_m^{i_m}\,dt_1\cdots dt_m\right)\right)^{-1}\right].$$

Then the estimate density is

$$x_n(t) = \exp\left(\sum_{i\in I_n} \lambda_i t^i\right).$$

4. Trigonometric polynomial cases. We first consider the trigonometrical case on the interval $[-\pi, \pi]$. Let $I_n = \{-n, \ldots, 0, \ldots, n\}$, $a_k(t) = e^{\imath kt}$, $k \in I_n$, where $\imath = \sqrt{-1}$. Then the problem becomes

$$(P_n) \qquad \begin{aligned} \min \quad & \int_{-\pi}^{\pi} [x(t)\log(x(t)) - x(t)]dt, \\ \text{s.t.} \quad & \int_{-\pi}^{\pi} x(t)e^{\imath kt}dt = b_k, \qquad k = -n, \ldots, 0, \ldots, n, \\ & 0 \le x(t) \in L_1[-\pi, \pi]. \end{aligned}$$

We consider only the case where $x(t)$ is real. In this case we have

$$b_{-k} = \bar{b}_k \quad \forall k.$$

Moreover, we may assume that $\bar{x}(t)$ is of the form

$$\bar{x}(t) = \exp\left(\sum_{k=-n}^{n} \lambda_k e^{\imath kt}\right),$$

and that we have

$$\lambda_{-k} = -\bar{\lambda}_k \quad \forall k,$$

since $\bar{x}$ is assumed to be real.

As to the integration property, for $k \neq 0$ we have

$$\begin{aligned} b_k &= \int_{-\pi}^{\pi} \exp\left(\sum_{l=-n}^{n} \lambda_l e^{\imath lt}\right) e^{\imath kt}dt \\ &= \frac{1}{\imath k} \exp\left(\sum_{l=-n}^{n} \lambda_l e^{\imath lt}\right) e^{\imath kt}|_{-\pi}^{\pi} \\ &\quad -\frac{1}{\imath k} \int_{-\pi}^{\pi} e^{\imath kt} \exp\left(\sum_{l=-n}^{n} \lambda_l e^{\imath lt}\right) \sum_{l=-n}^{n} \lambda_l \imath l e^{\imath lt}dt \\ &= -\frac{1}{k} \sum_{l=-n}^{n} \lambda_l l b_{k+l}. \end{aligned}$$

Using the property that $\lambda_{-k} = -\bar{\lambda}_k$, we have the linear system

$$b = C\bar{r} + Br,$$

where

$$b = \begin{bmatrix} -b_1 \\ -2b_2 \\ \vdots \\ -nb_n \end{bmatrix}, \qquad r = \begin{bmatrix} \lambda_1 \\ 2\lambda_2 \\ \vdots \\ n\lambda_n \end{bmatrix},$$

$$C = \begin{bmatrix} 0 & \bar{b}_1 & \cdots & \bar{b}_{n-1} \\ b_1 & 0 & \cdots & \bar{b}_{n-2} \\ \vdots & \vdots & \cdots & \vdots \\ b_{n-1} & b_{n-2} & \cdots & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} b_2 & b_3 & \cdots & b_{n+1} \\ b_3 & b_4 & \cdots & b_{n+2} \\ \vdots & \vdots & \cdots & \vdots \\ b_{n+1} & b_{n+2} & \cdots & b_{2n} \end{bmatrix}.$$

Solving this system, we can determine all λ_k, $k \neq 0$. Finally λ_0 can be obtained from

$$a_0 = \int_{-\pi}^{\pi} \exp\left(\sum_{k=-n}^{n} \lambda_k e^{\imath kt}\right) dt$$
$$= e^{\lambda_0} \int_{-\pi}^{\pi} \exp\left(\sum_{k\neq 0} \lambda_k e^{\imath kt}\right) dt.$$

We can also express everything above in real form. Let

$$\bar{x}(t) = \exp\left(\lambda_0 + \sum_{k=1}^{n} (\lambda_k \cos kt + \mu_k \sin kt)\right),$$

and the moments

$$a_0 = \int_{-\pi}^{\pi} \bar{x}(t)dt, \tag{9}$$

$$a_k = \int_{-\pi}^{\pi} \bar{x}(t) \cos ktdt, \quad k = 1, 2, \ldots, n, \tag{10}$$

$$b_k = \int_{-\pi}^{\pi} \bar{x}(t) \sin ktdt, \quad k = 1, 2, \ldots, n. \tag{11}$$

Then for $l = 1, 2, \ldots, n$, using trigonometric angle formula, we have

$$a_l = \int_{-\pi}^{\pi} \exp\left(\lambda_0 + \sum_{1}^{n} (\lambda_k \cos kt + \mu_k \sin kt)\right) \cos ltdt$$
$$= \frac{1}{2l} \sum_{k=1}^{n} k[\lambda_k(a_{l-k} - a_{l+k}) - \mu_k(b_{l-k} + b_{l+k})],$$

and

$$b_l = \frac{1}{2l} \sum_{k=1}^{n} k[-\lambda_k(b_{l+k} - b_{l-k}) + \mu_k(a_{l-k} + a_{l+k})].$$

Note that $\bar{x}$ is real, thus

$$a_{-k} = a_k \quad \forall k$$

and

$$b_{-k} = -b_k \quad \forall k,$$

The next algorithm then follows after some arithmetic calculation.

ALGORITHM 4. Let a_k, b_k, $k = 0, 1, \ldots, 2n$ be given moments in (9).
Step 1. Construct:

$$a = \begin{bmatrix} 2a_1 \\ 4a_2 \\ \vdots \\ 2na_n \end{bmatrix}, \qquad b = \begin{bmatrix} 2b_1 \\ 4b_2 \\ \vdots \\ 2nb_n \end{bmatrix},$$

$$A_1 = \begin{bmatrix} a_0 & a_1 & \cdots & a_{n-1} \\ a_1 & a_0 & \cdots & a_{n-2} \\ \vdots & \vdots & \cdots & \vdots \\ a_{n-1} & a_{n-2} & \cdots & a_0 \end{bmatrix}, \qquad A_2 = \begin{bmatrix} a_2 & a_3 & \cdots & a_{n+1} \\ a_3 & a_4 & \cdots & a_{n+2} \\ \vdots & \vdots & \cdots & \vdots \\ a_{n+1} & a_{n+2} & \cdots & a_{2n} \end{bmatrix},$$

$$B_1 = \begin{bmatrix} 0 & -b_1 & \cdots & -b_{n-1} \\ b_1 & 0 & \cdots & -b_{n-2} \\ \vdots & \vdots & \cdots & \vdots \\ b_{n-1} & b_{n-2} & \cdots & 0 \end{bmatrix}, \qquad B_2 = \begin{bmatrix} b_2 & b_3 & \cdots & b_{n+1} \\ b_3 & b_4 & \cdots & b_{n+2} \\ \vdots & \vdots & \cdots & \vdots \\ b_{n+1} & b_{n+2} & \cdots & b_{2n} \end{bmatrix}.$$

Solve the linear system

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} A_1 - A_2 & -B_1 - B_2 \\ B_1 - B_2 & A_1 + A_2 \end{bmatrix} \begin{bmatrix} r \\ s \end{bmatrix}. \tag{12}$$

Step 2. For $i = 1, 2, \ldots, n$, let

$$\lambda_i = \frac{r_i}{i}$$

and

$$\mu_i = \frac{s_i}{i}.$$

Step 3. Compute:

$$\lambda_0 = \log \left[\frac{a_0}{\int_{-\pi}^{\pi} \exp(\sum_{i=1}^{n} (\lambda_i \cos it + \mu_k \sin it)) dt} \right].$$

In a similar way, we can generalize this to m-dimensional space $\mathbb{R}^m$. Let $T = [-\pi, \pi]^m$, $\{a_i(t), i \in I_n\}$ be trigonometric polynomials of the form

$$e^{i(k_1 t_1 + \cdots + k_m t_m)}, \quad k_j = -n_j, \ldots, 0, \ldots, n_j, \quad j = 1, 2, \ldots, m.$$

Let

$$\begin{aligned} n &\triangleq (n_1, n_2, \ldots, n_m)^T \in \mathbf{Z}_+^m, \\ k &\triangleq (k_1, k_2, \ldots, k_m)^T \in \mathbf{Z}_+^m, \\ I_n &\triangleq \{k \in \mathbf{Z}_+^m \mid -n \le k \le n\}, \\ I_n(j) &\triangleq \{i \in I_n,\ i_j \neq 0\}, \end{aligned}$$

then

$$k(n) = \prod_{j=1}^{m} (2n_j + 1).$$

Assume

$$\bar{x}(t) = \exp\left(\sum_{k \in I_n} \lambda_k e^{ikt} \right),$$

where

$$t \triangleq (t_1, t_2, \ldots, t_m)^T \in I\!R^m,$$

$$\lambda \triangleq \{\lambda_k,\ k \in I_n\} \in \mathbf{C}^{k(n)},$$

and the moments are

$$b_l = \int_{[-\pi,\pi]^m} \bar{x}(t) e^{ilt} dt,\quad l \in I_n,$$

where

$$dt = dt_1 dt_2 \cdots dt_m.$$

By the integration procedure, we have

$$b_l = \frac{1}{l_j} \sum_{k \in I_n} k_j \lambda_k b_{l+k},\quad l \in I_n \setminus \{l_j = 0\},\quad j = 1, 2, \ldots, m.$$

Supposing that we know all the moments b_l, $l \in I_{2n}$, we can get all λ_k, $k \in I_n$, using the following algorithm.

ALGORITHM 5. Let b_l, $l \in I_{2n}$ be given moments.

Step 1. Solve the linear equations:

$$b_l = \frac{1}{l_1} \sum_{k \in I_n(1)} k_1 \lambda_k b_{l+k},\quad l \in I_n(1),$$
$$j = 1.$$

Step 2. If $j < m$, solve

$$l_{j+1} b_l - \sum_{i \in I_n(j+1) \setminus \{i_1 = \cdots = i_j = 0\}} i_{j+1} \lambda_i b_{i+l} = \sum_{i \in I_n(j+1) \cap \{i_1 = \cdots = i_j = 0\}} i_{j+1} \lambda_i b_{i+l}.$$

It holds that $j = j + 1$. Repeat Step 2.

Step 3. When $j = m$, compute

$$\lambda_0 = \log \left[\frac{a_0}{\int_{[-\pi,\pi]^m} \exp(\sum_{k \in I_n \setminus \{0\}} \lambda_k e^{ikt}) dt} \right].$$

5. Numerical results. We implemented our algorithms in Fortran 77 to solve one- and two-dimensional best-entropy moment problems with algebraic or trigonometric moment functions.

To make a comparison, we also implemented a classical Newton method combined with the Armijo step length search technique to solve the dual problem (see [1])

$$(D_n) \qquad \begin{array}{ll} \max & \Phi(\lambda) \triangleq \sum_{i \in I_n} \lambda_i b_i - \int_T \exp(\sum_{i \in I_n} \lambda_i a_i(t)) dt \\ \text{s.t.} & \lambda \in I\!R^{k(n)} \end{array}$$

using the same number of moments.

The following notations are helpful in reading the tables and figures below.

- $\bar{x}(t)$ (or $\bar{x}(t_1, t_2)$): the prior density function.
- sup-ERR: the supremum norm of $\bar{x} - x_n$, where x_n is the estimate density function constructed by the corresponding algorithm.
- L_1-ERR: the L_1-norm of $\bar{x} - x_n$.
- d-GAP: the duality gap defined by $V(P_n) - V(D_n)$.
- TIME: execution time (in seconds) used to compute the dual solution λ only.

Example 1 (see Table 1 and Figs. 1 and 2). We first consider a step function

$$\bar{x}(t) = 0.5\chi_{[0,0.5]} + 0.1$$

on interval $[0, 1]$, and use the first 25 algebraic moments to reconstruct $\bar{x}$. In Table 1, ALG1 means the Algorithm 1 given in §2, NEWTON(k) is Newton's method starting from the initial point

$$\lambda^0 = (\log b_0, 0, \ldots, 0) \in I\!R^{25}$$

and making k iterations. OPTIMAL means we use Newton's method to solve the problem (P_n) and iterate until the termination criteria are satisfied. In our case we use

$$\|\nabla\Phi(\lambda)\|_\infty < \varepsilon(= 0.0001),$$

and the optimal solution is then of the form

$$\sum_{i \in I_n} \lambda_i^n a_i(t),$$

where $\lambda^n \in I\!R^{k(n)}$ is obtained from OPTIMAL.

TABLE 1
Numerical results for Example 1.

	sup-ERR	L_1-ERR	d-GAP	TIME	
ALG 1	0.231054	0.040201	0.012864	0.0300	Fig.1
NEWTON(100)	0.819201	0.041938	0.014519	7.6685	
OPTIMAL	0.234438	0.024977	0.000081		Fig.2

Example 2 (see Table 2 and Figs. 3 and 4). We now consider the function

$$\bar{x}(t) = 0.3\chi_{[1,2]} + 0.6\chi_{[3,5]} + 0.1$$

on interval $[0, 2\pi]$, and use 25 trigonometric moments.

TABLE 2
Numerical results for Example 2.

	sup-ERR	L_1-ERR	d-GAP	TIME	
ALG 4	0.255256	0.437978	1.575890	0.0100	Fig.3
NEWTON(4)	0.249361	0.557484	0.555667	0.8498	
OPTIMAL	0.225884	0.212427	0.000035		Fig.4

For a smooth density, the results are much better as we would expect.

Example 3 (see Table 3 and Figs. 5 and 6). We fix a smooth density

$$\bar{x}(t) = t\sin^2(10t)$$

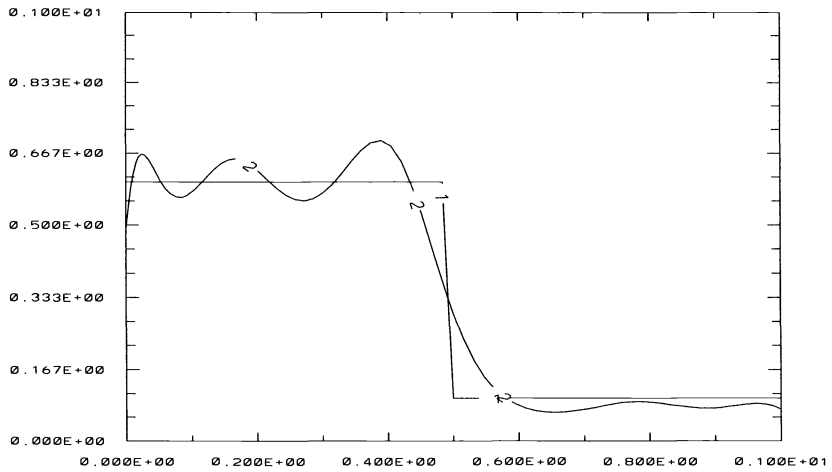


FIG. 1. *Prior and estimate density for Algorithm* 1.

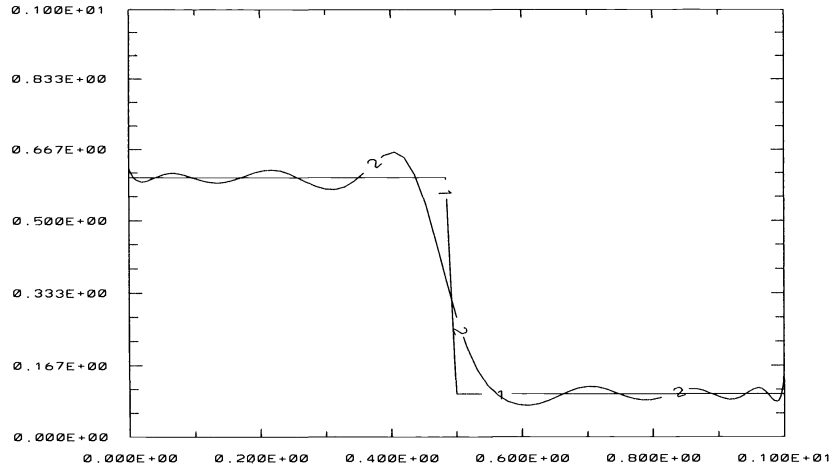


FIG. 2. *Prior and optimal solution to* (P_n).

on interval $[0, 1]$, and use 25 algebraic moments.

Note that the objective function we used here is the Boltzmann–Shannon entropy, it is neither the supremum norm nor the L_1-norm. We use these norms here just to compare the results and to measure the goodness of our reconstructions. Actually, it is the d-GAP that measures our success in getting our numerical estimates close to the optimal solution of (P_n).

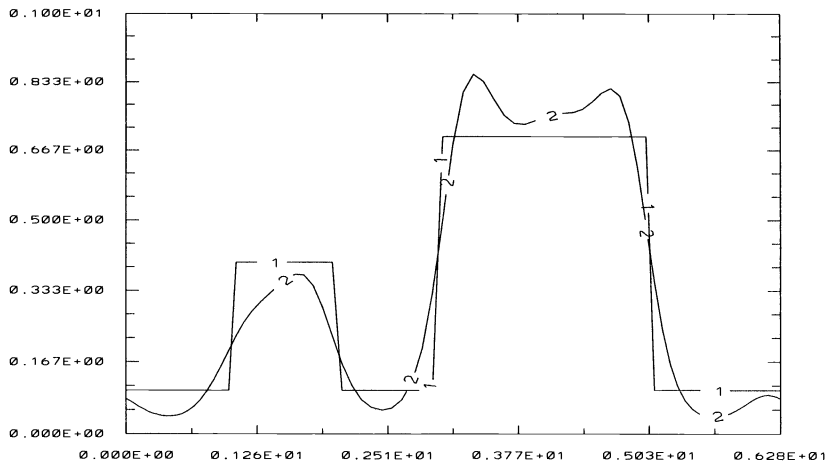


FIG. 3. *Prior and estimate density for Algorithm* 4.

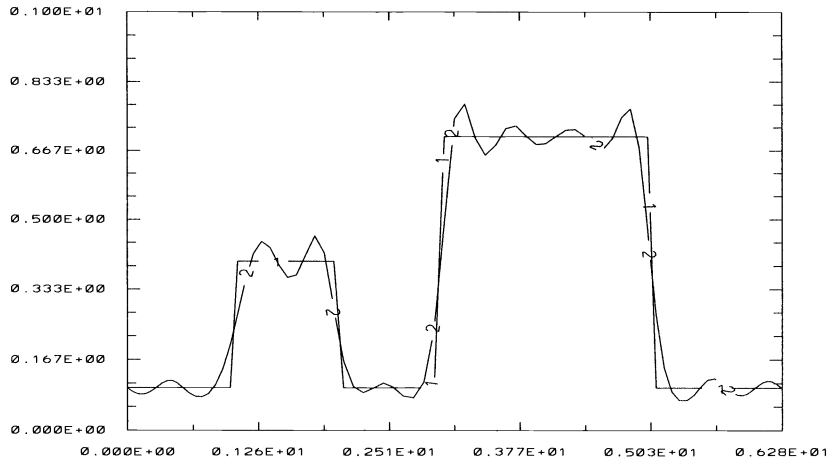


FIG. 4. *Prior and optimal solution to* (P_n).

Example 4 (see Table 4 and Figs. 7–13). We now deal with a two-dimensional function, first consider a smooth function,

$$\bar{x}(t_1, t_2) = t_1 t_2 (\sin(6t_1)\cos(8t_2))^2$$

on $[0, 1]^2$ (see Fig.7), and use 81 algebraic moments. To save time, we use the estimate density generated from our heuristic algorithm as the initial solution of the Newton method.

TABLE 3
Numerical results for Example 3.

	sup-ERR	L_1-ERR	d-GAP	TIME	
ALG 1.	0.121938	0.043898	0.008408	0.0400	Fig.5
NEWTON(15)	0.122698	0.036004	0.009664	1.7497	
OPTIMAL	0.114191	0.026806	0.000465		Fig.6

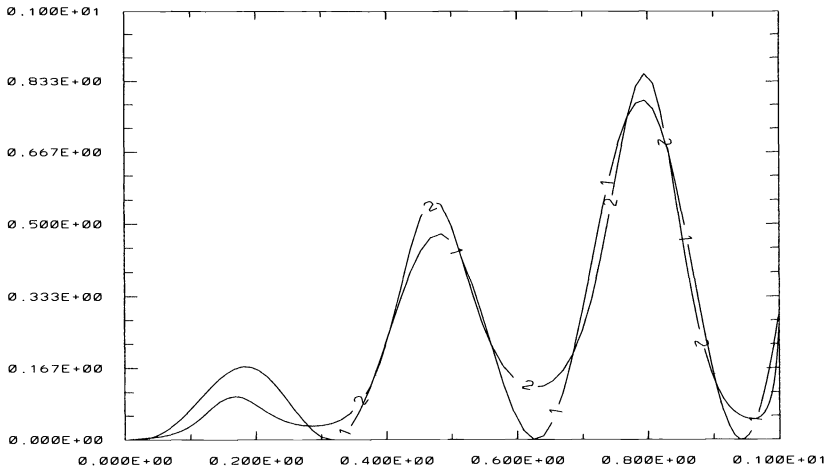


FIG. 5. *Prior and estimate density for Algorithm* 1.

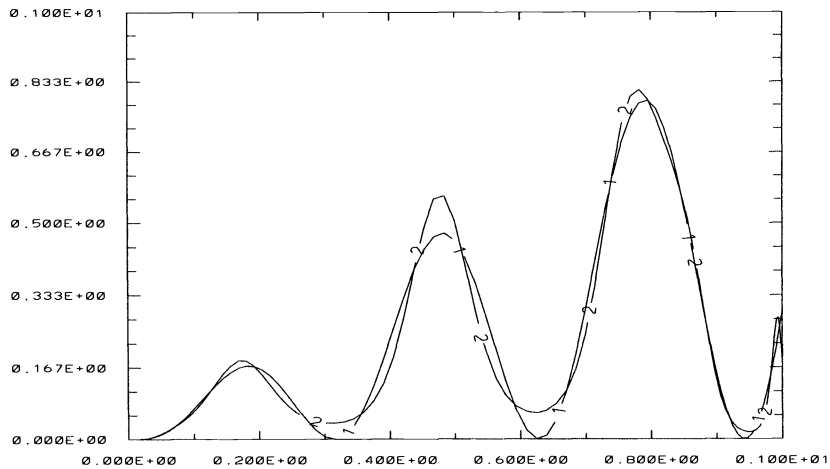


FIG. 6. *Prior and optimal solution to* (P_n).

TABLE 4
Numerical results for Example 4.

	sup-ERR	L_1-ERR	TIME	estimat.	sup-error
ALG 2.	0.24057	0.03226	0.21996	Fig.8	Fig.9
NEWTON(6)	1.91933	0.01440	52.6195	Fig.10	Fig.11
OPTIMAL	0.11596	0.01306		Fig.12	Fig.13

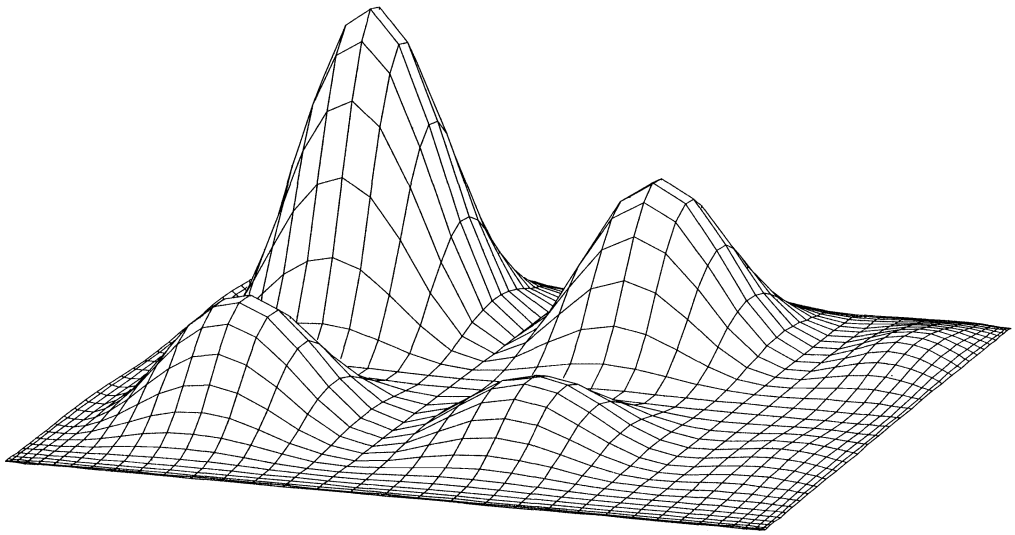


FIG. 7. *Prior function in Example* 4.

Example 5 (see Table 5 and Figs. 14–20). As the final example, we consider a step function on $[0,1]^2$ (see Fig. 14), given by

$$\bar{x}(t_1,t_2) = \begin{cases} 0.2, & 0 \le t_1 < 0.5,\ 0 \le t_2 < 0.5, \\ 0.4, & 0 \le t_1 < 0.5,\ 0.5 \le t_2 \le 1, \\ 0.6, & 0.5 \le t_1 \le 1,\ 0 \le t_2 < 0.5, \\ 0.8, & 0.5 \le t_1 \le 1,\ 0.5 \le t_2 \le 1 \end{cases}$$

and use 81 algebraic moments.

6. Notes about error analysis in $\mathbb{R}^1$. In this section, we give some error estimates in one-dimensional cases. We consider that $T = [0,1]$ or $[-\pi,\pi]$ and $\{a_i(t)\}$ are algebraic or trigonometric polynomials in only one variable. As we know, Algorithms 1–5 are exact when the underlying density $\bar{x}$ can be expressed as the exponential of a polynomial of $\{a_i,\ i \in I_n\}$. Now we assume that $\bar{x}$ is of almost this form; that is

$$\bar{x}(t) \cong \exp\left[\sum_{i\in I_n} \lambda_i a_i(t)\right]$$

in some sense, and we wish to determine arguments $\lambda_i,\ i \in I_n$.

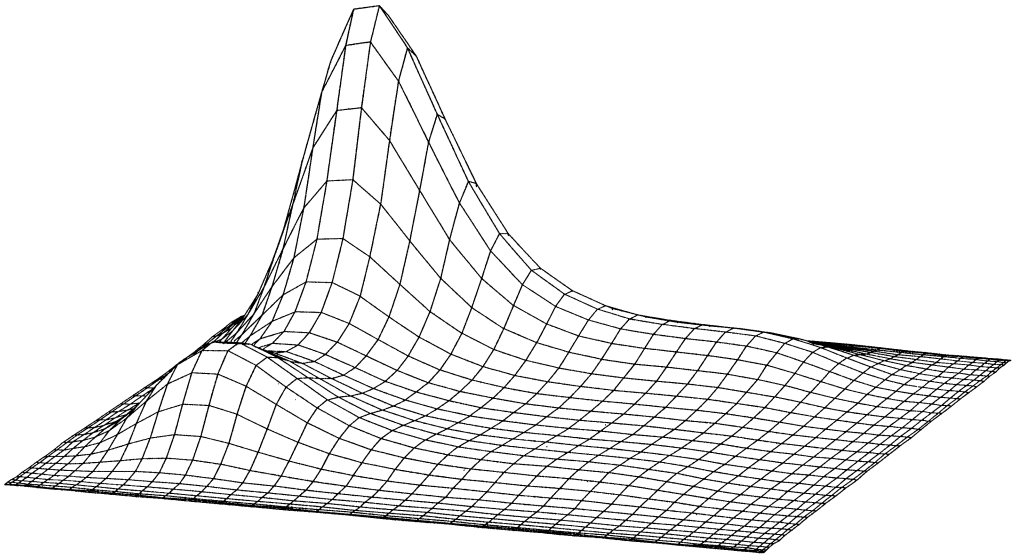


FIG. 8. *The estimate density for Algorithm* 2.

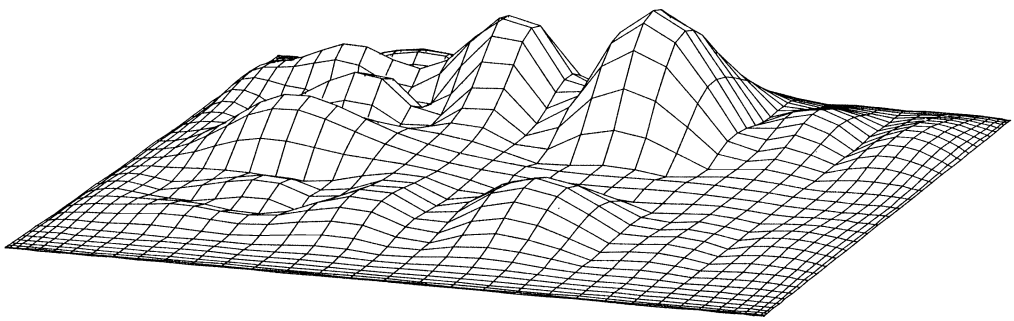


FIG. 9. *The supremum error function for Algorithm* 2.

TABLE 5
Numerical results for Example 5.

	sup-ERR	L_1-ERR	TIME	estimat.	sup-error
ALG 2.	0.26038	0.06210	0.22995	Fig.15	Fig.16
NEWTON(5)	0.24297	0.04099	40.3579	Fig.17	Fig.18
OPTIMAL	0.24160	0.04095		Fig.19	Fig.20

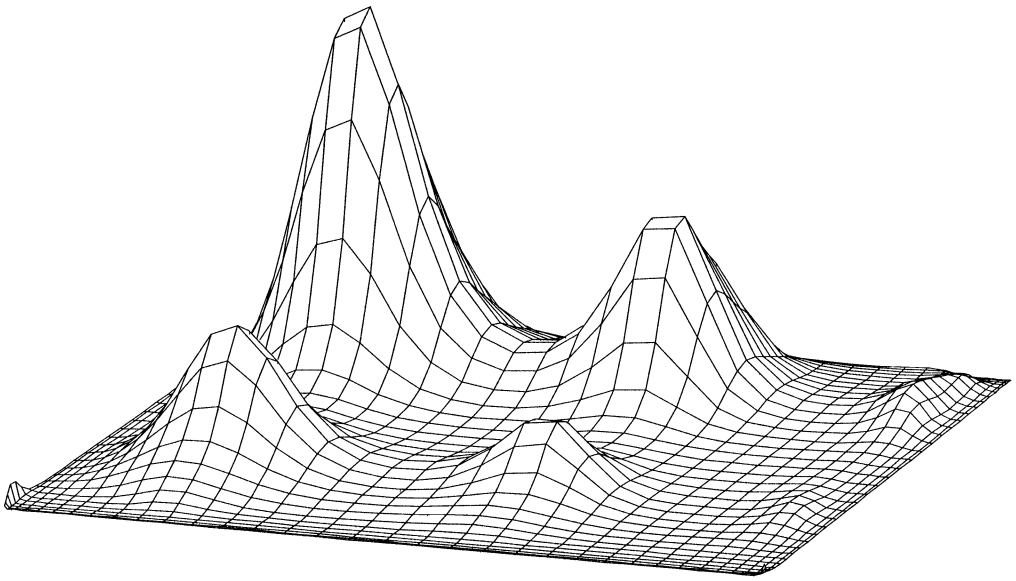


FIG. 10. *The estimate density after six more iterations using Newton's method.*

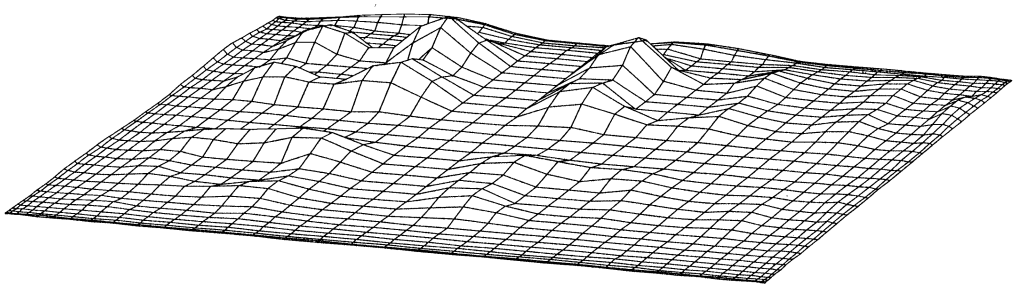


FIG. 11. *The supremum error function after doing six more iterations using Newton's method.*

We write

$$\tilde{b}_k = \int_T \exp\left[\sum_{i\in I_n} \lambda_i a_i(t)\right] a_k(t)dt, \qquad k \in I_n,$$

while

$$b_k = \int_T \bar{x}(t)a_k(t)dt, \qquad k \in I_n,$$

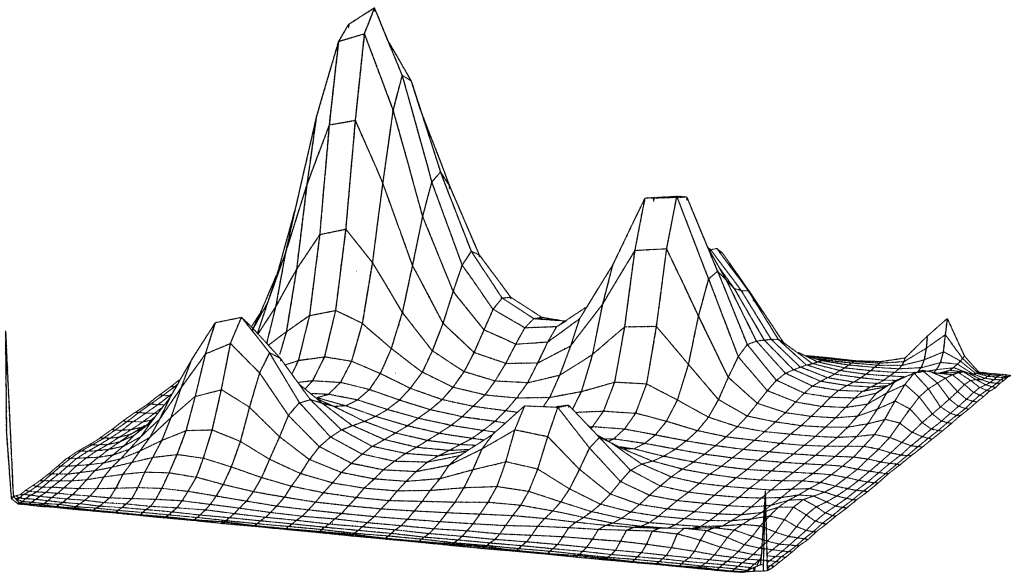


FIG. 12. *The optimal solution of* (P_n).

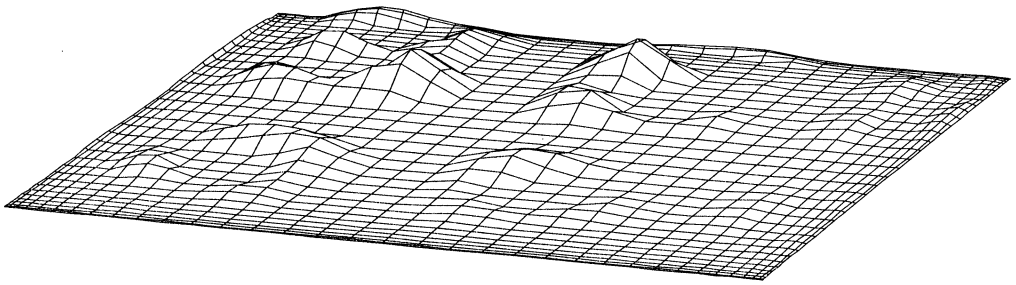


FIG. 13. *The supremum norm error function of the optimal solution of* (P_n).

and we denote $\widetilde{B}$ and B by the matrices generated in the algorithms using the data $\{\tilde{b}_i\}$ and $\{b_i\}$ respectively.

By the construction of the algorithms, λ can be determined by r, which solves a linear system

$$\tilde{B}r = \tilde{b}.$$

But from the input data $\{b_i\}$ and B, we can only obtain $\tilde{r}$, which solves the linear

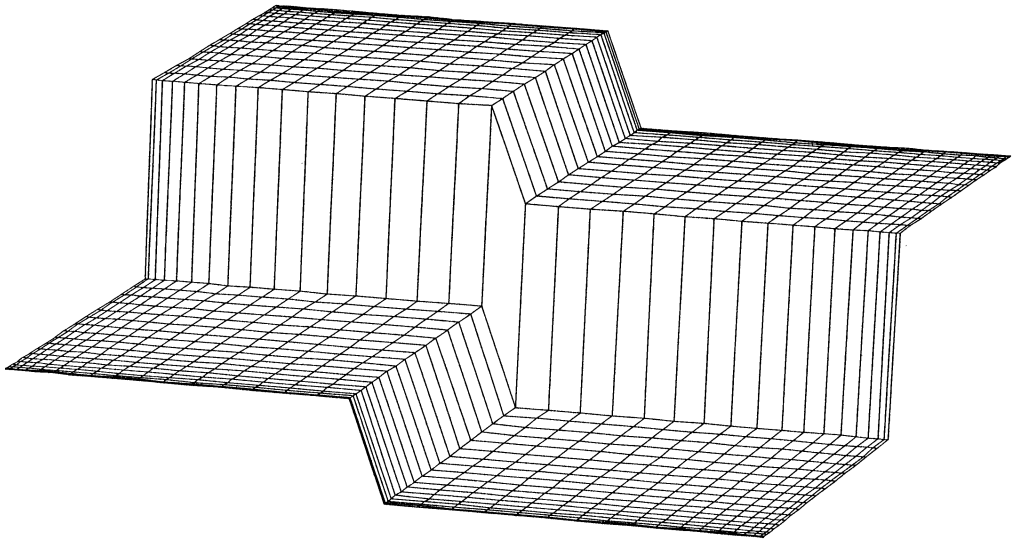


FIG. 14. *Prior function in Example* 5.

system

$$B\tilde{r} = b.$$

Since B is nonsingular under mild hypotheses, we can obtain $\tilde{r}$ and hence $\tilde{\lambda}$. We now need to estimate the error bounds of $\|\lambda - \tilde{\lambda}\|$ in some given norm. From the nonsingularity of the matrix B, it is easy to see that

$$\tilde{r} - r = B^{-1}(b - Br). \tag{13}$$

Case (A). Considering the algebraic case first, we have

$$\tilde{B} = \begin{bmatrix} 1 & \tilde{b}_1 & \tilde{b}_2 & \cdots & \tilde{b}_n \\ 1 & \tilde{b}_2 & \tilde{b}_3 & \cdots & \tilde{b}_{n+1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & \tilde{b}_{n+1} & \tilde{b}_{n+2} & \cdots & \tilde{b}_{2n} \end{bmatrix}, \qquad \tilde{b} = \begin{bmatrix} \tilde{b}_0 \\ 2\tilde{b}_1 \\ \vdots \\ (n+1)\tilde{b}_n \end{bmatrix},$$

$$B = \begin{bmatrix} 1 & b_1 & b_2 & \cdots & b_n \\ 1 & b_2 & b_3 & \cdots & b_{n+1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & b_{n+1} & b_{n+2} & \cdots & b_{2n} \end{bmatrix}, \qquad b = \begin{bmatrix} b_0 \\ 2b_1 \\ \vdots \\ (n+1)b_n \end{bmatrix}.$$

We assume

$$\bar{x}(t) = \exp\left[\sum_{i=0}^{n} \lambda_i t^i + \varepsilon_n(t)\right] \tag{14}$$

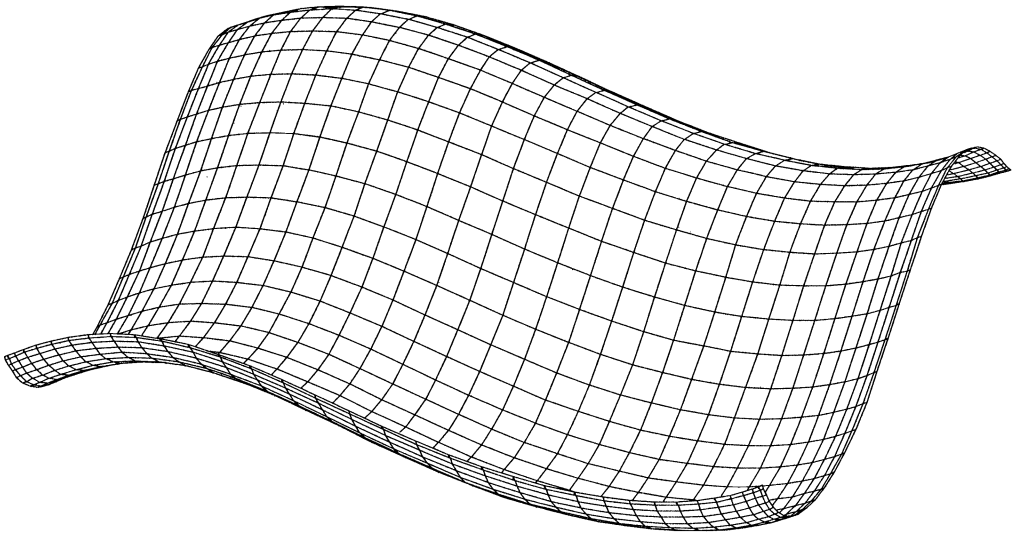


FIG. 15. *The estimate density for Algorithm* 1.

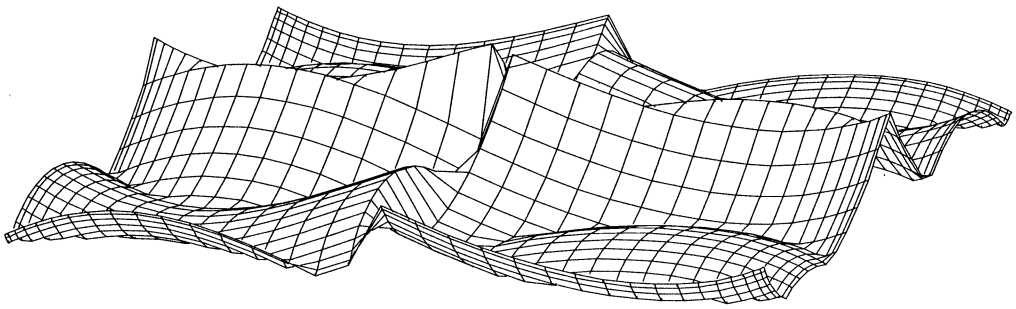


FIG. 16. *The supremum error function for Algorithm* 2.

and

$$\|\varepsilon_n(t)\|_\infty \leq \delta_n, \tag{15}$$

for $n = 0, 1, \ldots$.

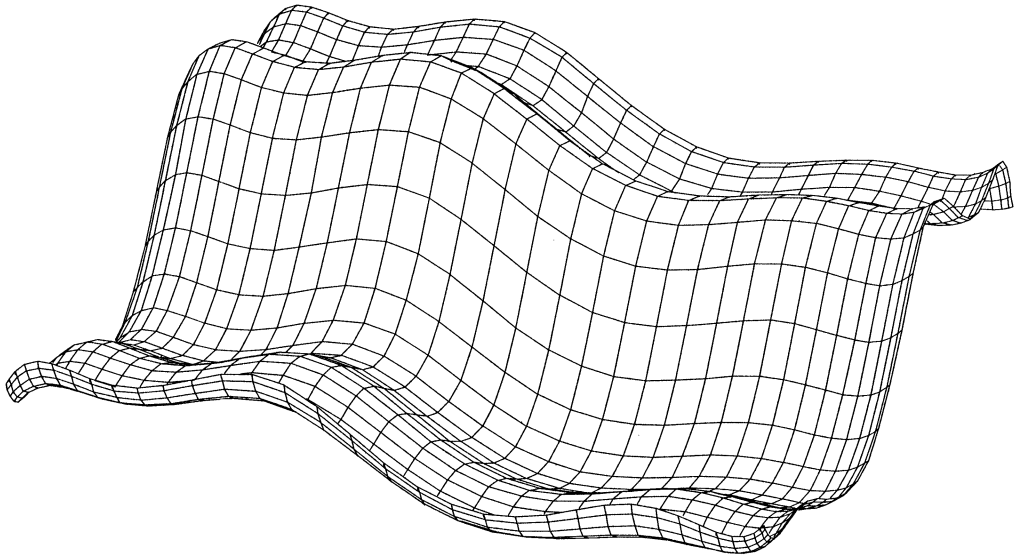


FIG. 17. *The estimate density after doing five more iterations using Newton's method.*

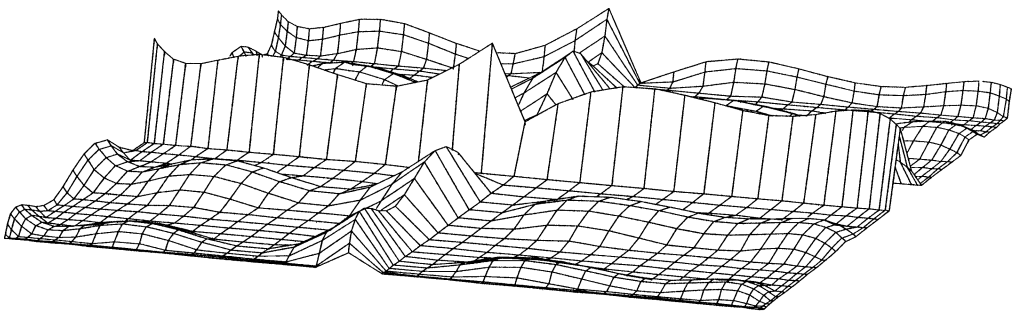


FIG. 18. *The supremum error function after doing five more iterations using Newton's method.*

Note that when $\varepsilon_n(\cdot)$ is differentiable on $[0,1]$

$$b_k \triangleq \int_0^1 \bar{x}(t)t^k dt$$
$$= \frac{1}{k+1}\bar{x}(1) - \frac{1}{k+1}\int_0^1 t^{k+1}\bar{x}(t)\left(\sum_{i=1}^{n} i\lambda_i t^{i-1} + \varepsilon_n'(t)\right)dt$$

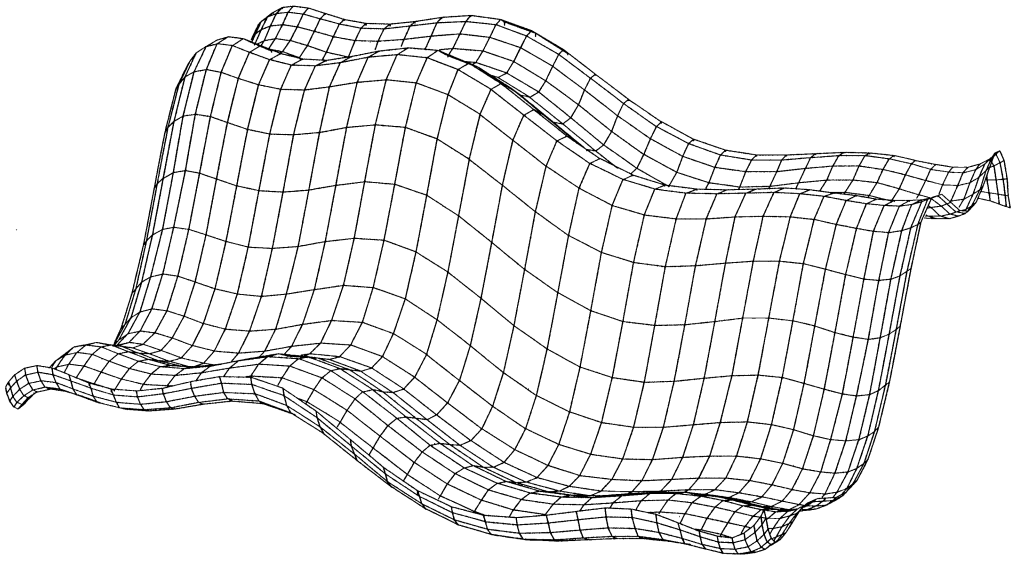


FIG. 19. *The optimal solution of* (P_n).

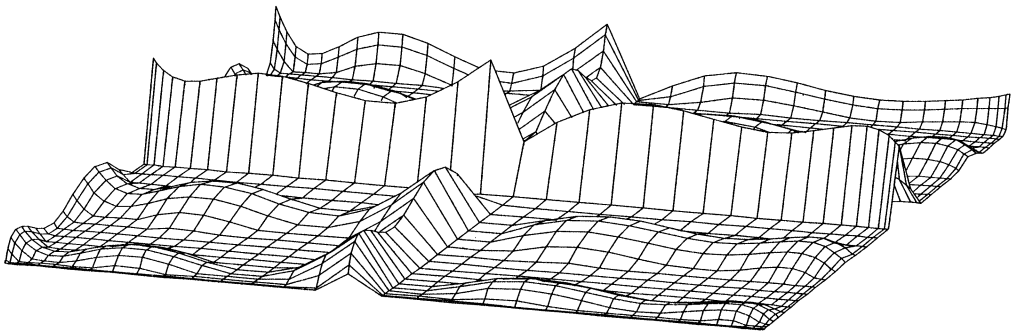


FIG. 20. *The supremum error function of the optimal solution of* (P_n).

$$= \frac{1}{k+1}\bar{x}(1) - \frac{1}{k+1}\sum_{i=1}^{n} i\lambda_i b_{i+k}$$

$$-\frac{1}{k+1}\int_0^1 t^{k+1}\bar{x}(t)\varepsilon_n'(t)dt.$$

Considering the kth component of $(b - Br)$ in (13), we have

$$\begin{aligned}(b - Br)_k &= (k+1)b_k + \sum_{i=1}^{n} i\lambda_i b_{k+i} - \exp\left[\sum_{i=0}^{n}\lambda_i\right] \\ &= \bar{x}(1)(1 - \exp[-\varepsilon_n(1)] - \varepsilon_n(1)\bar{x}(1) \\ &\quad + \int_0^1 \varepsilon_n(t)((k+1)\bar{x}(t)t^k + t^{k+1}\bar{x}'(t))dt \\ &= \bar{x}(1)[1 - \varepsilon_n(1) - \exp[-\varepsilon_n(1)]] \\ &\quad + \int_0^1 \varepsilon_n(t)((k+1)\bar{x}(t)t^k + t^{k+1}\bar{x}'(t))dt. \end{aligned} \tag{16}$$

LEMMA 2. *Suppose* $\bar{x} \in C^1[0,1]$ *is of the form in* (14), $\{a_i(t),\ i \in I_n\}$ *are algebraic polynomials* $1, t, \ldots, t^n$ *on* $[0,1]$. *Now* r, $\tilde{r}$, B, b *are as defined before. Then*

$$\|b - Br\|_\infty \le \left[\left(\frac{e^{\delta_{\max}}}{\delta_{\max}} + 2\right)\|\bar{x}\|_\infty + \|\bar{x}'\|_\infty\right]\delta_n,$$

where

$$\delta_{\max} = \max\{\delta_n,\ i = 0, 1, \ldots\},$$

hence

$$\|\tilde{r} - r\|_\infty \le C_1\|B^{-1}\|_\infty\delta_n,$$

where

$$C_1 = \left(\frac{e^{\delta_{\max}}}{\delta_{\max}} + 2\right)\|\bar{x}\|_\infty + \|\bar{x}'\|_\infty.$$

Proof. First we recall an inequality (proved in [2, Lem. 4.10]),

$$|e^x - 1| \le \frac{e^M - 1}{M}|x| \quad \text{for } |x| \le M. \tag{17}$$

By (16), we have

$$\begin{aligned}|(b - Br)_k| &\le \bar{x}(1)(e^{\delta_n} - 1) + |\varepsilon_n(1)|\bar{x}(1) \\ &\quad + \delta_n\int_0^1 |(k+1)\bar{x}(t)t^k + t^{k+1}\bar{x}'(t)|dt \\ &\le \|\bar{x}\|_\infty(e^{\delta_n} - 1) + \delta_n\|\bar{x}\|_\infty + \delta_n\left(\|\bar{x}\|_\infty + \|\bar{x}'\|_\infty\frac{1}{k+2}\right),\end{aligned}$$

and hence by (17),

$$\|b - Br\|_\infty \le \left[\left(\frac{e^{\delta_{\max}}}{\delta_{\max}} + 2\right)\|\bar{x}\|_\infty + \|\bar{x}'\|_\infty\right]\delta_n.$$

The result follows now from (13). □

From Lemma 2, we have, for $k = 1, 2, \ldots, n$,

$$\begin{aligned} |\lambda_k - \tilde{\lambda}_k| &= \frac{1}{k}|r_k - \tilde{r}_k| \\ &\leq \frac{1}{k}\|r - \tilde{r}\|_\infty \\ &\leq \frac{1}{k}\|B^{-1}\|_\infty C_1 \delta_n, \end{aligned} \tag{18}$$

for a constant C_1 depending on $\bar{x}$ and $\delta_{\max}$, but independent of n.

To estimate $|\lambda_0 - \tilde{\lambda}_0|$, we need the following mean value theorem.

LEMMA 3. *If $g(t) \geq 0$ is integrable, and $f(t) \geq 0$ is continuous on $[0,1]$, then there exists $\hat{t} \in [0,1]$, such that*

$$\int_0^1 f(t)g(t)dt = f(\hat{t})\int_0^1 g(t)dt.$$

We now give the error bound for $|\lambda_0 - \tilde{\lambda}_0|$. From the algorithm, we know that

$$e^{\tilde{\lambda}_0} = \frac{b_0}{\int_0^1 \exp\,[\sum_{i=1}^n \tilde{\lambda}_i t^i]dt}.$$

By (14) and Lemma 3, we have

$$\begin{aligned} \int_0^1 \exp\left[\sum_{i=1}^n \tilde{\lambda}_i t^i\right] dt &= \int_0^1 \exp\left[\sum_{i=1}^n \lambda_i t^i + \varepsilon_n(t)\right] \exp\left[\sum_{i=1}^n (\tilde{\lambda}_i - \lambda_i)t^i - \varepsilon_n(t)\right] dt \\ &= e^{-\lambda_0}\int_0^1 \bar{x}(t)\exp\left[\sum_{i=1}^n (\tilde{\lambda}_i - \lambda_i)t^i - \varepsilon_n(t)\right] dt \\ &= e^{-\lambda_0} b_0 \exp\left[\sum_{i=1}^n (\tilde{\lambda}_i - \lambda_i)\hat{t}^i - \varepsilon_n(\hat{t})\right] \end{aligned}$$

for some $\hat{t} \in [0,1]$. Thus

$$e^{\tilde{\lambda}_0} = \frac{e^{\lambda_0}}{\exp[\sum_{i=1}^n (\tilde{\lambda}_i - \lambda_i)\hat{t}^i - \varepsilon_n(\hat{t})]}$$

and

$$|\lambda_0 - \tilde{\lambda}_0| = |\sum_{i=1}^n (\tilde{\lambda}_i - \lambda_i)\hat{t}^i - \varepsilon_n(\hat{t})| \tag{19}$$

$$\leq \sum_{i=1}^n |\tilde{\lambda}_i - \lambda_i| + \delta_n \tag{20}$$

$$\leq (\sum_{i=1}^n \frac{1}{i})\|\tilde{r} - r\|_\infty + \delta_n \tag{21}$$

$$\leq \left(C_1\|B^{-1}\|_\infty \sum_{k=1}^n \frac{1}{k} + 1\right)\delta_n. \tag{22}$$

Noting that

$$\sum_{k=1}^{n} \frac{1}{k} < 1 + \int_1^n \frac{1}{x} dx = 1 + \log n,$$

we have the following theorem.

THEOREM 3. *Suppose that* $\log \bar{x} \in C^1[0,1]$, *that the moments are given by*

$$b_k = \int_0^1 \bar{x}(t) t^k dt, \qquad k = 0, 1, \ldots, n$$

and that the estimate density $\tilde{x}_n(t)$ *is computed by Algorithm* 1. *Then*

$$\|\tilde{x}_n - \bar{x}\|_\infty \leq \|\bar{x}\|_\infty (\exp(2E_n(C_1 \|B_n^{-1}\|_\infty (1 + \log n) + 1)) - 1),$$

where

$$E_n = \inf_{\lambda \in \mathbb{R}^{n+1}} \left\{ \| \log \bar{x} - \sum_{i=0}^{n} \lambda_i t^i \|_\infty \right\},$$

and C_1 *is a constant dependent only on* $\bar{x}$.

Proof. By the definition of E_n there exists $\lambda^n \in \mathbb{R}^{n+1}$ such that

$$\log \bar{x} = \sum_{i=0}^{n} \lambda_i^n t^i + \varepsilon_n(t)$$

and

$$\|\varepsilon_n(t)\|_\infty \leq \delta_n = E_n.$$

Using Algorithm 1, from (18) and (19), we have

$$|\lambda_k^n - \tilde{\lambda}_k^n| \leq \frac{1}{k} \|B_n^{-1}\|_\infty C_1 E_n$$

for $k = 1, 2, \ldots, n$, and

$$|\lambda_0^n - \tilde{\lambda}_0^n| \leq \left(C_1 \|B_n^{-1}\|_\infty \sum_{k=1}^{n} \frac{1}{k} + 1 \right) E_n.$$

Thus we have

$$\begin{aligned} \|\bar{x} - \tilde{x}_n\|_\infty &\leq \|\bar{x}\|_\infty \left\| 1 - \exp\left(\sum_{i=0}^{n} (\tilde{\lambda}_i^n - \lambda_i^n) t^i - \varepsilon_n(t) \right) \right\|_\infty \\ &\leq \|\bar{x}\|_\infty \left(\exp\left(\sum_{i=0}^{n} |\tilde{\lambda}_i^n - \lambda_i^n| + \delta_n \right) - 1 \right) \\ &\leq \|\bar{x}\|_\infty (\exp(2E_n(C_1 \|B_n^{-1}\|_\infty (1 + \log n) + 1)) - 1). \end{aligned}$$

□

Case (B). Similarly, for the trigonometric case, we assume $\bar{x}$ is of the form

$$\bar{x}(t) = \exp\left(\lambda_0 + \sum_{k=1}^{n} (\lambda_k \cos kt + \mu_k \sin kt) + \varepsilon_n(t) \right)$$

and

$$\|\varepsilon_n(t)\|_\infty \le \delta_n$$

for $n = 0, 1, \ldots$.

In the same way we proved for Lemma 2 , using the trigonometric angle formula, we note that,

$$\begin{aligned} a_k &\triangleq \int_{-\pi}^{\pi} \bar{x}(t) \cos kt dt \\ &= \frac{1}{k} \sin kt\bar{x}(t)|_{-\pi}^{\pi} - \frac{1}{k} \int_{-\pi}^{\pi} \sin kt\bar{x}'(t)dt \\ &= -\frac{1}{k} \int_{-\pi}^{\pi} \sin kt\bar{x}(t) \left(\sum_{j=1}^{n} (-j\lambda_j \sin jt + j\mu_j \cos jt) + \varepsilon_n'(t) \right) dt \\ &= \frac{1}{2k} \sum_{j=1}^{n} (j\lambda_j(a_{j-k} - a_{j+k}) + j\mu_j(b_{j-k} - b_{j+k})) \\ &\quad -\frac{1}{k} \int_{-\pi}^{\pi} \bar{x}(t) \sin kt\varepsilon_n'(t)dt. \end{aligned}$$

Hence for $k = 1, 2, \ldots, n$,

$$\begin{aligned} (b - Br)_k &\triangleq 2ka_k - \sum_{j=1}^{n} (a_{j-k} - a_{j+k}) j\lambda_j \\ &\quad + \sum_{j=1}^{n} (-b_{j-k} + b_{j+k}) j\mu_j \\ &= -\frac{1}{k} \int_{-\pi}^{\pi} \bar{x}(t) \sin kt\varepsilon_n'(t)dt \\ &= \frac{1}{k} \int_{-\pi}^{\pi} \varepsilon_n(t)(\bar{x}'(t) \sin kt + \bar{x}(t)k \cos kt)dt \end{aligned}$$

and

$$|(b - Br)_k| \le 2\pi\delta_n \left(\frac{1}{k} \|\bar{x}'\|_\infty + \|\bar{x}\|_\infty \right). \tag{23}$$

When we assume $\bar{x}(t)$ is periodic with the period 2π, that is

$$\bar{x}(-\pi) = \bar{x}(\pi),$$

then for $k = n + 1, \ldots, 2n$, we have the same inequality as (23). From Algorithm 4, we have

$$|\lambda_k - \tilde{\lambda}_k| \le \frac{1}{k} \|B^{-1}\|_\infty 2\pi\delta_n \left(\frac{1}{k} \|\bar{x}'\|_\infty + \|\bar{x}\|_\infty \right) \tag{24}$$

and

$$|\mu_k - \tilde{\mu}_k| \le \frac{1}{k} \|B^{-1}\|_\infty 2\pi\delta_n \left(\frac{1}{k} \|\bar{x}'\|_\infty + \|\bar{x}\|_\infty \right). \tag{25}$$

Here B is

$$\begin{bmatrix} A_1 - A_2 & -B_1 - B_2 \\ B_1 - B_2 & A_1 + A_2 \end{bmatrix}$$

constructed in Algorithm 4.

As to $|\lambda_0 - \tilde{\lambda}_0|$, note that in the algorithm, we have

$$e^{\tilde{\lambda}_0} = \frac{a_0}{\int_{-\pi}^{\pi} \exp(\sum_{j=1}^{n} \tilde{\lambda}_j \cos jt + \tilde{\mu}_j \sin jt) dt}.$$

Since

$$\begin{aligned} &\int_{-\pi}^{\pi} \exp\left(\sum_{j=1}^{n} \tilde{\lambda}_j \cos jt + \tilde{\mu}_j \sin jt\right) dt \\ &\quad = \int_{-\pi}^{\pi} \bar{x}(t) e^{-\lambda_0} \exp\left(\sum_{j=1}^{n} ((\tilde{\lambda}_j - \lambda_j) \cos jt + (\tilde{\mu}_j - \mu_j) \sin jt) - \varepsilon_n(t)\right) dt \\ &\quad = e^{-\lambda_0} \exp\left(\sum_{j=1}^{n} ((\tilde{\lambda}_j - \lambda_j) \cos j\hat{t} + (\tilde{\mu}_j - \mu_j) \sin j\hat{t}) - \varepsilon_n(\hat{t})\right) a_0 \end{aligned}$$

(by Lemma 3).

Thus

$$\begin{aligned} |\lambda_0 - \tilde{\lambda}_0| &= |\sum_{j=1}^{n} ((\tilde{\lambda}_j - \lambda_j) \cos j\hat{t} + (\tilde{\mu}_j - \mu_j) \sin j\hat{t}) - \varepsilon_n(\hat{t})| \\ &\leq \sum_{j=1}^{n} (|\tilde{\lambda}_j - \lambda_j| + |\tilde{\mu}_j - \mu_j|) + \delta_n. \end{aligned}$$

Combining this with (24) and (25), we have Theorem 4.

THEOREM 4. *Suppose that* $\log \bar{x} \in C^1[-\pi, \pi]$, $\bar{x}$ *is periodic with the period* 2π. *Given* $4n + 1$ *moments*

$$\begin{aligned} a_0 &= \int_{-\pi}^{\pi} \bar{x}(t) dt \\ a_k &= \int_{-\pi}^{\pi} \bar{x}(t) \cos kt dt \\ b_k &= \int_{-\pi}^{\pi} \bar{x}(t) \sin kt dt \\ &\qquad k = 1, 2, \ldots, 2n. \end{aligned}$$

Let $\tilde{x}_n(t)$ *be the estimate density constructed from the Algorithm* 4. *Then*

$$\|\tilde{x}_n - \bar{x}\|_\infty \leq \|\bar{x}\|_\infty (\exp(4\pi E_n \|B^{-1}\|_\infty (2\|\bar{x}'\|_\infty + \|\bar{x}\|_\infty (1 + \log n)) + E_n) - 1),$$

where

$$E_n \triangleq \inf \left\{ \|\log \bar{x} - \lambda_0 - \sum_{j=1}^{n} (\lambda_j \cos jt + \mu_j \sin jt)\|_\infty \mid (\lambda, \mu) \in \mathbb{R}^{2n+1} \right\}.$$

Proof. Similar to the proof to the Theorem 3. □

7. Conclusion. From the error bounds in Theorem 3, we see that the product

$$\|B_n^{-1}\|_\infty \cdot E_n$$

is an overestimate for the rate of the convergence of $\tilde{x}_n$ to $\bar{x}$. From approximation theory, Jackson's Theorem [7] tells us that if

$$\log \bar{x} \in C^r[0,1],$$

then

$$E_n = o\left(\frac{1}{n^r}\right).$$

Moreover, if $\log \bar{x}$ is analytic on $[0,1]$, then

$$E_n \leq Cq^n,$$

where C is a constant and $q < 1$. Unfortunately, we have not found any theoretical bound for $\|B_n^{-1}\|_\infty$. Numerical results indicate that

$$\|B_n^{-1}\|_\infty \to \infty,$$

(see Tables 6 and 7) and so when the number of moments gets too large, the computational results may not be reliable due to the accumulation of errors. But for the trigonometric case, when the prior density is smooth enough, we can see from Table 7 that $\|B_n^{-1}\|_\infty$ appears to be dominated by a polynomial, so that using Jackson's theorems, the convergence of our algorithm for trigonometric polynomial moments may follows.

TABLE 6
$\|B_n^{-1}\|_\infty$ *for algebraic moments and* $F_1 = 2|t - 0.5|$, $F_2 = \sin^2 t$, $F_3 = \chi_{[0.4,0.6]}$, $F_4 = t\sin^2(10t)$.

$\|B_n^{-1}\|_\infty$	F_1	F_2	F_3	F_4
n=2	0.320E02	0.469E02	0.498E02	0.477E02
3	0.578E03	0.240E04	0.219E05	0.195E04
4	0.206E05	0.105E06	0.121E08	0.732E05
5	0.608E06	0.450E07	0.107E11	0.363E07
6	0.231E08	0.160E09	0.895E13	0.151E09
7	0.594E09	0.560E10	0.746E16	0.416E10
8	0.223E11	0.209E12	0.866E19	0.152E12
9	0.667E12	0.717E13	0.904E22	0.642E13
10	0.233e14	0.233E15	0.110E26	0.176E15
11	0.675E15	0.844E16	0.179E29	0.840E16
12	0.250E17	0.289E18	0.355E32	0.274E18
13	0.758E18	0.948E19	0.965E35	0.856E19
14	0.257E20	0.326E21	0.451E38	0.326E21
15	0.816E21	0.112E23	—	0.949E22
16	0.289E23	0.372E24	—	0.374E24
17	0.885E24	0.124E26	—	0.123E26
18	0.307E26	0.428E27	—	0.422E27
19	0.989E27	0.144E29	—	0.158E29
20	0.342E29	0.470E30	—	0.477E30

Although the convergence of these algorithms is still unsettled, they often give very good estimates for the problem (P_n) and use much less time than Newton's method, as we saw in §6. If we use the heuristic solution as an initial estimate, then often only a couple of iterations are needed to get an almost optimal solution to (P_n).

TABLE 7
$\|B_n^{-1}\|_\infty$ *for trigonometric moments and* $F_1 = 1.5t$, $F_2 = \sin^2 t$, $F_3 = 0.8\chi_{[1.4,3.6]}$, $F_4 = t\sin^2(2t)$.

$\|B_n^{-1}\|_\infty$	F_1	F_2	F_3	F_4
n=3	0.40166	0.63662	0.932E00	0.12861
5	0.67313	0.63662	0.359E02	0.23679
7	0.99875	1.90986	0.311E04	0.50826
9	1.27501	1.90986	0.339E06	0.82539
11	1.55475	3.81972	0.375E08	0.91805
13	1.84964	3.81972	0.443E10	1.24088
15	2.16509	6.36620	0.667E12	1.77943
17	2.46792	6.36620	0.132E15	2.65173
19	2.77001	9.54940	0.229E17	2.93955
21	3.07217	9.54930	0.495E19	3.23528
23	3.36068	13.36902	0.151E22	4.00094
25	3.65724	13.36902	0.599E24	5.59461
27	3.96081	17.82575	0.253E27	6.09011
29	4.23835	17.82575	0.164E30	6.27491
31	4.53221	22.91831	0.194E33	7.28279
33	4.79253	22.91831	0.156E34	9.72277
35	5.08312	28.64789	0.158E34	10.49603
37	5.33845	28.64789	0.107E35	10.53350

REFERENCES

[1] J. M. BORWEIN AND A. S. LEWIS, *Convergence of best entropy estimates*, SIAM J. Optim., 1 (1991), pp. 191–205.

[2] ———, *Duality relationships for entropy-like minimization problems*, SIAM J. Control Theory Optim., 29 (1991), pp. 325–338.

[3] ———, *On the convergence of moment problems*, Trans. Amer. Math. Soc., 325 (1991), pp. 249–271.

[4] A. DECARREAU, D. HILHORST, C. LEMARÈCHAL, AND J. NAVAZA, *Dual methods in entropy maximization: application to some problems in crystallography*, SIAM J. Optim., 2 (1992), pp. 173–197.

[5] J. E. DENNIS AND R. B. SCHNABEL, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[6] D. G. LUENBERGER, *Optimization by Vector Space Methods*, John Wiley and Sons, Inc., New York, 1969.

[7] S. A. TELYAKOVSKII, *Research in the theory of approximation of functions at the mathematical institute of the academy of sciences*, Proceedings of the Steklov Institute of Mathematics, 1990.

SIAM J. OPTIMIZATION
Vol. 5, No. 1, pp. 100–113, February 1995

SYMMETRIC QUASIDEFINITE MATRICES*

ROBERT J. VANDERBEI†

Abstract. It is stated here that a symmetric matrix K is *quasidefinite* if it has the form

$$K = \begin{bmatrix} -E & A^T \\ A & F \end{bmatrix},$$

where E and F are symmetric positive definite matrices. Although such matrices are indefinite, it is shown that *any* symmetric permutation of a quasidefinite matrix yields a factorization LDL^T.

This result is applied to obtain a new approach for solving the symmetric indefinite systems arising in interior-point methods for linear and quadratic programming. These systems are typically solved either by reducing to a positive definite system or by performing a Bunch–Parlett factorization of the full indefinite system at every iteration. This is an intermediate approach based on reducing to a quasidefinite system. This approach entails less fill-in than further reducing to a positive definite system, but is based on a static ordering and is therefore more efficient than performing Bunch–Parlett factorizations of the original indefinite system.

Key words. matrix factorization, linear programming, interior-point methods

AMS subject classifications. primary 65F05; secondary 90C05

1. Introduction.

We call a symmetric matrix K *quasidefinite* if it has the form

$$K = \begin{bmatrix} -E & A^T \\ A & F \end{bmatrix},$$

where $E \in \Re^{n\times n}$ and $F \in \Re^{m\times m}$ are positive definite matrices with $m,\ n \geq 0$. The fact that quasidefinite matrices are nonsingular is trivial. To see it, consider the following system of equations:

$$\begin{bmatrix} -E & A^T \\ A & F \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} b \\ c \end{bmatrix}. \tag{1.1}$$

The positive definiteness of E allows us to use the first set of equations to solve for x in terms of y:

$$x = -E^{-1}(b - A^T y). \tag{1.2}$$

Substituting for x into the second set of equations yields

$$y = (F + AE^{-1}A^T)^{-1}(c + AE^{-1}b). \tag{1.3}$$

Again, the positive definiteness of E and F assures that the matrix $S := F + AE^{-1}A^T$ in (1.3) is also positive definite (and therefore nonsingular). Hence, there exists a unique solution to (1.1) for any b and c, which implies that K is nonsingular.

The above argument suggests an algorithm for solving (1.1). Namely, first solve for y using (1.3) and then solve for x using (1.2). However, when K is large and sparse,

* Received by the editors May 4, 1992; accepted for publication (in revised form) September 30, 1993.

† Program in Statistics and Operations Research, Princeton University, Princeton, New Jersey 08544 (rvdb@princeton.edu).

computational efficiency critically depends on maintaining sparsity in the matrix to be inverted in (1.3). Unfortunately, forming S can entail considerable fill-in. For example, if A has a single dense column, this matrix is completely full. Of course, one could try to solve the system in the other order, first solving for x directly and then solving for y as a function of x, but this approach encounters analogous inefficiencies when A has dense rows. In fact, both methods perform poorly when A has dense columns and dense rows.

A preferable approach is to apply an ordering heuristic (such as minimum-degree or minimum-local-fill) that prevents fill-in during the factorization of the *entire* matrix K. The caveat, however, is that *general* indefinite matrices are not guaranteed to be factorizable. The quasidefinite matrix K is indefinite, and so it is not clear a priori that one can (symmetrically) rearrange its rows and columns and factor the system in the resulting order. The fundamental result in this paper is that any symmetric permutation of a quasidefinite matrix is *guaranteed* to be factorizable.

It should be emphasized that no claim is made that all possible factorizations are equally stable numerically. Indeed, it is simple to give examples where one factorization is much worse than another (see §2). However, our aim is to apply the results presented here to the efficient implementation of interior-point methods for linear and quadratic programming, and in such cases we argue that there is not much disparity in the quality of the possible factorizations (where quality is measured by the relative sizes of the elements of the factorization). In fact, in the end they are all bad and yet it is rather remarkable that it is not difficult to obtain results with precision approximately equal to the square root of machine precision.

In situations where the relative sizes of the elements of a matrix factorization vary widely, it is important to limit the mixing of addition and subtraction operations in the calculation of the factors. This observation implies that, in the interior-point context, one should pivot out all the elements in either the upper left block (or the lower right block) first. While it is true that such strategies (which were called *conservative strategies* in [24]) are the safest, it is often possible to allow a little mixing in cases where such an allowance has a significant impact on the efficiency of the algorithm. The main result of this paper is that mathematically this poses no problem (but on a finite-precision machine one must be cautious about the degree of mixing allowed).

We end this section with a simple but important result.

THEOREM 1.1. *The inverse of a quasidefinite matrix is quasidefinite.*

Proof. Simple calculations show that

$$\begin{bmatrix} -E & A^T \\ A & F \end{bmatrix}^{-1} = \begin{bmatrix} -\bar{E} & \bar{A}^T \\ \bar{A} & \bar{F} \end{bmatrix},$$

where

$$\bar{E} = E^{-1} + E^{-1}A^T((F + AE^{-1}A^T)^{-1}AE^{-1}, \tag{1.4}$$

$$\bar{A} = (F + AE^{-1}A^T)^{-1}AE^{-1},$$

and

$$\bar{F} = (F + AE^{-1}A^T)^{-1}.$$

Applying the Sherman–Morrison–Woodbury formula to the right-hand side in (1.4), we see that

$$\bar{E} = (E + A^T F^{-1} A)^{-1}$$

and it follows that the inverse is quasidefinite. □

In the next section, we state and prove our main result. In §3, we apply this result to the system of equations arising in interior-point methods for mathematical programming. Then in §4 we present a simple example that illustrates the type of numerical difficulties that can arise. Finally, in §5, we present some numerical results.

2. The main result. We say that the symmetric nonsingular matrix K is *factorizable* if there exists a diagonal matrix D and a unit lower triangular matrix L such that $K = LDL^T$. The resulting pair (L, D) is then called a *factorization* of K. Furthermore, we say that the symmetric matrix K is *strongly factorizable* if every symmetric permutation of K yields a factorizable matrix. Thus, when K is strongly factorizable, there exists a factorization $PKP^T = LDL^T$ for any permutation P. To illustrate we present the following two examples.

Example 1. The matrix

$$\begin{bmatrix} 0 & 1 \\ 1 & 2 \end{bmatrix} \tag{2.1}$$

is not factorizable. To see this, note that for a 2×2 system LDL^T is simply

$$LDL^T = \begin{bmatrix} d_{11} & l_{21}d_{11} \\ l_{21}d_{11} & l_{21}^2 d_{11} + d_{22} \end{bmatrix}.$$

The zero in the upper left corner of (2.1) requires that $d_{11} = 0$, which in turn implies that

$$LDL^T = \begin{bmatrix} 0 & 0 \\ 0 & d_{22} \end{bmatrix}$$

for all unit lower triangular matrices L. Hence, it is clear that no factorization of (2.1) exists.

Example 2. The matrix

$$\begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix}$$

is factorizable but not strongly factorizable, since (2.1) is a symmetric permutation of this matrix.

When a factorization exists, it is unique. This is a well-known result. See, e.g., [8]. In the remainder of this section, we show that symmetric quasidefinite matrices form a class of strongly factorizable matrices.

THEOREM 2.1. *Symmetric quasidefinite matrices are strongly factorizable.*

Proof. Fix a permutation matrix P. It suffices to show that the leading principal submatrices of PKP^T are nonsingular. But these submatrices are of the form $\bar{P}K_S\bar{P}^T$, where K_S is a principal submatrix of K and $\bar{P}$ is a permutation matrix. A principal submatrix K_S is of the form

$$K_S = \begin{bmatrix} -E_S & A_S^T \\ A_S & F_S \end{bmatrix}, \tag{2.2}$$

where E_S and F_S are principal submatrices of E and F, respectively, and so are positive definite. Thus K_S is quasidefinite and hence nonsingular, as required. □

There exist symmetric quasidefinite matrices for which some permutations yield much better factorizations than others. For example, consider

$$\begin{bmatrix} -\epsilon & 1 \\ 1 & 1 \end{bmatrix}, \tag{2.3}$$

where $\epsilon > 0$ is a small number. This matrix is symmetric quasidefinite and hence is strongly factorizable, but the two possible factorizations (corresponding to the matrix itself and its symmetric permutation) have very different properties. Indeed, factoring the matrix as given yields

$$D = \begin{bmatrix} -\epsilon & 0 \\ 0 & 1+\frac{1}{\epsilon} \end{bmatrix}, \quad L = \begin{bmatrix} 1 & 0 \\ -\frac{1}{\epsilon} & 1 \end{bmatrix}, \tag{2.4}$$

whereas factoring the symmetrically permuted matrix gives

$$D = \begin{bmatrix} 1 & 0 \\ 0 & -(1+\epsilon) \end{bmatrix}, \quad L = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}. \tag{2.5}$$

It is clear that (2.5) is much better than (2.4). To quantify this notion, we introduce an a priori measure of stability τ defined as

$$\tau = \frac{\| \, |L| \, |D| \, |L^T| \, \|}{\|K\|}, \tag{2.6}$$

where $\|\cdot\|$ denotes the L^∞ matrix norm and $|\cdot|$ denotes the matrix whose elements are the absolute values of the indicated matrix. If τ is close to one, the factorization is stable (see [10, p. 136]). Larger values indicate less stability. For the matrix in (2.3), $\tau = 1 + 1/\epsilon$ whereas for its symmetric permutation we get $\tau = (3+\epsilon)/2$, which is clearly much better.

Our primary interest in symmetric quasidefinite matrices arises in the context of interior-point methods for linear and quadratic programming. We show in §3 that matrices such as the one considered here do not arise in that context.

3. Application to interior-point methods. Consider the following linear programming problem

$$\begin{aligned} &\text{minimize} && c^T x \\ &\text{subject to} && Ax \geq b, \\ & && x \geq 0, \end{aligned} \tag{3.1}$$

where A is an $m \times n$ matrix, c is an n-vector, and b is an m-vector. The dual of this problem is

$$\begin{aligned} &\text{maximize} && b^T y \\ &\text{subject to} && A^T y \leq c, \\ & && y \geq 0. \end{aligned}$$

Adding surplus variables w to the primal and slack variables z to the dual, we can rewrite the problems as follows

$$\begin{aligned} &\text{minimize} && c^T x \\ &\text{subject to} && Ax - w = b, \\ & && x, w \geq 0 \end{aligned} \tag{3.2}$$

and

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

For problems presented in this form, the system of equations that must be solved at each iteration of the interior-point algorithm has the following form involving a quasidefinite matrix:

$$\begin{bmatrix} -X^{-1}Z & A^T \\ A & Y^{-1}W \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \sigma \\ \rho \end{bmatrix}, \tag{3.3}$$

where X, Y, W, and Z denote diagonal matrices with the components of x, y, w, and z on their diagonals, respectively (and, as an interior-point method, all the components of these vectors are strictly positive at every iteration).

For interior-point methods, the main computational burden lies in solving systems of the form in (3.3). Early implementations for linear programming did not operate on system (3.3) directly, but rather dealt with the symmetric positive definite system obtained from (3.3) by solving for Δx in terms of Δy using the first set of equations and then eliminating Δx from the second set:

$$\Delta x = -XZ^{-1}(\sigma - A^T \Delta y) \tag{3.4}$$

and

$$(AXZ^{-1}A^T + Y^{-1}W)\Delta y = (\rho + AXZ^{-1}\sigma). \tag{3.5}$$

The advantage of this approach is that the matrix $AXZ^{-1}A^T + Y^{-1}W$ is symmetric and positive definite, so that well-known and well-behaved methods such as Cholesky factorization (which was used in the implementations described in [4], [12]–[15], [17], [18], [21], [23]) or preconditioned conjugate gradient (used in the implementations described in [4], [11], [16]) can be used to solve systems involving this matrix. However, the disadvantage is that $AXZ^{-1}A^T$ can be quite dense compared to A if A has dense columns.

Recent papers have suggested that it might be better to solve indefinite systems such as (3.3) at every iteration. This suggestion was first put forth by researchers in Stanford's Systems Optimization Laboratory ([5], [19], [9]) and by Turner [20]. Subsequently, Fourer and Mehrotra [6] began experimenting with the indefinite system approach. All of these papers rely on doing a Bunch–Parlett ([3], [2]) factorization of the indefinite system.

Solving the indefinite system mitigates the fill-in caused when dense columns are present, but Bunch–Parlett factorizations tend to be more computationally burdensome. As such, solving the indefinite system offers an advantage when dense columns are present, but tends to be slower on most other problems.

We apply Theorem 2.1 to obtain a robust procedure that is not hampered by dense columns. Unfortunately, linear programs as formulated usually do not fit directly into the form given in (3.1). For example, some of the variables might not be constrained to be nonnegative (we call these *free variables*), and some of the constraints might be equality instead of inequality constraints. It turns out that the algorithm can be modified to handle free variables as follows. For each free variable x_j, simply set the

corresponding dual slack variable z_j to zero everywhere it appears in the algorithm. Similarly, equality constraints are handled by setting w_i equal to zero for each equality constraint. This makes for a very simple algorithm, but regretably the matrix

$$K = \begin{bmatrix} -X^{-1}Z & A^T \\ A & Y^{-1}W \end{bmatrix} \tag{3.6}$$

is no longer quasidefinite as zeros have now appeared on the main diagonal. This problem is handled by introducing a two-tiered elimination scheme. In the first tier, we select pivot elements associated with some (or maybe even all) of the nonzero diagonal elements in (3.6). As we shall show, pivoting on these elements in any order is safe. Furthermore, the reduced system produced by symmetric Gaussian elimination is eventually guaranteed to be itself a quasidefinite system and so from that point on we can enter tier two and choose the remaining pivots in any order.

To make the above explanation more precise, we partition $X^{-1}Z$ and $Y^{-1}W$ into 2×2 blocks

$$X^{-1}Z = \begin{bmatrix} E_1 & \\ & E_2 \end{bmatrix},$$

$$Y^{-1}W = \begin{bmatrix} F_1 & \\ & F_2 \end{bmatrix},$$

putting all the zero elements (and perhaps some nonzeros) of $X^{-1}Z$ into E_2 and all the zero elements (and perhaps some nonzeros) of $Y^{-1}W$ into F_2. Then we partition system (3.3)

$$\begin{bmatrix} -E_1 & & A_{11}^T & A_{21}^T \\ & -E_2 & A_{12}^T & A_{22}^T \\ A_{11} & A_{12} & F_1 & \\ A_{21} & A_{22} & & F_2 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta y_1 \\ \Delta y_2 \end{bmatrix} = \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \rho_1 \\ \rho_2 \end{bmatrix}. \tag{3.7}$$

Since E_1 and F_1 are positive definite, we move those blocks to the upper left-hand corner

$$\begin{bmatrix} -E_1 & A_{11}^T & & A_{21}^T \\ A_{11} & F_1 & A_{12} & \\ & A_{12}^T & -E_2 & A_{22}^T \\ A_{21} & & A_{22} & F_2 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta y_1 \\ \Delta x_2 \\ \Delta y_2 \end{bmatrix} = \begin{bmatrix} \sigma_1 \\ \rho_1 \\ \sigma_2 \\ \rho_2 \end{bmatrix}. \tag{3.8}$$

Now, the upper left 2×2 block is quasidefinite and so can be used to solve (with pivots in any order) for Δx_1 and Δy_1:

$$\begin{bmatrix} \Delta x_1 \\ \Delta y_1 \end{bmatrix} = \begin{bmatrix} -E_1 & A_{11}^T \\ A_{11} & F_1 \end{bmatrix}^{-1} \left(\begin{bmatrix} \sigma_1 \\ \rho_1 \end{bmatrix} - \begin{bmatrix} & A_{21}^T \\ A_{12} & \end{bmatrix} \begin{bmatrix} \Delta x_2 \\ \Delta y_2 \end{bmatrix} \right).$$

Substituting this into the last equations in (3.8), we get the following system for Δx_2 and Δy_2:

$$\begin{aligned} &\left(\begin{bmatrix} -E_2 & A_{22}^T \\ A_{22} & F_2 \end{bmatrix} - \begin{bmatrix} & A_{12}^T \\ A_{21} & \end{bmatrix} \begin{bmatrix} -E_1 & A_{11}^T \\ A_{11} & F_1 \end{bmatrix}^{-1} \begin{bmatrix} & A_{21}^T \\ A_{12} & \end{bmatrix} \right) \begin{bmatrix} \Delta x_2 \\ \Delta y_2 \end{bmatrix} \\ &\qquad = \begin{bmatrix} \sigma_2 \\ \rho_2 \end{bmatrix} - \begin{bmatrix} & A_{12}^T \\ A_{21} & \end{bmatrix} \begin{bmatrix} -E_1 & A_{11}^T \\ A_{11} & F_1 \end{bmatrix}^{-1} \begin{bmatrix} \sigma_1 \\ \rho_1 \end{bmatrix}. \end{aligned} \tag{3.9}$$

In Theorem 1.1, we showed that the inverse of a quasidefinite matrix is quasidefinite. Hence, we introduce the following notation for the inverse appearing above:

$$\begin{bmatrix} -E_1 & A_{11}^T \\ A_{11} & F_1 \end{bmatrix}^{-1} = \begin{bmatrix} -\bar{E}_1 & \bar{A}_{11}^T \\ \bar{A}_{11} & \bar{F}_1 \end{bmatrix}.$$

Then the triple matrix product in (3.9) can be written as

$$\begin{bmatrix} & A_{12}^T \\ A_{21} & \end{bmatrix} \begin{bmatrix} -E_1 & A_{11}^T \\ A_{11} & F_1 \end{bmatrix}^{-1} \begin{bmatrix} & A_{21}^T \\ A_{12} & \end{bmatrix} = \begin{bmatrix} A_{12}^T \bar{F}_1 A_{12} & A_{12}^T \bar{A}_{11} A_{21}^T \\ A_{21} \bar{A}_{11}^T A_{12} & -A_{21} \bar{E}_1 A_{21}^T \end{bmatrix},$$

and we see that the system in (3.9) is quasidefinite if and only if

$$E_2 + A_{12}^T \bar{F}_1 A_{12} \quad \text{and} \quad F_2 + A_{21} \bar{E}_1 A_{21}^T$$

are positive definite. Clearly, the larger the dimension of E_1 and F_1, the greater the likelihood for this.

To summarize, our procedure is based on partitioning the rows and columns of A into two tiers. Elements belonging to the first tier are eliminated first (in any order) and then elements from the second tier are eliminated. As long as the tiers are chosen appropriately, this scheme is guaranteed to work. While it is certainly possible to go through once at the beginning and ensure that enough elements are put into tier one, experience has shown that simple, conservative heuristics work just as well (as long as they are conservative). For example, our code, which is called LOQO and is described in [22], [24], uses such a conservative approach. This code actually uses four tiers. The first tier corresponds to all variables except those that are free variables and those associated with dense columns. The second tier consists of all inequality constraints and the dense columns. The third tier then has the equality constraints and the fourth tier the free variables. Within tiers, elimination order is determined by one of the usual heuristics such as minimum-degree or minimum-local-fill. The heuristic for determining which columns to call dense works as follows. First, out of the n columns, look at the m sparsest. Multiply the density of the densest of these m columns by a number larger than one (10 is the default) and declare a column to be dense if and only if its density exceeds this threshold.

We now return to the question of numerical stability in the context of interior-point methods. It was proved in [1] that strict complementarity holds in the limit (at least in the case of continuous trajectories of the affine scaling algorithm but it seems to be true in general). In the present context, this means that the diagonal elements of $X^{-1}Z$ and $Y^{-1}W$ all tend to zero or infinity. In fact, numerical experience indicates that the rate at which the elements tend to zero or to infinity is the same from one element to the next. Hence, 2×2 matrices such as (2.3) do not arise. Instead, 2×2 matrices where both the diagonal elements go to zero, both go to infinity, or one goes to zero and the other goes to infinity are more relevant. For

$$\begin{bmatrix} -\epsilon & 1 \\ 1 & \epsilon \end{bmatrix},$$

we get $\tau = (1 + \epsilon + 2/\epsilon)/(1 + \epsilon) \approx 2/\epsilon$ and for its symmetric permutation τ is the same. On the other hand, for

$$\begin{bmatrix} -\epsilon & 1 \\ 1 & \frac{1}{\epsilon} \end{bmatrix},$$

we get $\tau = (1 + 3/\epsilon)/(1 + 1/\epsilon) \approx 3$ and for its symmetric permutation $\tau = 1$. The other cases are similar. What one observes is that the level of instability is essentially independent of the permutation.

However, the value of τ does not tell the whole story. In the next section, we consider a specific example that illustrates the situations that can arise.

4. An example. Consider the following linear programming problem:

$$\text{minimize } x_1 + x_2, \tag{4.1}$$

$$x_1 + 2x_2 \geq 1,$$
$$2x_1 + x_2 \geq 1,$$

$$x_1 \geq 0, \quad x_2 \text{ free.}$$

For this problem, the symmetric indefinite system that must be solved at each iteration involves a matrix whose lower triangular part has the following form:

$$\begin{bmatrix} -\epsilon_1 & & & \\ & 0 & & \\ 1 & 2 & \delta_1 & \\ 2 & 1 & & \delta_2 \end{bmatrix},$$

where ϵ_1, δ_1, and δ_2 are all positive and tending to zero (at roughly the same rate) as the iterations progress (the zero on the second diagonal position arises from the fact that x_2 is a free variable). The zero diagonal element could pose a problem and so any static ordering must anticipate this and defer this pivot till the end. Therefore, the lower triangular part of the matrix becomes

$$\begin{bmatrix} -\epsilon_1 & & & \\ 1 & \delta_1 & & \\ 2 & & \delta_2 & \\ & 2 & 1 & 0 \end{bmatrix}.$$

This matrix is factorizable since after the first pivot, the reduced matrix is symmetric quasidefinite. However, to see what happens, consider applying symmetric Gaussian elimination to this matrix. After eliminating the nonzeros under the first two columns, the lower triangle of the resulting 2×2 submatrix becomes

$$\begin{bmatrix} (\delta_2 + \frac{4}{\epsilon_1}) - \frac{4}{\epsilon_1^2(\delta_1 + \frac{1}{\epsilon_1})} & \\ 1 - \frac{4}{\epsilon_1(\delta_1 + \frac{1}{\epsilon_1})} & \frac{4}{(\delta_1 + \frac{1}{\epsilon_1})} \end{bmatrix}. \tag{4.2}$$

In the elimination process, the parenthesized expressions are evaluated before the other operations. Hence, in finite precision arithmetic, once each of the small parameters becomes smaller than the square root of the arithmetic's precision, these expressions simplify to

$$\delta_2 + \frac{4}{\epsilon_1} = \frac{4}{\epsilon_1} \quad \text{and} \quad \delta_1 + \frac{1}{\epsilon_1} = \frac{1}{\epsilon_1},$$

and so (4.2) becomes

$$\begin{bmatrix} 0 & \\ -3 & 4\epsilon_1 \end{bmatrix},$$

which clearly presents a problem for the next stage of the elimination. However, using exact arithmetic, (4.2) simplifies to

$$\begin{bmatrix} \delta_2 + \frac{4\delta_1}{1+\epsilon_1\delta_1} & \\ 1 - \frac{4}{1+\epsilon_1\delta_1} & \frac{4\epsilon_1}{1+\epsilon_1\delta_1} \end{bmatrix}. \tag{4.3}$$

From this exact expression it is clear that the given order and the order obtained by interchanging the last two pivots generate similar values for τ (since both diagonal entries are of the same magnitude). Hence, our estimate of instability τ, defined by (2.6), fails to differentiate between these two permutations. What has gone wrong is the common problem of mixing addition and subtraction of large numbers.

In addition, (4.3) suggests that whenever an exact zero appears on a diagonal it might be a good idea to set the value to either plus or minus the square root of the arithmetic's precision. In fact, this is what is done in our code (which is described in [24]) and we are able to solve (4.1) to full precision.

This example shows that instability can occur. However, when it does, one can still expect to get results as accurate as the square root of machine precision. This seems to hold true even for large problems. A partial explanation for this is the fact that interior-point methods have a certain autoscaling property (i.e., they attempt to follow the central trajectory), which helps to make the diagonal elements go to zero or to infinity all at the same rate.

5. Numerical experiments and conclusions. Using our code, we computed the ratio of the largest to the smallest of the absolute values of the diagonal elements of D on the last iteration of the algorithm. On the eighty or so test problems in the NETLIB suite [7] this ratio ranged between `1.0e+19` and infinity (infinity means that an exact zero appeared on the diagonal, which can happen when rank deficiency occurs due to primal degeneracy). These ratios are tabulated in Tables 1 and 2. Given such large values for this ratio, it is quite remarkable that the code was able to solve all but two problems (greenbeb and pilot87) to eight significant figures of accuracy (and pilot87 stopped just short with seven significant figures). These tests were performed on an IBM RS 6000, which implements the IEEE floating-point standard and therefore has 15 digits of precision (53 bits). It is also interesting to note that the two that ran into numerical trouble were not necessarily the ones with the largest ratios. It turns out that for many problems in this suite the matrix to which K in (3.6) is converging is actually a singular matrix (due to primal or dual degeneracy) and so numerical difficulties will exist regardless of the ordering.

We also computed the value of τ, defined by (2.6), for each of the test problems. For these computations, we scaled the matrix K by dividing each row and each column by the maximum between `1.0` and the square root of the corresponding diagonal element. Tables 3 and 4 show the value of τ on the last iteration. It turns out that in every case this was the largest value over all iterations of the algorithm. Again there seems to be no correlation between those problems that encountered numerical difficulties and those that had large τ values. This lack of correlation gives credence to the notion that numerical difficulty arises primarily from primal and dual degeneracy.

TABLE 1
Ratios of diagonals in factorization (problems 1-p).

Problem name	Ratio	Iterations	Primal infeas	Dual infeas	Significant figures
25fv47	5.8709e+37	29	4.61e−13	2.56e−13	8
80bau3b	Infinity	43	6.43e−10	1.28e−11	8
adlittle	2.2458e+21	14	8.87e−11	1.74e−16	8
afiro	4.1738e+24	13	6.00e−14	8.98e−14	8
agg	7.8694e+41	26	3.98e−12	3.62e−09	8
agg2	1.1431e+32	22	1.09e−14	4.99e−12	8
agg3	4.8834e+34	22	5.34e−15	5.24e−12	8
bandm	8.6392e+27	20	8.74e−11	5.95e−13	9
beaconfd	1.9200e+28	14	7.95e−11	6.14e−12	8
blend	5.8329e+24	14	1.90e−12	6.75e−11	9
bnl1	1.0244e+35	35	2.39e−12	2.23e−12	8
bnl2	3.4315e+54	40	1.62e−09	1.94e−11	8
boeing1	1.0681e+41	28	1.26e−08	2.09e−13	9
boeing2	5.1782e+38	28	1.47e−15	1.71e−10	8
bore3d	Infinity	17	1.72e−08	2.48e−16	8
brandy	3.3141e+31	22	7.33e−08	1.97e−11	9
capri	8.8349e+27	23	1.07e−12	4.52e−11	8
cycle	6.2422e+36	32	5.75e−09	3.27e−11	9
czprob	6.7245e+30	38	1.15e−11	1.08e−13	8
d2q06c	2.8169e+47	38	2.24e−10	1.50e−09	8
degen2	1.0067e+20	14	5.94e−10	3.75e−16	8
degen3	Infinity	17	9.54e−10	2.94e−12	8
e226	8.4768e+33	22	5.12e−13	9.02e−14	9
etamacro	2.0071e+31	30	3.19e−13	1.62e−14	8
fffff800	1.7007e+39	36	3.97e−13	2.53e−08	8
finnis	8.9977e+33	26	1.00e−13	4.91e−14	8
fit1d	1.6286e+22	21	4.81e−08	3.61e−15	9
fit1p	3.5353e+24	26	5.68e−10	5.52e−12	8
fit2d	5.1893e+19	24	1.50e−08	1.70e−16	8
fit2p	6.3007e+23	24	4.24e−11	1.86e−12	8
forplan	4.8253e+40	29	7.28e−14	1.33e−10	8
ganges	3.3747e+33	23	5.06e−12	3.88e−11	9
gfrdpnc	2.9347e+25	19	1.82e−10	3.61e−14	8
greenbea	8.1336e+44	50	2.30e−06	2.86e−12	8
greenbeb	2.8818e+28	30	1.80e−06	1.82e−10	3
grow15	2.3749e+33	23	2.35e−06	7.18e−14	10
grow22	3.1747e+33	27	7.99e−06	9.88e−15	10
grow7	6.0359e+31	20	2.62e−06	3.69e−13	10
israel	5.7337e+28	28	3.58e−16	4.36e−15	9
kb2	3.7487e+26	20	2.54e−06	5.84e−10	8
lotfi	1.1710e+44	25	1.83e−14	7.60e−12	8
maros	9.9261e+31	28	3.27e−10	9.37e−10	8
nesm	3.3790e+25	37	9.48e−13	2.61e−14	8
perold	1.7497e+36	39	1.79e−13	5.11e−11	9
pilot4	5.9526e+36	38	1.81e−11	1.51e−10	8
pilot87	1.7318e+50	45	3.45e−12	7.84e−10	7
pilotja	1.1399e+37	38	3.06e−12	5.56e−10	8
pilotnov	1.1636e+36	29	2.24e−11	6.77e−11	8
pilots	1.1568e+45	44	5.17e−12	1.07e−08	8
pilotwe	2.8908e+32	39	7.47e−12	2.37e−14	8

TABLE 2
Ratios of diagonals in factorization (problems r-z).

Problem name	Ratio	Iterations	Primal infeas	Dual infeas	Significant figures
recipe	1.6775e+31	13	3.21e−08	7.81e−11	9
sc105	6.9689e+26	14	1.89e−13	1.56e−12	8
sc205	8.9380e+30	17	2.38e−14	4.35e−12	9
sc50a	4.2398e+26	14	2.02e−14	9.54e−13	9
sc50b	2.2211e+26	13	1.20e−14	5.61e−13	9
scagr25	3.4668e+24	18	5.96e−13	8.69e−14	9
scagr7	7.6158e+21	15	2.55e−13	4.72e−12	8
scfxm1	2.0143e+33	26	6.42e−10	3.19e−10	8
scfxm2	4.1487e+35	28	2.05e−08	1.80e−11	9
scfxm3	1.2522e+36	28	2.29e−07	3.43e−11	8
scorpion	Infinity	16	1.66e−10	1.35e−15	8
scrs8	4.1479e+39	23	1.11e−10	1.90e−16	8
scsd1	3.9397e+20	15	1.04e−11	2.30e−16	9
scsd6	3.3169e+19	17	7.44e−11	3.50e−16	8
scsd8	9.2737e+19	17	2.43e−10	8.85e−16	8
sctap1	6.3855e+21	18	6.23e−11	3.55e−13	8
sctap2	Infinity	16	2.84e−10	1.43e−12	8
sctap3	9.0610e+18	16	1.76e−10	1.51e−12	8
seba	1.3798e+27	19	8.25e−09	4.10e−12	8
share1b	8.1766e+33	26	1.25e−10	2.12e−11	8
share2b	1.1939e+22	14	6.02e−09	1.19e−14	8
shell	5.0921e+27	25	5.99e−14	3.11e−12	8
ship04l	3.2699e+26	20	9.49e−11	1.22e−15	9
ship04s	6.1811e+25	19	1.76e−10	1.38e−15	8
ship08l	3.5321e+27	20	6.52e−11	2.02e−15	8
ship08s	4.0247e+27	20	4.07e−11	2.60e−15	9
ship12l	1.9722e+27	24	2.35e−10	4.88e−15	8
ship12s	4.9804e+27	22	1.72e−10	1.27e−14	8
sierra	1.2960e+34	21	1.06e−11	1.83e−12	8
stair	1.4413e+30	21	2.63e−11	1.78e−12	9
standata	4.0684e+30	23	1.34e−12	8.55e−14	9
standmps	1.0390e+34	32	3.12e−13	9.57e−13	8
stocfor1	7.6992e+24	17	2.13e−08	5.74e−11	8
stocfor2	1.0840e+28	31	2.83e−10	1.26e−10	8
tuff	1.3385e+36	25	2.09e−12	2.87e−13	8
vtpbase	9.4701e+28	28	2.17e−11	1.66e−06	9
wood1p	1.3333e+24	28	1.94e−06	3.45e−14	8
woodw	4.6615e+28	27	3.07e−09	4.97e−14	9

TABLE 3
A priori test of stability (problems 1-p).

Problem	$\|K\|$	$\|L\|$	$\|D\|$	last τ
25fv47	4.32e+02	1.08e+19	1.08e+19	2.51e+16
80bau3b	2.16e+02	3.98e+13	3.96e+13	7.28e+11
adlittle	1.03e+02	1.76e+09	3.03e+10	3.24e+08
afiro	3.36e+00	2.87e+12	2.04e+12	4.04e+12
agg	4.28e+02	3.62e+14	5.54e+16	4.68e+14
agg2	4.29e+02	1.20e+14	6.50e+13	2.14e+12
agg3	4.29e+02	2.08e+14	1.13e+14	3.68e+12
bandm	1.06e+03	1.38e+13	7.63e+13	9.54e+11
beaconfd	1.10e+03	2.72e+13	2.09e+15	5.00e+12
blend	1.05e+02	2.54e+12	2.42e+12	1.09e+11
bnl1	4.18e+02	1.45e+14	9.67e+13	9.55e+11
bnl2	5.07e+02	4.51e+26	4.51e+26	2.63e+24
boeing1	9.92e+02	5.93e+17	1.49e+14	1.03e+12
boeing2	1.03e+03	1.50e+14	1.29e+14	5.81e+11
bore3d	3.13e+03	2.98e+14	3.33e+14	2.51e+11
brandy	9.43e+02	9.69e+16	7.65e+15	3.85e+15
capri	5.61e+02	1.06e+14	1.05e+14	5.63e+11
cycle	3.37e+03	2.16e+18	2.03e+15	2.24e+15
czprob	1.48e+02	1.24e+13	1.20e+13	9.19e+10
d2q06c	7.32e+03	1.88e+19	1.48e+18	5.42e+15
degen2	3.51e+01	4.44e+08	1.29e+08	1.70e+08
degen3	9.37e+01	6.28e+09	3.33e+09	1.25e+09
e226	9.84e+02	1.74e+13	1.33e+13	1.71e+11
etamacro	3.08e+03	9.76e+12	1.91e+13	7.12e+10
fffff800	1.13e+05	8.10e+15	6.53e+15	8.91e+10
finnis	3.95e+01	1.34e+17	1.34e+17	3.39e+15
fit1d	3.25e+03	4.78e+10	7.31e+12	1.67e+10
fit1p	1.42e+05	1.50e+11	5.17e+11	3.24e+07
fit2d	3.22e+03	4.68e+08	8.24e+09	1.24e+08
fit2p	2.73e+05	7.94e+11	7.94e+11	2.90e+06
forplan	3.09e+03	1.83e+17	2.44e+20	2.35e+17
ganges	1.20e+01	6.84e+16	6.29e+16	2.02e+16
gfrdpnc	2.35e+03	7.44e+12	6.01e+12	1.12e+10
greenbea	1.00e+02	3.21e+22	3.61e+22	8.68e+20
greenbeb	1.41e+02	3.01e+14	2.39e+14	8.63e+12
grow15	5.21e+00	5.86e+16	4.01e+16	2.54e+16
grow22	5.21e+00	5.67e+16	5.51e+16	2.60e+16
grow7	5.21e+00	9.28e+15	6.36e+15	3.72e+15
israel	9.22e+03	3.11e+12	3.82e+12	1.86e+09
kb2	6.02e+02	1.48e+13	2.55e+13	1.55e+11
lotfi	4.83e+03	1.53e+22	1.53e+24	3.17e+20
maros	5.10e+04	1.58e+16	1.23e+16	4.56e+13
nesm	9.35e+01	1.21e+12	1.21e+12	5.46e+10
perold	7.85e+04	1.83e+17	1.91e+19	2.12e+15
pilot4	6.85e+04	1.02e+17	4.35e+17	6.18e+14
pilot87	1.10e+03	1.90e+16	7.55e+16	1.01e+14
pilotja	1.56e+06	1.36e+17	2.20e+18	4.55e+13
pilotnov	1.19e+07	2.06e+17	2.84e+18	1.64e+13
pilots	2.66e+02	9.69e+15	1.04e+16	9.73e+13
pilotwe	7.50e+03	2.27e+15	2.54e+17	8.56e+13

TABLE 4
A priori test of stability (problems r-z).

Problem	$\|K\|$	$\|L\|$	$\|D\|$	last τ
recipe	9.15e+02	5.90e+15	5.68e+15	3.15e+13
sc105	5.52e+00	3.66e+13	3.66e+13	3.83e+13
sc205	5.52e+00	1.88e+15	1.88e+15	1.88e+15
sc50a	5.50e+00	1.50e+13	1.29e+13	1.24e+13
sc50b	1.00e+01	6.51e+12	6.51e+12	3.24e+12
scagr25	1.70e+01	1.99e+12	1.90e+12	3.61e+11
scagr7	1.70e+01	1.43e+11	9.15e+10	5.38e+10
scfxm1	8.26e+02	1.86e+16	9.30e+17	2.34e+15
scfxm2	8.24e+02	9.11e+17	4.55e+19	1.15e+17
scfxm3	8.26e+02	1.57e+18	7.87e+19	1.98e+17
scorpion	6.85e+00	6.49e+07	6.34e+07	2.47e+07
scrs8	3.59e+02	8.15e+19	8.08e+19	4.80e+17
scsd1	4.32e+00	2.30e+09	2.30e+09	2.00e+09
scsd6	5.79e+00	2.43e+09	2.09e+09	1.43e+09
scsd8	4.31e+00	1.62e+10	1.04e+10	1.15e+10
sctap1	2.89e+02	7.26e+09	2.58e+09	1.37e+09
sctap2	4.65e+02	4.63e+09	1.46e+10	9.49e+08
sctap3	4.65e+02	5.09e+09	1.76e+10	1.04e+09
seba	1.34e+03	3.71e+13	3.71e+13	2.78e+10
share1b	2.05e+03	7.36e+16	1.12e+17	2.26e+14
share2b	7.89e+02	5.47e+11	3.43e+11	2.68e+11
shell	1.80e+01	6.40e+11	4.88e+11	1.55e+11
ship04l	7.10e+01	6.57e+09	3.25e+08	3.86e+08
ship04s	4.90e+01	8.03e+08	1.42e+07	6.86e+07
ship08l	7.10e+01	2.08e+08	1.53e+08	5.21e+06
ship08s	4.10e+01	7.76e+08	1.01e+08	5.76e+07
ship12l	5.70e+01	1.93e+09	5.94e+08	9.70e+07
ship12s	3.10e+01	3.18e+08	1.44e+08	2.70e+07
sierra	1.00e+05	4.82e+10	4.68e+10	1.43e+06
stair	3.40e+01	1.70e+15	1.70e+15	2.00e+14
standata	1.34e+02	6.19e+13	6.14e+13	5.43e+11
standmps	1.05e+03	8.54e+12	8.48e+12	9.36e+09
stocfor1	1.10e+03	2.22e+12	3.48e+12	1.05e+10
stocfor2	1.20e+03	8.15e+13	1.33e+14	5.46e+11
tuff	1.92e+03	5.09e+17	1.16e+17	1.31e+15
vtpbase	2.91e+02	2.41e+14	2.32e+14	1.16e+12
woodlp	1.91e+04	5.40e+12	6.54e+13	7.57e+11
woodw	6.24e+04	6.61e+12	3.16e+15	3.55e+11

Acknowledgments. The author thanks Tami Carpenter and Michael Saunders, the associate editor, for carefully reading the paper and suggesting several improvements. He also wishes to acknowledge an anonymous referee who suggested the proof given for Theorem 2.1, which is shorter than the author's original proof, as well as other important improvements.

REFERENCES

[1] I. ADLER AND R. D. C. MONTEIRO, *Limiting behavior of the affine scaling continuous trajectories for linear programming*, Math. Programming, 50 (1991), pp. 29–51.

[2] J. R. BUNCH AND L. C. KAUFMAN, *Some stable methods for calculating inertia and solving symmetric linear equations*, Math. Comput., 31 (1977), pp. 163–179.

[3] J. R. BUNCH AND B. N. PARLETT, *Direct methods for solving symmetric indefinite systems of linear equations*, SIAM J. Numer. Anal., 8 (1971), pp. 639–655.

[4] Y. C. CHENG, D. J. HOUCK, J. M.LIU, M. S. MEKETON, L. SLUTSMAN, R. J. VANDERBEI, AND P. WANG, *The AT&T KORBX system*, AT&T Tech. J., 68 (1989), pp. 7–19.

[5] A. L. FORSGREN AND W. MURRAY, *Newton methods for large-scale linear equality-constrained minimization*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 560–587.

[6] R. FOURER AND S. MEHROTRA, *Solving symmetric indefinite systems in an interior point method for linear programming*, Math. Programming, 62 (1993), pp. 15–40.

[7] D. M. GAY, *Electronic mail distribution of linear programming test problems*, Mathematical Programming Society COAL Newslettter, 13 (1985), pp. 10–12.

[8] A. GEORGE AND J. LIU, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[9] P. E. GILL, W. MURRAY, D. B. PONCELEÓN, AND M. A. SAUNDERS, *Preconditioners for indefinite systems arising in optimization*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 292–311.

[10] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, 1989.

[11] N. K. KARMARKAR AND K. G. RAMAKRISHNAN, *Computational results of an interior point algorithm for large scale linear programming*, Math. Programming, 52 (1991), pp. 555–586.

[12] I. J. LUSTIG, R. E. MARSTEN, AND D. F. SHANNO *On implementing Mehrotra's predictor-corrector interior point method for linear programming*, SIAM J. Optim., 2 (1992), pp. 435–449.

[13] ———, *Computational experience with a primal-dual interior point method for linear programming*, Linear Algebra Appl., 152 (1991), pp. 191–222.

[14] R. E. MARSTEN, M. J. SALTZMAN, D. F. SHANNO, G. S. PIERCE, AND J. F. BALLINTIJN, *Implementation of a dual interior point algorithm for linear programming*, ORSA J. Computing, 1 (1989), pp. 287–297.

[15] K. A. MCSHANE, C. L. MONMA, AND D. F. SHANNO, *An implementation of a primal-dual interior point method for linear programming*, ORSA J. Computing, 1 (1989), pp. 70–83.

[16] S. MEHROTRA, *Implementations of affine scaling methods: Approximate solutions of systems of linear equations using preconditioned conjugate gradient methods*, Tech. Report 89-04, Dept. of Indus. Engrg. and Mgmt. Sci., Northwestern University, Evanston, IL, 1989.

[17] ———, *Implementations of affine scaling methods: towards faster implementations with complete Cholesky factor in use*, Tech. Report 89-15, Dept. of Indus. Engrg. and Mgmt. Sci., Northwestern University, Evanston, IL, 1989.

[18] ———, *On the implementation of a (primal-dual) interior point method*, Tech. Report 90-03, Dept. of Indus. Engrg. and Mgmt. Sci., Northwestern University., Evanston, IL, 1990.

[19] D. B. PONCELEÓN, *Barrier methods for large-scale quadratic programming*, Technical Report SOL 91-2, Stanford University, Stanford, CA, October 1991.

[20] K. TURNER, *Computing projections for the Karmarkar algorithm*, Linear Algebra Appl., 152 (1991), pp. 141–154.

[21] R. J. VANDERBEI, *A brief description of ALPO*, OR Letters, 1991, pp. 531–534,

[22] ———, *Loqo users manual*, Tech. Report SOR 92-5, Princeton University, Princeton, NJ, 1992.

[23] ———, *ALPO: Another linear program optimizer*, ORSA J. Computing, 5 (1993), pp. 134–146.

[24] R. J. VANDERBEI AND T. J. CARPENTER, *Symmetric indefinite systems for interior-point methods*, Math. Programming, 58 (1993), pp. 1–32.

SIAM J. OPTIMIZATION
Vol. 5, No. 1, pp. 114–128, February 1995

006

ON THE PRIMAL-DUAL STEEPEST DESCENT ALGORITHM FOR EXTENDED LINEAR-QUADRATIC PROGRAMMING *

CIYOU ZHU†

Abstract. The aim of this paper is two-fold. First, new variants are proposed for the primal-dual steepest descent algorithm as one in the family of primal-dual projected gradient algorithms developed by Zhu and Rockafellar [*SIAM J. Optim.*, 3 (1993), pp. 751–783] for large-scale extended linear-quadratic programming. The variants include a second update scheme for the iterates, where the primal-dual feedback is arranged in a new pattern, as well as alternatives for the "perfect line search" in the original version of the reference. Second, new linear convergence results are proved for all these variants of the algorithm, including the original version as a special case, without the additional assumptions used by Zhu and Rockafellar. For the variants with the second update scheme, a much sharper estimation for the rate of convergence is obtained due to the new primal-dual feedback pattern.

Key words. extended linear-quadratic programming, large-scale optimization, projected gradient, primal-dual feedback

AMS subject classifications. 65K05, 65K10, 90C20

1. Introduction.

The primal-dual steepest descent algorithm (PDSD) is one in the family of primal-dual projected gradient algorithms proposed by Zhu and Rockafellar [1] for large-scale extended linear-quadratic programming, which arises as a flexible modeling scheme in dynamic and stochastic optimization [2]–[10].

Let $L(u,v)$ be the *Lagrangian function* defined as

$$L(u,v) = p\cdot u + \tfrac{1}{2}u\cdot Pu + q\cdot v - \tfrac{1}{2}v\cdot Qv - v\cdot Ru, \tag{1.1}$$

where $p \in \mathbb{R}^n$, $q \in \mathbb{R}^m$, $R \in \mathbb{R}^{m\times n}$ and the matrices $P \in \mathbb{R}^{n\times n}$ and $Q \in \mathbb{R}^{m\times m}$ are symmetric and positive semidefinite. Let U and V be nonempty polyhedral convex sets in $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively. The primal problem of extended linear-quadratic programming is to

$$(\mathcal{P}) \qquad \text{minimize} f(u) \quad \text{over all} \quad u \in U, \quad \text{where} \quad f(u) := \sup_{v\in V} L(u,v).$$

Associated with this primal problem is the dual problem

$$(\mathcal{Q}) \qquad \text{maximize} g(v) \quad \text{over all} \quad v \in V, \quad \text{where} \quad g(v) := \inf_{u\in U} L(u,v).$$

The problems $(\mathcal{P})$ and $(\mathcal{Q})$ are called *fully quadratic* if both the matrices P and Q are actually positive definite. The basic properties of the objective functions f and g, and the duality relationship between $(\mathcal{P})$ and $(\mathcal{Q})$ are included in the following two theorems.

---

*Received by the editors July 20, 1992; accepted for publication (in revised form) October 10, 1993. This work was supported by Eliezer Naddor Postdoctoral Fellowship in Mathematical Sciences at the Department of Mathematical Sciences, The Johns Hopkins University, Baltimore, Maryland 21218-2689.

†Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, Illinois 60439 (czhu@mcs.anl.gov).

THEOREM 1.1 [5] (Properties of the objective functions). *The objective functions f in $(\mathcal{P})$ and g in $(\mathcal{Q})$ are piecewise linear-quadratic: in each case the space can be partitioned in principle into a finite collection of polyhedral cells, relative to which the function has a linear or quadratic formula. Moreover, f is convex while g is concave. In the fully quadratic case of $(\mathcal{P})$ and $(\mathcal{Q})$ f is strongly convex and g is strongly concave, both functions having continuous first derivatives.*

THEOREM 1.2 [5], [2] (Duality and optimality). (a) *If either of the optimal values* $\inf(\mathcal{P})$ *or* $\sup(\mathcal{Q})$ *is finite, then both are finite and equal, in which event optimal solutions $\bar{u}$ and $\bar{v}$ exist for the two problems. In the fully quadratic case, both the optimal values* $\inf(\mathcal{P})$ *and* $\sup(\mathcal{Q})$ *are finite and equal, and the optimal solutions $\bar{u}$ and $\bar{v}$ are unique.*

(b) *A pair $(\bar{u}, \bar{v})$ is a saddlepoint of $L(u, v)$ over $U \times V$ if and only if $\bar{u}$ solves $(\mathcal{P})$ and $\bar{v}$ solves $(\mathcal{Q})$, or equivalently, $f(\bar{u}) = g(\bar{v})$.*

Hence the extended linear-quadratic programming can be cast in the form of finding a saddlepoint $(\bar{u}, \bar{v})$ of the Lagrangian $L(u, v)$ over $U \times V$. With the notations

$$
\begin{aligned}
\rho_{V,Q}(r) &= \sup_{v \in V}\{r \cdot v - \tfrac{1}{2} v \cdot Q v\} \quad \text{for} \quad r \in \mathbb{R}^m, \\
\rho_{U,P}(s) &= \sup_{u \in U}\{s \cdot u - \tfrac{1}{2} u \cdot P u\} \quad \text{for} \quad s \in \mathbb{R}^n,
\end{aligned} \tag{1.2}
$$

the objective functions in $(\mathcal{P})$ and $(\mathcal{Q})$ can be written as

$$
\begin{aligned}
f(u) &= p \cdot u + \tfrac{1}{2} u \cdot P u + \rho_{V,Q}(q - Ru), \\
g(v) &= q \cdot v - \tfrac{1}{2} v \cdot Q v - \rho_{U,P}(R^T v - p).
\end{aligned} \tag{1.3}
$$

According to Rockafellar [5], the ρ terms here can represent "sharp" constraints as well as penalty terms of piecewise linear-quadratic nature. These terms provide rich possibilities in mathematical modeling.

The extended linear-quadratic programming problems in multistage or stochastic optimization are usually of very high dimension on one hand, while on the other hand, possess special structures, such as the so-called *Lagrangian decomposability* [7] (cf. also §2). A foundation for numerical schemes regarding these problems has been laid out in Rockafellar and Wets [2] and Rockafellar [7] and elaborated on for problems in multistage format in Rockafellar [8]. The PDSD algorithm [1] is designed specifically to take advantage of these results and to cope with the high dimensionality. The algorithm works with local structure in the primal and dual problems simultaneously. Computations for problems in multistage format could be handled through the system dynamics in such a way that no huge R matrix should be formed explicitly. A novel kind of primal-dual feedback is introduced between the primal part and the dual part of the algorithm to trigger advantageous *interactive restarts* [1]. The algorithm is capable of solving extended linear-quadratic programming problems of both the primal and dual dimensions up to 100,000 effectively on a DECstation 3100 [1].

The convergence of the PDSD algorithm was proved in [1] as a special case of the results on the family of the primal-dual projected gradient algorithms. However, the estimation on the rate of convergence there is of asymptotic nature and seems far behind its practical performance. Moreover, the results were obtained under some additional *critical face conditions* [1]. The primal-dual feedback, which plays an important role in the practical performance of the algorithm, has no effect in the derivation of these theoretical estimations.

In this paper, we propose new variants for the algorithm and prove improved results on the rate of convergence. First, in §2, we propose a second update scheme for the iterates, where the primal-dual feedback is arranged in a new pattern. We also give "fixed" or "adaptive" step length rules as alternatives of the "perfect line search" used in the original version. All these variants, including the original version of the algorithm, are put in a unified framework. Then, in §3, we prove new linear convergence results for all these variants without the critical face conditions. The results are of a global nature, and the estimates on the rates of convergence are much improved compared with the ones in [1]. For the variants with the new update scheme, sharper estimates for the rates are obtainable due to the new primal-dual feedback pattern. Finally, in §4, we discuss numerical test results and other possible update schemes.

2. The primal-dual steepest descent algorithm. The family of primal-dual projected gradient algorithms in [1], as well as the finite-envelope algorithm developed earlier by Rockafellar and Wets [2]–[7] are all designed for solving large-scale extended linear-quadratic programming problems arising in multistage or stochastic optimization, where the problems exhibit the Lagrangian decomposability (or double decomposability) [7]. The latter means that for any fixed $u \in U$ it is relatively easy to maximize $L(u, v)$ over $v \in V$, and likewise, for any fixed $v \in V$ it is relatively easy to minimize $L(u, v)$ over $u \in U$. This is the case, for example, when the matrices P and Q are block diagonal, and the sets U and V are corresponding Cartesian products of polyhedra of low dimensions. These subproblems of maximization and minimization calculate not only the objective values $f(u)$ and $g(v)$ but also, in the fully quadratic case when L is strongly convex-concave, the uniquely determined vectors

$$F(u) = \operatorname*{argmax}_{v \in V} L(u, v) \quad \text{and} \quad G(v) = \operatorname*{argmin}_{u \in U} L(u, v). \tag{2.1}$$

The mappings F and G play a central role in the PDSD algorithm.

From [7] and [1] we cite several fundamental properties that are useful later in this paper. In notation, we write

$$\|w\|_M = [w \cdot Mw]^{\frac{1}{2}}$$

for the norm corresponding to a symmetric positive definite matrix M. It reduces to the ordinary Euclidean norm when M is the identity matrix. In this latter case, the subscript is dropped. We use the related operator norm for matrices and use $[w_1, w_2]$ to denote the line segment between two points w_1 and w_2. We impose *the blanket assumption that the problem is fully quadratic* for the rest of the paper and refer consistently to

$$\bar{u} = \text{the unique optimal solution to } (\mathcal{P}),$$
$$\bar{v} = \text{the unique optimal solution to } (\mathcal{Q}).$$

In the case when the problem under consideration is not fully quadratic, an outer loop of *proximal point iteration* can be used to create fully quadratic inner loop problems. See [2], [7], and [11] for related discussions.

Let $P^{1/2}$ and $Q^{1/2}$ be the "square roots" of P and Q, respectively, defined via orthogonal factorization. Define

$$\gamma := \gamma(P, Q, R) := \|Q^{-\frac{1}{2}} R P^{-\frac{1}{2}}\|. \tag{2.2}$$

PROPOSITION 2.1 [7] (Optimality estimates). *Suppose u and v are elements of U and V satisfying $f(u) - g(v) \leq \varepsilon$ for a certain $\varepsilon \geq 0$. Then u and v are ε-optimal in the sense that $|f(u) - f(\bar{u})| \leq \varepsilon$ and $|g(v) - g(\bar{v})| \leq \varepsilon$. Moreover,*

$$\|u - \bar{u}\|_P^2 + \|v - \bar{v}\|_Q^2 \leq 2\varepsilon.$$

PROPOSITION 2.2 [7] (Regularity properties). *The functions f and g are continuously differentiable everywhere with*

$$\nabla f(u) = \nabla_u L(u, F(u)) \quad \text{and} \quad \nabla g(v) = \nabla_v L(G(v), v),$$

while the mappings F and G defined by (2.1) *are Lipschitz continuous with*

$$\|F(u') - F(u)\|_Q \leq \gamma \|u' - u\|_P \quad \text{for all} \quad u \quad \text{and} \quad u',$$
$$\|G(v') - G(v)\|_P \leq \gamma \|v' - v\|_Q \quad \text{for all} \quad v \quad \text{and} \quad v'.$$

PROPOSITION 2.3 [7], [1] (Modified gradient projection). *For arbitrary $u \in U$ and $v \in V$,*

$$G(F(u)) - u = P\text{-projection of} \quad -\nabla_P f(u) \quad \text{on} \quad U - u,$$
$$F(G(v)) - v = Q\text{-projection of} \quad \nabla_Q g(v) \quad \text{on} \quad V - v,$$

where $\nabla_P f(u) = P^{-1}\nabla f(u)$ symbolizes the gradient of f relative to the P-norm, while $\nabla_Q g(v) = Q^{-1}\nabla g(v)$ symbolizes the gradient of g relative to the Q-norm. Moreover, the vector $G(F(u)) - u$ is a feasible descent direction of f at u unless $u = \bar{u}$. Similarly, the vector $F(G(v)) - v$ is a feasible ascent direction of g at v unless $v = \bar{v}$.

The PDSD algorithm first searches on line segments $[u, G(F(u))]$ and $[v, F(G(v))]$ in primal and dual variables, respectively, to get some *intermediate points* as candidates for the next iterates. (Proposition 2.3 above suggests the name "projected gradient.") Then a novel kind of primal-dual feedback is incorporated in the updating. In the case of "forward feedback," the next iterates are chosen between the intermediate points and their images under the mappings F and G, while in the case of "backward feedback," the next iterates are chosen between the intermediate points and the images of the current iterates under the mappings F and G. This kind of interactive effects ties the primal and dual part of the operation closely and has proved to be important to the performance of the algorithm.

In the following, we introduce new variants of the PDSD algorithm. The second update scheme for the iterates corresponds to the backward feedback for which a sharper bound for the rate of convergence will be obtained. We also give alternatives for the "perfect line search" used in the original version. We put all these variants, including two different update schemes and three step length rules, in a unified framework. We refer to the algorithm with, say, update scheme (2) and step length rule (iii), as PDSD–2(iii). Under this convention, the PDSD algorithm in [1] should be referred to as PDSD–1(i).

PRIMAL-DUAL STEEPEST DESCENT ALGORITHM.

Step 0 (initialization). Set $\nu := 0$ (iteration counter). Specify starting points $u^0 \in U$ and $v^0 \in V$. Choose one of the step length rules in Step 2. (If rule (iii) is chosen, then also choose some constant $\delta \in (0, 1)$ and let $\alpha_{-1} = \beta_{-1} = 1$.) Choose one of the update schemes in Step 3. Construct primal and dual sequences $\{u^\nu\} \subset U$ and $\{v^\nu\} \subset V$ as follows.

Step 1 (optimality test). If

$$\min\{ f(u^\nu), f\big(G(v^\nu)\big) \} - \max\{ g(v^\nu), g\big(F(u^\nu)\big) \} = 0,$$

then terminate with

$$\begin{aligned} \bar{u} &= \operatorname{argmin}\{ f(u) \,|\, u = u^\nu, \quad \text{or} \quad u = G(v^\nu) \}, \\ \bar{v} &= \operatorname{argmax}\{ g(v) \,|\, v = v^\nu, \quad \text{or} \quad v = F(u^\nu) \} \end{aligned}$$

being optimal solutions to $(\mathcal{P})$ and $(\mathcal{Q})$.

Step 2 (line search). Use one of the following step length rules chosen at initialization to determine α_ν and β_ν for generating intermediate points

$$\begin{aligned} \hat{u}^{\nu+1} &:= (1-\alpha_\nu)u^\nu + \alpha_\nu G(F(u^\nu)), \\ \hat{v}^{\nu+1} &:= (1-\beta_\nu)v^\nu + \beta_\nu F(G(v^\nu)) \end{aligned}$$

in primal and dual variables, respectively.

(i) Perfect line search:

$$\begin{aligned} \alpha_\nu &:= \operatorname*{argmin}_{\alpha\in[0,1]} f\big((1-\alpha)u^\nu + \alpha G(F(u^\nu))\big), \\ \beta_\nu &:= \operatorname*{argmax}_{\beta\in[0,1]} g\big((1-\beta)v^\nu + \beta F(G(v^\nu))\big). \end{aligned}$$

(ii) Fixed step lengths:

$$\alpha_\nu := \min\left\{1, \frac{1}{2\gamma^2}\right\} \quad \text{and} \quad \beta_\nu := \min\left\{1, \frac{1}{2\gamma^2}\right\}.$$

(We adopt the convention $0^{-1} = +\infty$ in this paper.)

(iii) Adaptive step lengths:

$$\begin{aligned} \alpha_\nu := \max\big\{ \alpha_{\nu-1}\delta^j \,\big|\, & f\big((1-\alpha_{\nu-1}\delta^j)u^\nu + \alpha_{\nu-1}\delta^j G(F(u^\nu))\big) - f(u^\nu) \\ & \le \big(f(u^\nu) - g(F(u^\nu))\big)(-\tfrac{1}{2}\alpha_{\nu-1}\delta^j),\ j \in \{0,1,2,\ldots\} \big\}, \\ \beta_\nu := \max\big\{ \beta_{\nu-1}\delta^j \,\big|\, & g(v^\nu) - g\big((1-\beta_{\nu-1}\delta^j)v^\nu + \beta_{\nu-1}\delta^j F(G(v^\nu))\big) \\ & \le \big(f(G(v^\nu)) - g(v^\nu)\big)(-\tfrac{1}{2}\beta_{\nu-1}\delta^j),\ j \in \{0,1,2,\ldots\} \big\}. \end{aligned}$$

Step 3 (update the iterates). Use one of the following rules chosen at initialization to determine the next iterates.

(1) Update with forward feedback:

$$\begin{aligned} u^{\nu+1} &:= \operatorname{argmin}\{ f(u) \,|\, u = \hat{u}^{\nu+1} \quad \text{or} \quad u = G(\hat{v}^{\nu+1}) \}, \\ v^{\nu+1} &:= \operatorname{argmax}\{ g(v) \,|\, v = \hat{v}^{\nu+1} \quad \text{or} \quad v = F(\hat{u}^{\nu+1}) \}. \end{aligned}$$

(If both the arguments give the same objective value, use the first one in updating for decisiveness. The same rule applies also to the next set of formulas.)

(2) Update with backward feedback:

$$\begin{aligned} u^{\nu+1} &:= \operatorname{argmin}\{ f(u) \,|\, u = \hat{u}^{\nu+1} \quad \text{or} \quad u = G(v^\nu) \}, \\ v^{\nu+1} &:= \operatorname{argmax}\{ g(v) \,|\, v = \hat{v}^{\nu+1} \quad \text{or} \quad v = F(u^\nu) \}. \end{aligned}$$

Then return to Step 1 with the counter ν increased by 1.

Observe that the primal-dual feedback also takes place in the optimality test. It follows from Proposition 2.2 that $F(u^\nu) \to \bar{v}$ and $G(v^\nu) \to \bar{u}$ as $u^\nu \to \bar{u}$ and $v^\nu \to \bar{v}$. With the optimality test in Step 1, the algorithm will terminate if either $u^\nu = \bar{u}$ or $v^\nu = \bar{v}$ by Theorem 1.2.

In Step 2, there are three step length rules to choose from. By Theorem 1.1 and Proposition 2.2, the objective functions in the line searches are piecewise quadratic and continuously differentiable. In the typical decomposable case when P and Q are diagonal, and U and V are "boxes" representing upper and lower bounds, one can further get the explicit expressions for the derivatives of these functions. By taking advantage of all these properties, even the perfect line search will not be prohibitively difficult. In our numerical experimentations, the perfect line search takes approximately two-thirds of the time in each iteration.

An interesting result of Theorem 3.1 in §3 is that the same estimated rate of convergence as for the perfect line search (i) can be reached by certain fixed step lengths in rule (ii). However the parameter γ of the problem, which determines the length of steps in (ii), is usually unavailable. Therefore we provide a third rule with adaptive step lengths that resembles the Armijo stepsize rule for unconstrained minimization. However, here we use certain duality gap, instead of the slope of the line search function, in determining the step lengths. Theorem 3.2 in §3 shows that the adaptive step length is well defined, that the step lengths will be fixed after a finite number of adaptations, and that an estimated rate of convergence very close to the one with perfect line search is obtainable.

Update scheme (1) in Step 3 can also be written as

$$u^{\nu+1} := \begin{cases} \hat{u}^{\nu+1}, & \text{if } f(\hat{u}^{\nu+1}) \le f(G(\hat{v}^{\nu+1})), \\ G(\hat{v}^{\nu+1}) & \text{otherwise,} \end{cases} \tag{2.3}$$

$$v^{\nu+1} := \begin{cases} \hat{v}^{\nu+1}, & \text{if } g(\hat{v}^{\nu+1}) \ge g(F(\hat{u}^{\nu+1})), \\ F(\hat{u}^{\nu+1}) & \text{otherwise.} \end{cases} \tag{2.4}$$

We say that there is an *interactive restart in the primal variable* if $u^{\nu+1} = G(\hat{v}^{\nu+1})$, in which case, the primal iterate is updated by using the dual information. Similarly, we say that there is an *interactive restart in the dual variable* if $v^{\nu+1} = F(\hat{u}^{\nu+1})$, in which case, the dual iterate is updated by using the primal information. Update scheme (2) can be written in the same manner as

$$u^{\nu+1} := \begin{cases} \hat{u}^{\nu+1}, & \text{if } f(\hat{u}^{\nu+1}) \le f(G(v^\nu)), \\ G(v^\nu) & \text{otherwise.} \end{cases} \tag{2.5}$$

$$v^{\nu+1} := \begin{cases} \hat{v}^{\nu+1}, & \text{if } g(\hat{v}^{\nu+1}) \ge g(F(u^\nu)), \\ F(u^\nu) & \text{otherwise,} \end{cases} \tag{2.6}$$

with the interactive restarts defined accordingly. Although the practical performance of the algorithm with these two different update schemes is very close in our tests, a sharper bound for the rate of convergence of the algorithm with scheme (2) will be obtained in the next section.

To conclude §2, we give a lemma that is used later in deriving convergence results. The proof of the lemma follows closely the idea in the proofs of Rockafellar and Wets [2, Prop. 3 and Thm. 5].

LEMMA 2.1. *For any* $u \in U$,

$$f\big((1-\alpha)u + \alpha G(F(u))\big) - f(u) \le \big(f(u) - g(F(u))\big)(-\alpha + \gamma^2\alpha^2) \tag{2.7}$$

for all $\alpha \in [0,1]$. *Similarly, for any* $v \in V$,

$$g(v) - g\big((1-\beta)v + \beta F(G(v))\big) \le \big(f(G(v)) - g(v)\big)(-\beta + \gamma^2\beta^2) \tag{2.8}$$

for all $\beta \in [0,1]$.

Proof. For any $u_0 \in U$, denote $v_1 := F(u_0)$ and $u_2 := G(v_1)$. Then the Lagrangian $L(u,v)$ can be written in the expanded form at (u, v_1) as

$$L(u,v) = L(u,v_1) + \nabla_v L(u,v_1)\cdot(v - v_1) - \tfrac{1}{2}(v - v_1)\cdot Q(v - v_1),$$

where the term $\nabla_v L(u,v_1)\cdot(v - v_1)$ can be further written as

$$\nabla_v L(u,v_1)\cdot(v - v_1) = \nabla_v L(u_0,v_1)\cdot(v - v_1) - (v - v_1)\cdot R(u - u_0).$$

Note that $v_1 = F(u_0)$ means v_1 is the argmax of $L(u_0, v)$ on V, which in turn implies $\nabla_v L(u_0,v_1)\cdot(v - v_1) \le 0$ for all $v \in V$. Hence

$$L(u,v) \le L(u,v_1) - (v - v_1)\cdot R(u - u_0) - \tfrac{1}{2}(v - v_1)\cdot Q(v - v_1). \tag{2.9}$$

Now for any $u \in [u_0, u_2]$ and $v = F(u)$, it follows from (2.9) that

$$\begin{aligned} &L\big(u, F(u)\big) - L(u,v_1) \\ &\le -\big(F(u) - v_1\big)\cdot R(u - u_0) - \tfrac{1}{2}\big(F(u) - v_1\big)\cdot Q\big(F(u) - v_1\big) \\ &\le \max_{w \in \mathbb{R}^m} \{w\cdot R(u - u_0) - \tfrac{1}{2} w\cdot Q w\} \\ &= \tfrac{1}{2}(u - u_0)\cdot\big(R^T Q^{-1} R\big)(u - u_0) \\ &= \tfrac{1}{2}\|(Q^{-\frac{1}{2}} R P^{-\frac{1}{2}})P^{\frac{1}{2}}(u - u_0)\|^2 \\ &\le \tfrac{1}{2}\gamma^2 \|u - u_0\|_P^2. \end{aligned} \tag{2.10}$$

However, $L\big(u, F(u)\big) = f(u)$ and

$$\begin{aligned} L\big((1-\alpha)u_0 + \alpha u_2, v_1\big) &\le (1-\alpha)L(u_0,v_1) + \alpha L(u_2,v_1) \\ &= (1-\alpha)f(u_0) + \alpha g(v_1) \end{aligned}$$

for $0 \le \alpha \le 1$. Thus, by taking $u = (1-\alpha)u_0 + \alpha u_2$ in (2.10), we get

$$f\big((1-\alpha)u_0 + \alpha u_2\big) - f(u_0) + \alpha\big(f(u_0) - g(v_1)\big) \le \tfrac{1}{2}\alpha^2\gamma^2\|u_2 - u_0\|_P^2. \tag{2.11}$$

On the other hand,

$$\begin{aligned} f(u_0) - g(v_1) &= L(u_0,v_1) - L(u_2,v_1) \\ &= \nabla_u L(u_2,v_1)\cdot(u_0 - u_2) + \tfrac{1}{2}(u_0 - u_2)\cdot P(u_0 - u_2) \end{aligned}$$

by the definition of v_1 and u_2. Observe that $\nabla_u L(u_2,v_1)\cdot(u_0 - u_2) \ge 0$ since u_2 is the argmin of $L(u,v_1)$ on U. Therefore

$$f(u_0) - g(v_1) \ge \tfrac{1}{2}(u_0 - u_2)\cdot P(u_0 - u_2) = \tfrac{1}{2}\|u_2 - u_0\|_P^2. \tag{2.12}$$

Combining (2.11) and (2.12), we get

$$f\big(u_0+\alpha(u_2-u_0)\big)-f(u_0)\le\big(f(u_0)-g(v_1)\big)(-\alpha+\gamma^2\alpha^2)$$

for $0\le\alpha\le 1$, which yields (2.7). Inequality (2.8) can be proved similarly. □

3. Global linear convergence of the PDSD algorithm. In this section, we prove linear convergence results for all the six variants of the PDSD algorithm formulated in §2. We first give results for the algorithms with (i) perfect line search and (ii) fixed step lengths. Define the function $\theta:[0,+\infty)\to(0,1)$ as

$$\theta(s)=\begin{cases}1-s & \text{if } s<\frac{1}{2},\\ \frac{1}{4s} & \text{if } s\ge\frac{1}{2}.\end{cases}\tag{3.1}$$

THEOREM 3.1 (Convergence of PDSD with step length rules (i) and (ii)).

(a) *The sequences* $\{f(u^\nu)\}$ *and* $\{g(v^\nu)\}$ *generated by* PDSD–1(i) *or* PDSD–1(ii) *converge linearly to the common optimal value* $f(\bar u)=g(\bar v)$ *in the sense that*

$$f(u^{\nu+1})-f(\bar u)\le\big(1-\theta(\gamma^2)\big)\big(f(u^\nu)-f(\bar u)\big),\tag{3.2}$$

$$g(\bar v)-g(v^{\nu+1})\le\big(1-\theta(\gamma^2)\big)\big(g(\bar v)-g(v^\nu)\big).\tag{3.3}$$

Moreover,

$$\|u^{\nu+1}-\bar u\|_P^2+\|v^{\nu+1}-\bar v\|_Q^2\le 2\big(1-\theta(\gamma^2)\big)^{\nu+1}\big(f(u^0)-g(u^0)\big).\tag{3.4}$$

(b) *The sequences* $\{f(u^\nu)\}$ *and* $\{g(v^\nu)\}$ *generated by* PDSD–2(i) *or* PDSD–2(ii) *converge linearly to the common optimal value* $f(\bar u)=g(\bar v)$ *in the sense that*

$$f(u^{\nu+1})-g(v^{\nu+1})\le\frac{1-\theta(\gamma^2)}{1+\theta(\gamma^2)}\big(f(u^\nu)-g(v^\nu)\big).\tag{3.5}$$

Moreover,

$$\|u^{\nu+1}-\bar u\|_P^2+\|v^{\nu+1}-\bar v\|_Q^2\le 2\Big(\frac{1-\theta(\gamma^2)}{1+\theta(\gamma^2)}\Big)^{\nu+1}\big(f(u^0)-g(u^0)\big).\tag{3.6}$$

Proof. It follows from (2.7) that

$$f\big((1-\alpha)u^\nu+\alpha G(F(u^\nu))\big)-f(u^\nu)\le\big(f(u^\nu)-g(F(u^\nu))\big)(-\alpha+\gamma^2\alpha^2)\tag{3.7}$$

for all $\alpha\in[0,1]$. But $\min\{-\alpha+\gamma^2\alpha^2\mid 0\le\alpha\le 1\}=-\theta(\gamma^2)$ with

$$\operatorname{argmin}\{-\alpha+\gamma^2\alpha^2\mid 0\le\alpha\le 1\}=\min\Big\{1,\frac{1}{2\gamma^2}\Big\}.$$

Hence for the fixed step length $\alpha_\nu=\min\{1,\frac{1}{2\gamma^2}\}$ in rule (ii),

$$f\big((1-\alpha_\nu)u^\nu+\alpha_\nu G(F(u^\nu))\big)-f(u^\nu)\le\big(f(u^\nu)-g(F(u^\nu))\big)\big(-\theta(\gamma^2)\big).\tag{3.8}$$

Obviously (3.8) is also true for the step length $\alpha_\nu=\operatorname{argmin}_{\alpha\in[0,1]}f\big((1-\alpha)u^\nu+\alpha G(F(u^\nu))\big)$ in rule (i), since the perfect line search should not make the first term

of (3.8) any larger. According to the update scheme in Step 3, we have $f(u^{\nu+1}) \leq f((1-\alpha_\nu)u^\nu + \alpha_\nu G(F(u^\nu)))$. Therefore

$$f(u^{\nu+1}) - f(u^\nu) \leq \big(f(u^\nu) - g(F(u^\nu))\big)\big(-\theta(\gamma^2)\big). \tag{3.9}$$

Similarly it follows from (2.8) that

$$g(v^\nu) - g\big((1-\beta)v^\nu + \beta F(G(v^\nu))\big) \leq \big(f(G(v^\nu)) - g(v^\nu)\big)\big(-\beta + \gamma^2\beta^2\big) \tag{3.10}$$

for all $\beta \in [0,1]$, which yields

$$g(v^\nu) - g(v^{\nu+1}) \leq \big(f(G(v^\nu)) - g(v^\nu)\big)\big(-\theta(\gamma^2)\big). \tag{3.11}$$

Combining (3.9) and (3.11), we get

$$f(u^\nu) - g(v^\nu) - f(u^{\nu+1}) + g(v^{\nu+1}) \geq \theta(\gamma^2)\big(f(u^\nu) - g(v^\nu) - g(F(u^\nu)) + f(G(v^\nu))\big). \tag{3.12}$$

With the νth *duality gap* ε_ν and the νth *auxiliary duality gap* $\tilde{\varepsilon}_\nu$ defined as

$$\varepsilon_\nu := f(u^\nu) - g(v^\nu) \quad \text{and} \quad \tilde{\varepsilon}_\nu := f(G(v^\nu)) - g(F(u^\nu)), \tag{3.13}$$

respectively, (3.12) can be written in the form

$$\varepsilon_\nu - \varepsilon_{\nu+1} \geq \theta(\gamma^2)(\varepsilon_\nu + \tilde{\varepsilon}_\nu)$$

or, equivalently,

$$\varepsilon_{\nu+1} \leq \big(1 - \theta(\gamma^2)\big)\varepsilon_\nu - \theta(\gamma^2)\tilde{\varepsilon}_\nu. \tag{3.14}$$

If update scheme (2) is used in Step 3 of the algorithm, then $f(u^{\nu+1}) \leq f(G(v^\nu))$ and $g(v^{\nu+1}) \geq g(F(u^\nu))$. Hence $\varepsilon_{\nu+1} \leq \tilde{\varepsilon}_\nu$. Therefore (3.14) implies

$$\varepsilon_{\nu+1} \leq \big(1 - \theta(\gamma^2)\big)\varepsilon_\nu - \theta(\gamma^2)\varepsilon_{\nu+1},$$

from which (3.5) follows. Using (3.5) for $\nu = 0, 1, \ldots$, we get

$$f(u^{\nu+1}) - g(v^{\nu+1}) \leq \Big(\frac{1-\theta(\gamma^2)}{1+\theta(\gamma^2)}\Big)^{\nu+1}\big(f(u^0) - g(v^0)\big),$$

which yields (3.6) by Proposition 2.1.

If update scheme (1) is used in Step 3 of the algorithm, then the relation $\varepsilon_{\nu+1} \leq \tilde{\varepsilon}_\nu$ is not necessarily true. However, by Theorem 1.2,

$$f(u) \geq f(\bar{u}) = g(\bar{v}) \geq g(v) \quad \text{for all} \quad u \in U,\ v \in V.$$

Hence it follows from (3.9) and (3.11) that

$$f(u^{\nu+1}) - f(u^\nu) \leq \big(f(u^\nu) - f(\bar{u})\big)\big(-\theta(\gamma^2)\big),$$
$$g(v^\nu) - g(v^{\nu+1}) \leq \big(g(\bar{v}) - g(v^\nu)\big)\big(-\theta(\gamma^2)\big).$$

These two inequalities yield (3.2) and (3.3), respectively. Moreover, observe that $\tilde{\varepsilon}_\nu \geq 0$. Hence by (3.14), we have

$$f(u^{\nu+1}) - g(v^{\nu+1}) \leq \big(1 - \theta(\gamma^2)\big)\big(f(u^\nu) - g(v^\nu)\big). \tag{3.15}$$

Using (3.15) for $\nu = 0, 1, \ldots$, we get

$$f(u^{\nu+1}) - g(v^{\nu+1}) \leq \big(1 - \theta(\gamma^2)\big)^{\nu+1}\big(f(u^0) - g(v^0)\big),$$

which yields (3.4) by Proposition 2.1. □

Next we give convergence results for the algorithm with adaptive step lengths (iii). We must show, in the first place, that these step lengths are well defined. Let the function $\tilde{\theta} : [0, +\infty) \to (0, 1)$ be defined as

$$\tilde{\theta}(s) = \min\left\{\frac{1}{2}, \frac{1}{4s}\right\}. \tag{3.16}$$

Obviously $\theta(s) \geq \tilde{\theta}(s)$ for all $s \in [0, +\infty)$, and the equality holds when $s \geq \frac{1}{2}$.

THEOREM 3.2. (Convergence of PDSD with step length rule (iii)).

(a) *For any choice of* $\delta \in (0, 1)$, *the step lengths* α_ν *and* β_ν *in the* PDSD *algorithm with rule* (iii) *are well defined. Both* α_ν *and* β_ν *are nonincreasing as* ν *increases, and*

$$\alpha_\nu > \delta \min\left\{1, \frac{1}{2\gamma^2}\right\} \quad \text{and} \quad \beta_\nu > \delta \min\left\{1, \frac{1}{2\gamma^2}\right\} \tag{3.17}$$

for all ν. *Moreover, both* α_ν *and* β_ν *will be fixed after a finite number of iterations.*

(b) *The sequences* $\{f(u^\nu)\}$ *and* $\{g(v^\nu)\}$ *generated by* PDSD–1(iii) *converge linearly to the common optimal value* $f(\bar{u}) = g(\bar{v})$ *in the sense that*

$$f(u^{\nu+1}) - f(\bar{u}) \leq \big(1 - \delta\tilde{\theta}(\gamma^2)\big)\big(f(u^\nu) - f(\bar{u})\big), \tag{3.18}$$

$$g(\bar{v}) - g(v^{\nu+1}) \leq \big(1 - \delta\tilde{\theta}(\gamma^2)\big)\big(g(\bar{v}) - g(v^\nu)\big). \tag{3.19}$$

Moreover,

$$\|u^{\nu+1} - \bar{u}\|_P^2 + \|v^{\nu+1} - \bar{v}\|_Q^2 \leq 2\big(1 - \delta\tilde{\theta}(\gamma^2)\big)^{\nu+1}\big(f(u^0) - g(u^0)\big). \tag{3.20}$$

(c) *The sequences* $\{f(u^\nu)\}$ *and* $\{g(v^\nu)\}$ *generated by* PDSD–2(iii) *converge linearly to the common optimal value* $f(\bar{u}) = g(\bar{v})$ *in the sense that*

$$f(u^{\nu+1}) - g(v^{\nu+1}) \leq \frac{1 - \delta\tilde{\theta}(\gamma^2)}{1 + \delta\tilde{\theta}(\gamma^2)}\big(f(u^\nu) - g(v^\nu)\big), \tag{3.21}$$

Moreover,

$$\|u^{\nu+1} - \bar{u}\|_P^2 + \|v^{\nu+1} - \bar{v}\|_Q^2 \leq 2\Big(\frac{1 - \delta\tilde{\theta}(\gamma^2)}{1 + \delta\tilde{\theta}(\gamma^2)}\Big)^{\nu+1}\big(f(u^0) - g(u^0)\big). \tag{3.22}$$

Proof. First, we claim that for all nonnegative $\alpha \leq \min\{1, \frac{1}{2\gamma^2}\}$,

$$f\big((1-\alpha)u^\nu + \alpha G(F(u^\nu))\big) - f(u^\nu) \leq \big(f(u^\nu) - g(F(u^\nu))\big)\big(\frac{-\alpha}{2}\big). \tag{3.23}$$

This follows directly from (2.7) and the fact that

$$-\alpha+\gamma^2\alpha^2 \le \frac{-\alpha}{2} \quad \text{for all } 0 \le \alpha \le \min\left\{1, \frac{1}{2\gamma^2}\right\}.$$

Hence the step length $\alpha_\nu = \alpha_{\nu-1}\delta^j$ in rule (iii), where j is the *first* element in the ordered nonnegative integer set $\{0, 1, 2, \ldots\}$ satisfying

(3.24)
$$f\big((1-\alpha_{\nu-1}\delta^j)u^\nu + \alpha_{\nu-1}\delta^j G(F(u^\nu))\big) - f(u^\nu) \le \big(f(u^\nu) - g(F(u^\nu))\big)\big(-\tfrac{1}{2}\alpha_{\nu-1}\delta^j\big),$$

is well defined. Obviously $\{\alpha_\nu\}$ is nonincreasing.

According to the claim and the step rule, we have either $\alpha_\nu = \alpha_{\nu-1}$ or $\alpha_{\nu-1}\delta^{j-1} > \min\{1, \frac{1}{2\gamma^2}\}$ with $j \ge 1$, because otherwise $\alpha_{\nu-1}\delta^{j-1}$ instead of $\alpha_{\nu-1}\delta^j$ will be taken as the step length α_ν. Suppose $\alpha_{\nu-1} > \delta\min\{1, \frac{1}{2\gamma^2}\}$. Then in either case,

$$\alpha_\nu = \alpha_{\nu-1}\delta^j > \delta\min\left\{1, \frac{1}{2\gamma^2}\right\}. \tag{3.25}$$

Note that $\alpha_{-1} = 1 > \delta$. This proves the first inequality in (3.17) by induction. The second inequality in (3.17) regarding β_ν can be proved similarly, and the last conclusion in part (a) is now obvious.

Combining (3.24) and (3.25), we have

$$f\big((1-\alpha_\nu)u^\nu + \alpha_\nu G(F(u^\nu))\big) - f(u^\nu) \le \big(f(u^\nu) - g(F(u^\nu))\big)\left(\frac{-\delta}{2}\min\left\{1, \frac{1}{2\gamma^2}\right\}\right).$$

Therefore, by observing $f(u^{\nu+1}) \le f\big((1-\alpha_\nu)u^\nu + \alpha_\nu G(F(u^\nu))\big)$ in the updating, we get

$$f(u^{\nu+1}) - f(u^\nu) \le \big(f(u^\nu) - g(F(u^\nu))\big)\big(-\delta\tilde{\theta}(\gamma^2)\big). \tag{3.26}$$

Similarly, we have

$$g(v^\nu) - g(v^{\nu+1}) \le \big(f(G(v^\nu)) - g(v^\nu)\big)\big(-\delta\tilde{\theta}(\gamma^2)\big). \tag{3.27}$$

Now (3.26) and (3.27) lead to the conclusions in (b) and (c) in the same manner as (3.9) and (3.11) lead to the conclusions in Theorem 3.1. □

Theorems 3.1 and 3.2 provide global linear convergence results for all the variants of the PDSD algorithm formulated in §2 without any additional assumptions. The parameter $\gamma = \|Q^{-1/2}RP^{-1/2}\|$ of the problem plays an important role in the estimations regarding the rates of convergence of the algorithm. It also characterizes the Lipschitzian constant for the mappings F and G in Proposition 2.2. In fact, γ can be viewed as a normalized measure of the "coupling" between the primal and dual variables of the problem. In the extremal case when $\gamma = 0$ (which implies $R = 0$), we have $F(u) = \bar{v}$ for all u and $G(v) = \bar{u}$ for all v. Hence the algorithm will terminate in one iteration. On the other hand, *a large γ implies a difficult problem* for the algorithm.

It follows from Theorem 3.1 that for problems with large γ, the duality gap $\varepsilon_\nu = f(u^\nu) - g(v^\nu)$ of the iterates generated by PDSD–1(i) or PDSD–1(ii) decreases at least with the ratio

$$1 - \theta(\gamma^2) = 1 - \frac{1}{4\gamma^2}, \tag{3.28}$$

while the one generated by PDSD–2(i) or PDSD–2(ii) decreases at least with the ratio

$$1 - \frac{1-\theta(\gamma^2)}{1+\theta(\gamma^2)} \sim 1 - \frac{1}{2\gamma^2}. \tag{3.29}$$

These are much improved estimates compared with the earlier results in [1, Thm. 4.2] with an asymptotic ratio

$$1 - \frac{1}{4(\gamma^2+1)^4 + 5(\gamma^2+1)^2 + 2(\gamma^2+1)} \sim 1 - \frac{1}{4\gamma^8}$$

under the critical face conditions. However if the iterates eventually reach the corresponding critical faces, [1, Thm. 4.2] still gives a better asymptotic ratio

$$\Big(1 - \frac{1}{0.5(\gamma^2+1)+0.5}\Big)^2 \sim 1 - \frac{1}{0.25\gamma^2} \quad (\text{for large}\gamma\ )$$

under the perfect line search. This is consistent with the observation that the algorithm with perfect line search often gives better progress per step towards the end of iteration than other line search rules in our numerical tests.

The fixed step length in rule (ii) is related to the parameter γ of the problem, which is usually unavailable. According to Theorem 3.2, the convergence ratios in Theorem 3.1 for problems with $\gamma^2 \geq \frac{1}{2}$ could be approached with the adaptive step lengths in rule (iii). Moreover, these step lengths will eventually be fixed after a finite number of iterations. Comparing the estimations in Theorem 3.2 with the ones in Theorem 3.1, one may get the impression that a choice of δ close to 1 would eventually give better ratios per step. But such a choice will, at the same time, increase the number of trials in identifying the proper step length. Hence in the practical implementation of rule (iii), one must compromise between these two ends. Besides that, one can also start the trial of j there with some negative integer instead of 0. Then the step length would be allowed to increase if a larger progress in the line search is possible.

4. Numerical test results and other update schemes. Although the estimated rates for PDSD–2 are better than the ones for PDSD–1, we find in our numerical tests that their practical performances are actually very close. As a comparison, we ran PDSD–1(i) and PDSD–2(i) on the *transverse family* of the test problems 0.4–9.4 used in [1], where both the primal and the dual dimensions were 5140. The stopping criterion in the optimality test for the practical implementation of the algorithm is

$$\min\{\, f(u^\nu), f\big(G(v^\nu)\big)\,\} - \max\{\, g(v^\nu), g\big(F(u^\nu)\big)\,\} \leq \varepsilon, \tag{4.1}$$

where $\varepsilon > 0$ is a prespecified threshold for the duality gap. The results in terms of CPU times, as well as numbers of iterations, are given in Table 1. For instance, 45(8/6) in the iterations column of PDSD–2(i) for Problem 0.4 means that the algorithm terminates successfully in 45 iterations, with 8 interactive primal restarts, and 6 interactive dual restarts during the process. (The tests were run on a DECstation 3100 with double precision, where the software had been updated since the test in [1].)

We also tried the algorithm without the primal-dual feedback in the update, i.e., take

$$u^{\nu+1} := \hat{u}^{\nu+1} \quad \text{and} \quad v^{\nu+1} := \hat{v}^{\nu+1}$$

directly in the updating of Step 3. Then the algorithm will generate two unrelated sequences in primal and dual variables, respectively, until the stopping criterion (4.1) on the duality gap is satisfied. We refer to this *extra* version for test purposes as PDSD–0. (In the case of perfect line search, it can be proved by using [1, Prop. 5.1] that the dual part of this extra version reduces to a special case of the finite generation algorithm [2].) The corresponding results are shown in the columns headed PDSD–0(i). The notation ** in these columns signifies that the algorithm failed to meet the termination criterion in 100 iterations, in which case the figure for CPU time is preceded by * since it only indicates the time of the first 100 iterations. The test results show clearly the importance of the primal-dual feedback. Both PDSD–1(i) and PDSD–2(i) perform much better than PDSD–0(i).

TABLE 1

Test results on problems 0.4–9.4 [1].

		CPU time (sec.)			Iterations		
Prb.	Size	PDSD–1(i)	–2(i)	–0(i)	PDSD–1(i)	–2(i)	–0(i)
0.4	5140	110	141	*337	32(7/6)	45(8/6)	**
1.4	5140	183	172	*356	52(4/6)	50(3/6)	**
2.4	5140	147	224	*341	42(8/4)	67(10/3)	**
3.4	5140	35	42	212	9(4/4)	13(3/3)	68
4.4	5140	72	72	*346	19(7/4)	22(6/4)	**
5.4	5140	51	66	178	13(6/4)	20(7/2)	52
6.4	5140	62	74	82	16(5/7)	23(7/7)	24
7.4	5140	64	72	92	18(8/3)	22(6/3)	28
8.4	5140	189	180	*341	55(5/5)	54(3/4)	**
9.4	5140	62	65	110	17(6/7)	20(4/1)	35

There are other possible variants for the algorithm. Notice that the iteration of PDSD–2(i) can be written as

$$u^{\nu+1} := \operatorname{argmin}\{\, f(u) \mid u \in [u^\nu, G(F(u^\nu))] \quad \text{or} \quad u \in G(v^\nu)\,\},$$
$$v^{\nu+1} := \operatorname{argmax}\{\, g(v) \mid v \in [v^\nu, F(G(v^\nu))] \quad \text{or} \quad v \in F(u^\nu)\,\}.$$

This suggests a third update scheme with four perfect line searches in each iteration:

$$u^{\nu+1} := \operatorname{argmin}\{\, f(u) \mid u \in [u^\nu, G(F(u^\nu))] \quad \text{or} \quad u \in [G(v^\nu), G(F(G(v^\nu)))]\,\}, \tag{4.2}$$
$$v^{\nu+1} := \operatorname{argmax}\{\, g(v) \mid v \in [v^\nu, F(G(v^\nu))] \quad \text{or} \quad v \in [F(u^\nu), F(G(F(u^\nu)))]\,\}. \tag{4.3}$$

Obviously, it should converge at least as fast as PDSD–2(i).

Recall that the intermediate points resulted from line searches on $[u^\nu, G(F(u^\nu))]$ and $[v^\nu, F(G(v^\nu))]$ are denoted by $\hat{u}^{\nu+1}$ and $\hat{u}^{\nu+1}$, respectively. Let $\tilde{u}^{\nu+1}$ and $\tilde{u}^{\nu+1}$ be the corresponding line search results in primal and dual on $[G(v^\nu), G(F(G(v^\nu)))]$ and $[F(u^\nu), F(G(F(u^\nu)))]$, respectively. With reasoning similar to the one leading to (3.8), we are able to get

$$f(u^\nu) - f(\hat{u}^{\nu+1}) \geq \big(f(u^\nu) - g(F(u^\nu))\big)\theta(\gamma^2), \tag{4.4}$$
$$g(\hat{u}^{\nu+1}) - g(u^\nu) \geq \big(f(G(v^\nu)) - g(v^\nu)\big)\theta(\gamma^2), \tag{4.5}$$
$$f(G(v^\nu)) - f(\tilde{u}^{\nu+1}) \geq \big(f(G(v^\nu)) - g(F(G(v^\nu)))\big)\theta(\gamma^2), \tag{4.6}$$
$$g(\tilde{u}^{\nu+1}) - g(F(v^\nu)) \geq \big(f(G(F(u^\nu))) - g(F(u^\nu))\big)\theta(\gamma^2). \tag{4.7}$$

Now (4.4) and (4.5) yield

$$f(\hat{u}^{\nu+1}) - g(\hat{v}^{\nu+1}) \leq \big(1-\theta(\gamma^2)\big)\big(f(u^\nu) - g(v^\nu)\big) - \theta(\gamma^2)\big(f(G(v^\nu)) - g(F(u^\nu))\big), \tag{4.8}$$

while (4.5) and (4.6) yield

$$f(\tilde{u}^{\nu+1}) - g(\tilde{v}^{\nu+1}) \leq \big(1-\theta(\gamma^2)\big)\big(f(G(v^\nu)) - g(F(u^\nu))\big) - \theta(\gamma^2)\big(f(G(F(u^\nu))) - g(F(G(v^\nu)))\big). \tag{4.9}$$

Eliminating the term $f(G(v^\nu)) - g(F(u^\nu))$ in (4.8) and (4.9), we get

$$\begin{aligned}&\big(1-\theta(\gamma^2)\big)\big(f(\hat{u}^{\nu+1}) - g(\hat{v}^{\nu+1})\big) + \theta(\gamma^2)\big(f(\tilde{u}^{\nu+1}) - g(\tilde{v}^{\nu+1})\big)\\ &\quad\leq \big(1-\theta(\gamma^2)\big)^2\big(f(u^\nu) - g(v^\nu)\big) - \theta(\gamma^2)\big(f(G(F(u^\nu))) - g(F(G(v^\nu)))\big).\end{aligned} \tag{4.10}$$

According to the update scheme, the duality gap $\varepsilon_{\nu+1}$ should be no larger than either $f(\hat{u}^{\nu+1}) - g(\hat{v}^{\nu+1})$ or $f(\tilde{u}^{\nu+1}) - g(\tilde{v}^{\nu+1})$ or $f(G(F(u^\nu))) - g(F(G(v^\nu)))$. Hence we obtain an estimate

$$\frac{\varepsilon_{\nu+1}}{\varepsilon_\nu} \leq \frac{\big(1-\theta(\gamma^2)\big)^2}{1+\big(\theta(\gamma^2)\big)^2}$$

from (4.10) for the third update scheme in (4.2) and (4.3). For problems with large γ, this is a slightly better result compared with (3.5) for PDSD–2(i) at the cost of two additional line searches.

Acknowledgments. The author would like to thank two anonymous referees for their very helpful comments and suggestions. The third update scheme in (4.2) and (4.3) was due to a suggestion by one of the referees.

REFERENCES

[1] C. ZHU AND R. T. ROCKAFELLAR, *Primal-dual projected gradient algorithms for extended linear-quadratic programming*, SIAM J. Optim. 3 (1993), pp. 751–783.

[2] R.T. ROCKAFELLAR AND R.J.-B. WETS, *A Lagrangian finite generation technique for solving linear-quadratic problems in stochastic programming*, Math. Programming Stud., 28 (1986), pp. 63–93.

[3] ———, *Linear-quadratic problems with stochastic penalties: the finite generation algorithm*, in Numerical Techniques for Stochastic Optimization Problems, Y. Ermoliev and R. J.-B. Wets, eds., Lecture Notes in Control and Information Sciences No. 81, Springer-Verlag, 1987, pp. 545–560.

[4] R. T. ROCKAFELLAR, *A generalized approach to linear-quadratic programming*, in Proc. International Conf. on Numerical Optimization and Appl., Xi'an, China, 1986, pp. 58–66.

[5] ———, *Linear-quadratic programming and optimal control*, SIAM J. Control Optim., 25 (1987), pp. 781–814.

[6] 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.

[7] R.T. ROCKAFELLAR, *Computational schemes for solving large-scale problems in extended linear-quadratic programming*, Math. Programming, 48 (1990), pp. 447–474.

[8] ———, *Large-scale extended linear-quadratic programming and multistage optimization*, in Proc. Fifth Mexico–U.S. Workshop on Numerical Analysis, S. Gomez, J.-P. Hennart, R. Tapia, eds., Society for Industrial and Applied Mathematics, Philadelphia, 1990.

[9] A. KING, *An implementation of the Lagrangian finite generation method*, in Numerical Techniques for Stochastic Programming Problems, Y. Ermoliev and R. J.-B. Wets, eds., Springer-Verlag, New York, Berlin, 1988.

[10] J. M. WAGNER, *Stochastic Programming with Recourse Applied to Groundwater Quality Management*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, 1988.

[11] C. ZHU, *Modified proximal point algorithm for extended linear-quadratic programming*, Computational Optim. Appl., 1 (1992), pp. 185–205.

SIAM J. OPTIMIZATION
Vol. 5, No. 1, pp. 129–148, February 1995

007

A POSITIVE ALGORITHM FOR THE NONLINEAR COMPLEMENTARITY PROBLEM*

RENATO D. C. MONTEIRO†, JONG-SHI PANG‡, AND TAO WANG‡

Abstract. In this paper, the authors describe and establish the convergence of a new iterative method for solving the (nonmonotone) nonlinear complementarity problem (NCP). The method utilizes ideas from two distinct approaches for solving this problem and combines them into one unified framework. One of these is the infeasible-interior-point approach that computes an approximate solution to the NCP by staying in the interior of the nonnegative orthant; the other approach is typified by the NE/SQP method which is based on a generalized Gauss–Newton scheme applied to a constrained nonsmooth-equations formulation of the complementarity problem. The new method, called a *positive algorithm* for the NCP, generates a sequence of positive vectors by solving a sequence of linear equations (as in a typical interior-point method) whose solutions (if nonzero) provide descent directions for a certain merit function that is derived from the NE/SQP iteration function modified for use in an interior-point context.

Key words. complementarity problems, interior-point methods, nonsmooth equations

AMS subject classifications. 90C30, 90C33, 49M37

1. Introduction.

The idea of solving complementarity problems by staying in the interior of the feasible region can be traced to a paper published in 1980 by McLinden [18]. Although no explicit algorithm was formulated, the idea of tracing an interior path as a possible solution procedure was quite evident in this paper and the existence of the "central path" was demonstrated in the case of a complementarity problem with a maximal monotone multifunction. Unfortunately, this paper was not widely known. Of course, McLinden's idea is central to many of today's interior-point methods for solving a wide variety of mathematical programming problems.

In recent years, interior-point methods for solving complementarity problems have been the subject of many studies [4], [6]–[14], [19], [20], [26], [28], [31]–[35]. Among these, the monograph [9] presents a unified treatment of the original family of (feasible) interior-point methods for the linear complementarity problem (LCP) that requires all iterates to be strictly feasible; this volume also contains an extensive list of references for the interior-point methods up to the year 1990.

A proposal by Lustig [16] and the subsequent computational study [17] have led researchers to investigate the family of infeasible interior-point methods. The main feature of these methods for solving a complementarity problem is that the iterates are positive vectors, albeit not necessarily feasible to the problem, and have some desirable limiting properties. There are many papers dealing with these methods for solving linear programs; for the linear complementarity problem, we mention [31] and [34]. Most recently, the paper by Kojima, Noma, and Yoshise [15] presents a wide class

* Received by the editors December 31, 1992; accepted for publication (in revised form) November 15, 1993.

† School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332 (monteiro@isye.gatech.edu). The work of this author was based on research supported by National Science Foundation grant DDM-9109404 and Office of Naval Research grant N00014-93-1-0234. This work was done while this author was an assistant professor in the Systems and Industrial Engineering Department of the University of Arizona.

‡ Department of Mathematical Sciences, The Johns Hopkins University, Baltimore, Maryland 21218 (jsp@vicp.mts.jhu.edu) and (wang_tbrutus.mts.jhu.edu). The work of these authors was based on research supported by National Science Foundation grants DDM-9104078 and CCR-9213739 and Office of Naval Research grant N00014-93-1-0228.

of infeasible interior point methods for solving a monotone nonlinear complementarity problem (NCP).

Inclusive of the early work of McLinden, all interior-point methods for solving complementarity problems to date have invariably relied on a certain monotonicity assumption (or more generally, a $\mathbf{P}_0$-property) see [8], [9], [15] and [1, §5.9]. This fact helps to explain why the interior-point methods proposed so far for solving nonlinear programs are restricted to the class of linearly constrained convex programs [21]–[23]. In essence, such a monotonicity assumption (or $\mathbf{P}_0$-property) is needed to ensure the nonsingularity of a key matrix that is used to define the main computational step of the methods. The major objective of this paper is to propose an interior-point method for solving a general, nonmonotone NCP. In a subsequent paper, we study the specialization of this method to the Karush–Kuhn–Tucker optimality conditions of a general nonlinear program formulated as a mixed NCP.

Our proposed method is an infeasible interior-point potential reduction algorithm; it involves two major ideas. One is to maintain the positivity of the iterates while a certain merit function is being decreased; this joint task is accomplished by means of the modified Armijo technique described in [22]. The other idea is to make use of the iteration function in the recent NE/SQP method [2], [27] to define a suitable merit function. The resulting algorithm does not rely on any monotonicity assumption of the problem. Instead, a key condition, called *s-regularity*, plays an important role in the convergence analysis.

This interior-point method, which we call a *positive algorithm* for the NCP to signify that the iterates are positive vectors but not necessarily feasible to the problem, consists of solving a sequence of linear equations each defined by a symmetric positive definite matrix; the unique solutions of these equations, if nonzero, provide descent directions for a special logarithmic merit function, which is a combination of the NE/SQP iteration function and the positivity conditions. The NE/SQP method also maintains the nonnegativity of the iterates; but it does so by imposing this requirement as constraints in the direction-finding subproblems that are then either (convex) quadratic [27] or linear programs [2]. Consequently, the positive algorithm may be considered as providing an interior-point approach to alleviate the direction-finding task in the NE/SQP method.

2. Preliminaries. Given a function $f : R^n \to R^n$, which is assumed to be continuously differentiable in an open set containing R^n_+, the nonlinear complementarity problem, denoted NCP (f), is to find a vector $x \in R^n$ such that

$$x \geq 0, \quad f(x) \geq 0, \quad x^T f(x) = 0.$$

The reader is referred to [5] for a comprehensive review of the theory and applications of this problem. In [27], the NE/SQP method was proposed as a solution procedure for this problem. Clearly, the NCP (f) is equivalent to the following nonnegatively constrained system of nonsmooth equations:

$$H(x) := \min(x, f(x)) = 0, \qquad x \geq 0, \tag{1}$$

where min denotes the componentwise minimum of two vectors. The NE/SQP method generates a sequence of nonnegative iterates by successively solving a sequence of nonnegatively constrained least-squares subproblems whose solutions provide descent directions for the merit function

$$\theta(x) = H(x)^T H(x).$$

Exploiting the idea of staying in the positive orthant, we describe an iterative algorithm for solving the NCP (f) in which each direction-finding step requires the solution of a single system of linear equations.

It would be useful to summarize the key properties of the norm function θ. Clearly, θ is nonnegative; its zeros are precisely the solutions of the NCP (f). The function θ is generally not Fréchet differentiable at an arbitrary vector, but it has a strong Fréchet derivative at all its zeros [25, Prop. 1]. Moreover, θ is directionally differentiable everywhere with the directional derivative at a vector x along the direction d given by [24]

$$\theta'(x,d) = \sum_{i:x_i<f_i(x)} x_i d_i + \sum_{i:x_i=f_i(x)} x_i \min(d_i, \nabla f_i(x)^T d) + \sum_{i:x_i>f_i(x)} f_i(x)\nabla f_i(x)^T d.$$

Motivated by this expression, we define three fundamental index sets associated with an arbitrary vector $z \in R^n$:

$$I_x(z) = \{i : z_i < f_i(z)\}, \quad I_e(z) = \{i : z_i = f_i(z)\}, \quad I_f(z) = \{i : z_i > f_i(z)\}.$$

For notational convenience, we let $J_f(z) = I_x(z) \cup I_e(z)$. We note immediately that if $x \in R^n$ is nonnegative, then for all vectors $d \in R^n$,

$$\theta'(x,d) \le \sum_{i\in J_f(x)} x_i d_i + \sum_{i\in I_f(x)} f_i(x)\nabla f_i(x)^T d.$$

This inequality is the key to the descent step in the algorithm to be described later. To write the inequality in a more compact form, we define the $n \times n$ matrix $A(x)$ whose ith column is given by

$$A(x)_i = \begin{cases} e^i & \text{if } i \in J_f(x), \\ \nabla f_i(x) & \text{if } i \in I_f(x), \end{cases}$$

where e^i is the ith coordinate vector. In terms of this matrix, the above inequality becomes

$$\theta'(x,d) \le H(x)^T A(x)^T d. \tag{2}$$

It is important to note that the directional derivative $\theta'(x,d)$ is generally not a continuous function in x for a fixed but arbitrary d and neither is the matrix-valued function $A(x)$. However, the next result gives an important generalization of the inequality (2). A variant of this result can be found in [27, Lem. 4]. For the sake of completeness, we give a simpler and more direct proof of this result here.

PROPOSITION 2.1. *Let $x^* \in R^n_{++}$ be arbitrary. Then for any sequence $\{x^k\} \subseteq R^n_{++}$ converging to x^* and any sequence $\{h^k\} \subseteq R^n\backslash\{0\}$ converging to 0, there holds*

$$\limsup_{k\to\infty} \frac{\theta(x^k+h^k) - \theta(x^k) - H(x^k)^T A(x^k)^T h^k}{\|h^k\|} \le 0. \tag{3}$$

Proof. For each $i = 1, \ldots, n$, let

$$\Delta H_i(x^k,h^k) \equiv \frac{1}{2}H_i^2(x^k+h^k) - \frac{1}{2}H_i^2(x^k) - H_i(x^k)[A(x^k)_i]^T h^k,$$
$$\Delta f_i(x^k,h^k) \equiv \frac{1}{2}f_i^2(x^k+h^k) - \frac{1}{2}f_i^2(x^k) - f_i(x^k)\nabla f_i(x^k)^T h^k,$$
$$\Delta x_i(x^k,h^k) \equiv \frac{1}{2}(x_i^k+h_i^k)^2 - \frac{1}{2}(x_i^k)^2 - x_i^k h_i^k.$$

We claim that for all k sufficiently large,

$$\Delta H_i(x^k, h^k) \leq \max\{\Delta f_i(x^k, h^k)\,,\, \Delta x_i(x^k, h^k)\}. \tag{4}$$

Indeed, there are three cases to consider: whether $i \in I_f(x^*)$, $i \in I_x(x^*)$, or $i \in I_e(x^*)$. Assume first that $i \in I_f(x^*)$, that is $f_i(x^*) < x_i^*$. Using the fact that both sequences $\{x^k\}$ and $\{x^k + h^k\}$ converge to x^* and a simple continuity argument, we obtain $f_i(x^k) < x_i^k$ and $f_i(x^k + h^k) < x_i^k + h_i^k$ for all k sufficiently large. Using the definition of the functions $H(\cdot)$ and $A(\cdot)$, we then obtain

$$\Delta H_i(x^k, h^k) = \Delta f_i(x^k, h^k);$$

hence (4) holds. For the case in which $i \in I_x(x^*)$, that is $f_i(x^*) > x_i^*$, we can similarly show that

$$\Delta H_i(x^k, h^k) = \Delta x_i(x^k, h^k);$$

so (4) also holds. Consider now the case in which $i \in I_e(x^*)$, that is $f_i(x^*) = x_i^* > 0$. Then, for all k sufficiently large, we have $f_i(x^k + h^k) > 0$ and $x_i^k + h_i^k > 0$. This implies that

$$H_i^2(x^k + h^k) = \min\{f_i^2(x^k + h^k)\,,\, (x_i^k + h_i^k)^2\}.$$

Using this relation and considering whether $i \in I_f(x^k)$ or $i \in J_f(x^k)$, we can easily verify that (4) holds. We can now complete the proof of (3). Indeed, using (4) we obtain that

$$\limsup_{k\to\infty} \frac{\Delta H_i(x^k, h^k)}{\|h^k\|} \leq 0 \quad \forall i = 1, \ldots, n.$$

Since

$$\theta(x^k + h^k) - \theta(x^k) - H(x^k)^T A(x^k)^T h^k = \sum_{i=1}^{n} \Delta H_i(x^k, h^k),$$

relation (3) follows. □

3. Some important functions. The merit function to be used in the algorithm is defined as follows. Let $c > 0$ and $\zeta > n$ be given scalars. Define

$$\Omega \equiv \{x \in R^n_{++} \mid \theta(x) > 0\}$$

and let

$$\psi_c(x) \equiv c \log \theta(x) + \zeta \log(\theta(x) + e^T x) - \sum_{i=1}^{n} \log x_i \quad \forall x \in \Omega,$$

where e is the vector of all ones. The scalar c is a penalty parameter that will be changed in the algorithm if it is deemed to be too small; unlike c, ζ is fixed throughout. The third term in the function ψ_c is the logarithmic barrier function to prevent the iterates from reaching the boundary of the nonnegative orthant. The middle term is used to balance the third term; this is analogous to the potential function introduced in [30] for linear programs that have been used extensively in many primal-dual interior-point methods; see also [15] for a related merit function.

Clearly, we have

$$\psi_c(x) \geq c \log \theta(x) + (\zeta - n) \log(\theta(x) + e^T x) \quad \forall x \in \Omega$$

and

$$\psi_c(x) \geq (c + \zeta) \log \theta(x) - \sum_{i=1}^{n} \log x_i \quad \forall x \in \Omega.$$

These inequalities have two important implications that we summarize in the result below.

PROPOSITION 3.1. *For a fixed $c > 0$, the following statements hold:*
(a) *if $\{x^k\} \subseteq \Omega$ and $\lim_{k\to\infty} \psi_c(x^k) = -\infty$, then $\lim_{k\to\infty} \theta(x^k) = 0$;*
(b) *for any $\alpha > 0$ and $t \in R$, there exist constants $a > 0$ and $b > 0$ such that*

$$[x \in R^n_{++}, \psi_c(x) \leq t, \theta(x) \geq \alpha] \implies a \leq x_i \leq b \quad \forall i = 1, \ldots, n.$$

The function ψ_c is directionally differentiable everywhere with the directional derivative at the vector $x \in \Omega$ along the direction $d \in R^n$ given by

$$\psi_c'(x, d) = \frac{c}{\theta(x)} \theta'(x, d) + \frac{\zeta}{\theta(x) + e^T x} (\theta'(x, d) + e^T d) - \sum_{i=1}^{n} \frac{d_i}{x_i}.$$

Recalling the inequality (2), we define the forcing function

$$z_c(x, d) = \frac{c}{\theta(x)} H(x)^T A(x)^T d + \frac{\zeta}{\theta(x) + e^T x} (A(x)H(x) + e)^T d - \sum_{i=1}^{n} \frac{d_i}{x_i}$$

whose role in the algorithm will become obvious momentarily. Clearly, by (2), we have

$$\psi_c'(x, d) \leq z_c(x, d) \tag{5}$$

for all $x \in R^n_{++}$ with $\theta(x) > 0$ and all $d \in R^n$. Consequently, given such a vector x, if we can find a vector d such that $z_c(x, d) < 0$, then d is a descent direction for the function ψ_c at x. To generate such a direction, we consider the system of linear equations

$$(A(x)A(x)^T + X^{-2})d + w_c(x) = 0, \tag{6}$$

where

$$w_c(x) = \frac{c}{\theta(x)} A(x)H(x) + \frac{\zeta}{\theta(x) + e^T x} (A(x)H(x) + e) - x^{-1},$$

with X being the diagonal matrix with x_i's on the diagonal and x^{-1} being the vector whose ith component is equal to $1/x_i$. The system of linear equations (6) is equivalent to the least-squares problem

$$\text{minimize} \quad (\|A(x)^T d\|_2^2 + \|X^{-1} d\|_2^2) + z_c(x, d),$$

where the minimization is over all vectors $d \in R^n$ with the vector x fixed.

Noting that the matrix

$$M(x) = A(x)A(x)^T + X^{-2} \tag{7}$$

is symmetric positive definite, we let d_x be the unique solution of (6). Then we have

$$z_c(x, d_x) = -(\|A(x)^T d_x\|_2^2 + \|X^{-1} d_x\|_2^2) \leq 0;$$

moreover, $z_c(x, d_x) < 0$ if and only if $w_c(x) \neq 0$. Consequently, if $w_c(x) \neq 0$, then d_x is a descent direction for the function ψ_c at the point x. Nevertheless, if $w_c(x)$ is equal to zero, then $d_x = 0$ and we need to generate an alternate direction. In this case, we double the penalty constant c (actually, any scaling exceeding 1 suffices) and solve another system of the form (6) with the modified c.

With a (nonzero) descent direction d_x successfully computed, we then perform a line search starting at x with the objective of decreasing the merit function ψ_c by a sufficient amount while preserving the positivity of the next iterate. Details of such a line search step can be found in [22]. The main procedure is then repeated with the new iterate replacing the old one if termination with regard to a prescribed rule still has not occurred.

We close this section with an immediate consequence of Proposition 2.1.

PROPOSITION 3.2. *Let $x^* \in \Omega$ be arbitrary. Then for any sequence $\{x^k\} \subseteq R^n_{++}$ converging to x^*, any sequence $\{d^k\}$ with $\{(X^k)^{-1} d^k\}$ bounded, and any sequence $\{\lambda_k\}$ of positive scalars converging to zero, there holds*

$$\limsup_{k \to \infty} \frac{\psi_c(x^k + \lambda_k d^k) - \psi_c(x^k) - \lambda_k z_c(x^k, d^k)}{\lambda_k} \leq 0. \tag{8}$$

Proof. Since $\{(X^k)^{-1} d^k\}$ is bounded, $\{\lambda_k\} \to 0$ and $x_i^k + \lambda_k d_i^k = x_i^k(1 + \lambda_k d_i^k / x_i^k)$, it follows that $x^k + \lambda_k d^k$ is positive for all k sufficiently large. Using the fact that

$$\log(1 + s) = s + o(s), \quad \text{where } \lim_{s \to 0} \frac{o(s)}{s} = 0,$$

we obtain that for each $i = 1, \ldots, n$,

$$\lim_{k \to \infty} \frac{\log(x_i^k + \lambda_k d_i^k) - \log x_i^k - \lambda_k d_i^k / x_i^k}{\lambda_k} = \lim_{k \to \infty} \frac{\log(1 + \lambda_k d_i^k / x_i^k) - \lambda_k d_i^k / x_i^k}{\lambda_k} = 0.$$

Using this limit, Proposition 2.1 and the definition of $\psi_c(\cdot)$, we can now easily derive (8). □

4. The positive algorithm. Summarizing the ideas outlined in the previous section, we now present the details of the long awaited algorithm for solving the NCP (f), where f is an arbitrary continuously differentiable function.

Step 0. (Initialization) Let $\zeta > n$, $\delta > 0$, and $\sigma, \alpha, \rho \in (0, 1)$ be given constants. Choose a scalar $c_0 > 0$ and a vector $x^0 > 0$ arbitrarily. Set $k = 0$.

Step 1. (Direction generation) Compute $w^k = w_{c_k}(x^k)$. If $\|w^k\| \leq \delta$, set

$$c_{k+1} = 2c_k \quad \text{and} \quad x^{k+1} = x^k.$$

Replace k by $k + 1$ and return to the beginning of this step. If $\|w^k\| > \delta$, let d^k be the unique solution to the system of linear equations:

$$M(x^k)d + w^k = 0. \tag{9}$$

Step 2. (Armijo line search) Determine the maximum stepsize

$$\tau_k^0 = \sup\{\tau : x^k + \tau d^k \geq 0\}$$

and let

$$\hat{\tau}_k \begin{cases} = \alpha\tau_k^0 & \text{if } \tau_k^0 < \infty, \\ \geq 1/\|d^k\| & \text{if } \tau_k^0 = \infty. \end{cases}$$

Let m_k be the smallest nonnegative integer m such that

$$\psi_{c_k}(x^k + \hat{\tau}_k\rho^m d^k) - \psi_{c_k}(x^k) < \sigma\hat{\tau}_k\rho^m z_{c_k}(x^k, d^k). \tag{10}$$

Set

$$c_{k+1} = c_k \quad \text{and} \quad x^{k+1} = x^k + \hat{\tau}_k\rho^{m_k} d^k.$$

Step 3. (Termination check) If x^{k+1} fails a termination check (for example, if $\theta(x^{k+1}) > \varepsilon$ for a prescribed tolerance $\varepsilon > 0$), return to step 1 with k replaced by $k+1$.

We make several remarks about the above algorithm. First, the use of the scalar δ to guard against a zero vector $w_{c_k}(x^k)$ is an extension of the discussion in the last section. As we shall see from the convergence analysis in §5, it is not enough to just check whether the vector $w_{c_k}(x^k)$ is zero or not; we actually need to ensure that this vector is not too small in norm for the direction d^k to be useful. Second, the maximum stepsize τ_k^0 and the scalar $\alpha \in (0,1)$ together will ensure that the next iterate x^{k+1} remains a positive vector. Finally, a standard argument in an Armijo line search and the inequality (5) will ensure that the integer m_k can be determined in a finite number of trials; this proof is omitted. Moreover, in case $m_k \geq 1$, we must have

$$\psi_{c_k}(x^k + \hat{\tau}_k\rho^{m_k-1} d^k) - \psi_{c_k}(x^k) \geq \sigma\hat{\tau}_k\rho^{m_k-1} z_{c_k}(x^k, d^k) \tag{11}$$

by the definition of m_k.

It would be useful to compare the positive algorithm with the framework proposed in [15] for solving the NCP (f). For this purpose, we consider this problem as being defined by the following conditions:

$$y - f(x) = 0, \tag{12}$$

$$x \geq 0, \tag{13}$$

$$y \geq 0, \tag{14}$$

$$x^T y = 0. \tag{15}$$

The positive algorithm generates a sequence of iterates $\{x^k\}$, which induces a corresponding sequence $\{y^k\}$ via the relation $y^k = f(x^k)$ for all k. Hence, the combined sequence $\{(x^k, y^k)\}$ satisfies the conditions (12) and (13) but not necessarily (14) or (15); in fact, the latter two conditions are the goal of the positive algorithm. On the other hand, the methods described in [15] generate a sequence $\{(x^k, y^k)\}$ that satisfies (13) and (14) but not necessarily (12) or (15), which is the goal of these other infeasible interior-point methods for solving the NCP (f).

5. Convergence analysis. In this section, we analyze the limiting properties of an infinite sequence $\{x^k\}$ generated by the positive algorithm. By the infinite nature of this sequence, we have $\theta(x^k) > 0$ for all k. We divide the analysis into two cases, depending on whether the penalty constant c_k is updated infinitely often or only finitely many times. We first take up the latter case.

Finite update of c. The next result analyzes the case in which $\|w^k\| > \delta$ for all indices k sufficiently large.

THEOREM 5.1. *Suppose that the penalty sequence $\{c_k\}$ is updated finitely many times in the positive algorithm. Then,*

$$\lim_{k\to\infty} \theta(x^k) = 0. \tag{16}$$

Consequently, every accumulation point of $\{x^k\}$, if it exists, must be a solution of the NCP *(f).*

Proof. Since $\{c_k\}$ is updated finitely many times, there exists an index $k_0 \geq 0$ and a constant $c > 0$ such that $c_k = c$ for all $k \geq k_0$. Assume by contradiction that (16) does not hold. Then there exist a constant $\varepsilon > 0$ and a subsequence $\{x^k\}_{k\in\mathcal{K}}$ such that $\mathcal{K} \subseteq \{k_0, k_0+1, \ldots\}$ and

$$\inf_{k\in\mathcal{K}} \theta(x^k) \geq \varepsilon. \tag{17}$$

Inequality (10) and the fact that $z_c(x^k, d^k) < 0$, for all $k \geq k_0$, imply that $\{\psi_c(x^k) : k \geq k_0\}$ is decreasing. Moreover, (17) and Proposition 3.1(a) imply that $\{\psi_c(x^k) : k \in \mathcal{K}\}$ is bounded below. Hence, this sequence converges and, by (10), we have

$$\lim_{k(\in\mathcal{K})\to\infty} \hat{\tau}_k \rho^{m_k} z_c(x^k, d^k) = 0. \tag{18}$$

We next show that

$$\inf_{k\in\mathcal{K}} \frac{|z_c(x^k, d^k)|}{\|d^k\|} > 0. \tag{19}$$

Indeed, we know that $\{x^k\}_{k\in\mathcal{K}} \subseteq \{x \in R^n_{++} : \psi_c(x) \leq t,\ \theta(x) \geq \varepsilon\}$, where $t \equiv \psi_c(x^{k_0})$. By Proposition 3.1(b), $\{x^k\}_{k\in\mathcal{K}}$ is bounded and for all $k \in \mathcal{K}$,

$$x^k_i \geq a, \quad i = 1, \ldots, n \tag{20}$$

for some constant $a > 0$. Using this fact, we can easily show that

$$\|M(x^k)\|\ \|M(x^k)^{-1}\| \leq T \quad \forall k \in \mathcal{K} \tag{21}$$

for some constant $T > 0$. Hence, using the fact that $\|w_c(x^k)\| \geq \delta$, we obtain that for all $k \in \mathcal{K}$,

$$\begin{aligned}\frac{|z_c(x^k, d^k)|}{\|d^k\|} = \frac{|w_c(x^k)^T d^k|}{\|d^k\|} &= \frac{\|w_c(x^k)^T M(x^k)^{-1} w_c(x^k)\|}{\|M(x^k)^{-1} w_c(x^k)\|} \\ &\geq \frac{\|M(x^k)\|^{-1}\|w_c(x^k)\|^2}{\|M(x^k)^{-1}\|\ \|w_c(x^k)\|} = \frac{\|w_c(x^k)\|}{\|M(x^k)\|\ \|M(x^k)^{-1}\|} \geq \frac{\delta}{T}.\end{aligned}$$

Hence, (19) holds. Combining (18) and (19), we obtain

$$\lim_{k(\in\mathcal{K})\to\infty} \hat{\tau}_k \rho^{m_k} \|d^k\| = 0. \tag{22}$$

We next show that

$$\inf_{k\in\mathcal{K}} \hat{\tau}_k \|d^k\| > 0. \tag{23}$$

Indeed, if $k \in \mathcal{K}$ is such that $\tau_k^0 = \infty$, then

$$\hat{\tau}_k \|d^k\| \geq 1$$

by the definition of $\hat{\tau}_k$. On the other hand, if $\tau_k^0 < \infty$, then using (20) and the fact that $x^k + \tau_k^0 d^k$ has some component equal to zero, we can easily deduce that

$$\tau_k^0 \|d^k\| \geq a. \tag{24}$$

Consequently, (23) holds since $\hat{\tau}_k = \alpha\tau_k^0$ for all $k \geq 0$. Combining (22) and (23), we obtain $\lim_{k(\in\mathcal{K})\to\infty} \rho^{m_k} = 0$. Hence, $m_k \geq 1$ for all $k \in \mathcal{K}$ sufficiently large. Since $\{x^k\}_{k\in\mathcal{K}}$ is bounded, we may take $\mathcal{K}' \subseteq \mathcal{K}$ such that $\lim_{k(\in\mathcal{K}')\to\infty} x^k = x^* \in R^n_{++}$ and $m_k \geq 1$ for all $k \in \mathcal{K}'$. By (11), we obtain

$$\frac{\psi_c(x^k + \hat{\tau}_k \rho^{m_k-1} d^k) - \psi_c(x^k)}{\hat{\tau}_k \rho^{m_k-1}\|d^k\|} \geq \sigma \frac{z_c(x^k, d^k)}{\|d^k\|},$$

or equivalently,

$$\frac{\psi_c(x^k + \lambda_k p^k) - \psi_c(x^k) - \lambda_k w_c(x^k)^T p^k}{\lambda_k} \geq -(1-\sigma)\frac{z_c(x^k, d^k)}{\|d^k\|} \geq \frac{(1-\sigma)\delta}{T}, \tag{25}$$

where $\lambda_k \equiv \hat{\tau}_k \rho^{m_k-1}\|d^k\|$ and $p^k \equiv d^k/\|d^k\|$. Note that relation (22) implies that $\lim_{k(\in\mathcal{K})\to\infty} \lambda_k = 0$. Also, it is easy to verify that $\{(X^k)^{-1}p^k\}$ is bounded. Hence, by Proposition 3.2, we know that the lim sup of the left-hand side of (25) is nonpositive and this violates (25). We have thus obtained a contradiction and therefore (16) must hold. □

We point out that other choices for the matrix $M(x)$ used in the computation of the search direction are possible. In addition to the symmetry and positive definiteness of $M(x)$, all that is required is that the condition number $\mathrm{cond}(M(x))$ of the matrix $M(x)$, defined as

$$\mathrm{cond}(M(x)) \equiv \|M(x)\|\,\|M(x)^{-1}\|,$$

be uniformly bounded on any compact subset of R^n_{++}; cf. (21). The above convergence proof of Theorem 5.1 (and thus the limit (16)) remains valid under this condition. Note that this condition is much weaker than the requirement that $\mathrm{cond}(M(x))$ be uniformly bounded on the whole R^n_{++}. Our choice of $M(x)$ in (7) satisfies the first condition but not the latter one.

It is important to point out that Theorem 5.1 is established under absolutely no assumption on the function f other than its continuous differentiability. Note also that the theorem does not require the boundedness of the sequence $\{x^k\}$; indeed, as the following example shows, this sequence may be unbounded if no restriction is imposed on the function f.

Example. Let

$$f(x) = e^{-x}, \qquad x \in R.$$

For $x > 0$ sufficiently large, it is easy to obtain the functions H and w_c as follows:

$$H(x) = e^{-x} \quad \text{and}$$
$$w_c(x) = -2c + \zeta \tfrac{1-e^{-2x}}{x+e^{-2x}} - x^{-1}.$$

Note that $\lim_{x\to\infty} w_c(x) = -2c$; hence, provided that $2c > \delta$, we must have $|w_c(x)| > \delta$ for $x > 0$ sufficiently large. Consequently, corresponding to such an x, the search direction at x is

$$d_x = \frac{-w_c(x)}{e^{-2x} + x^{-2}} > \frac{\delta}{e^{-2x} + x^{-2}} > 0.$$

Thus, if we initiate the positive algorithm with x^0 sufficiently large and the constant $c_0 > \delta/2$, the algorithm will generate an increasing sequence $\{x^k\}$ with c_k remaining constant. This sequence cannot be bounded for otherwise its limit point would be a strictly positive solution of the NCP (f); but the only solution to the NCP (f) is $x = 0$.

Boundedness of iterates. The above example is not surprising because generally, if the NCP (f) has no solution, then although the limiting value of the sequence $\{\theta(x^k)\}$ is zero, the sequence of iterates $\{x^k\}$ must be unbounded. Consequently, some conditions on f must be needed for the latter sequence to be bounded. The following discussion pertains to this boundedness issue of $\{x^k\}$.

We recall some properties of a vector-valued mapping. A mapping $F : R^n \to R^n$ is *norm-coercive* on a set $X \subseteq R^n$ if

$$\lim_{\|x\|\to\infty, x\in X} \|F(x)\| = \infty;$$

F is *coercive* on X in the *Hadamard sense* if

$$\lim_{\|x\|\to\infty, x\in X} \frac{\|x * F(x)\|}{\|x\|} = \infty,$$

where $a*b$ denotes the Hadamard product of two vectors $a, b \in R^n$, i.e., $(a*b)_i = a_i b_i$ for all i. It is easy to see that a mapping F is norm-coercive on X if and only if for all $t \geq 0$, the level set

$$\{x \in X : \|F(x)\| \leq t\}$$

is bounded; moreover, coercivity in the Hadamard sense implies norm-coercivity. Examples of mappings that are coercive on R^n_+ in the Hadamard sense include the strongly copositive mappings and the uniform **P**-functions. The former are those mappings F for which there exists a constant $\gamma_1 > 0$ such that

$$\max_{1\leq i\leq n} x_i F_i(x) \geq \gamma_1 \|x\|_2^2 \quad \text{for all } x \in R^n_+;$$

and the latter are those mappings F for which there exists a constant $\gamma_2 > 0$ such that

$$\max_{1\leq i\leq n} (x_i - y_i)(F_i(x) - F_i(y)) \geq \gamma_2 \|x - y\|_2^2 \quad \text{for all } x, y \in R^n.$$

Given a mapping $F : R^n \to R^n$, a *principal subfunction* of F is defined as follows. For an arbitrary index set $\alpha \subseteq \{1, \ldots, n\}$ with cardinality k and complement β and an

arbitrary vector $a_\beta \in R^{n-k}$, the function $G : R^k \to R^k$ defined by $G(x_\alpha) = F_\alpha(x_\alpha, a_\beta)$ is a k-dimensional principal subfunction of F.

COROLLARY 5.2. *Suppose that the penalty parameter c_k is updated finitely many times in the positive algorithm. Then the sequence $\{x^k\}$ is bounded under any one of the following conditions:*

(a) *there exists a scalar $t > 0$ such that the level set*

$$L_t := \{x \in R^n_{++} : \theta(x) \leq t\}$$

is bounded;

(b) *f is (globally) Lipschitzian on R^n_+ and every k-dimensional principal subfunction $f_\alpha(\cdot, a_\beta)$ of f is norm-coercive on R^k_+ for every fixed vector $a_\beta \in R^{n-k}_+$;*

(c) *f is (globally) Lipschitzian and coercive in the Hadamard sense on R^n_+.*

Proof. By Theorem 5.1, the sequence $\{\theta(x^k)\}$ converges to zero. Hence, for all k sufficiently large, x^k is contained in the level set L_t. Consequently, (a) implies the boundedness of $\{x^k\}$. By the proof of [24, Lem. 4], it follows that (b) implies (a). Finally, we show that (c) implies (b). But this is an easy consequence of the identity

$$a_\beta * f_\beta(x_\alpha, a_\beta) = a_\beta * f_\beta(a_\alpha, a_\beta) + a_\beta * (f_\beta(x_\alpha, a_\beta) - f_\beta(a_\alpha, a_\beta))$$

and the Lipschitzian property of f, which implies that

$$\limsup_{\|x_\alpha\| \to \infty, x_\alpha \geq 0} \frac{\|a_\beta * (f_\beta(x_\alpha, a_\beta) - f_\beta(a_\alpha, a_\beta))\|}{\|x_\alpha\|} < \infty.$$

Thus, by the coercivity of f on R^n_+ in the Hadamard sense, it follows that the principal subfunction $f_\alpha(\cdot, a_\beta)$ must be coercive on R^k_+ in the Hadamard sense. As a consequence of an observation preceding this corollary, it follows that this subfunction is norm-coercive on R^k_+. □

Our next result concerns the LCP that corresponds to the NCP (f) in which f is an affine mapping. The proof of this result requires a fundamental continuity property of the solution set of the LCP regarded as a multifunction of the constant vector of the problem [1, Thm. 7.2.1]. To explain the latter property, consider the LCP defined by the vector $q \in R^n$ and matrix $M \in R^{n \times n}$:

$$x \geq 0, \quad q + Mx \geq 0, \quad x^T(q + Mx) = 0. \tag{26}$$

We let $\mathrm{SOL}_M(q)$ denote the (possibly empty) solution set of this problem. As a multifunction in q, SOL_M is locally upper Lipschitzian in the following sense: for a fixed but arbitrary q, there exist a constant $L > 0$ and a neighborhood V of q such that for all $q' \in V$,

$$\mathrm{SOL}_M(q') \subseteq \mathrm{SOL}_M(q) + L\|q - q'\|_2 \mathcal{B},$$

where $\mathcal{B}$ denotes the (closed) unit ball in R^n with the Euclidean norm. There are two immediate consequences of this result: one is that if $\mathrm{SOL}_M(q^k)$ is nonempty for a sequence of vectors $\{q^k\}$ converging to q, then $\mathrm{SOL}_M(q)$ is nonempty; moreover, if the latter solution set is bounded, then all solution sets $\mathrm{SOL}_M(q')$ with q' sufficiently close to q are uniformly bounded; see [1].

COROLLARY 5.3. *Suppose that the penalty parameter c_k is updated finitely many times when the positive algorithm is applied to the* LCP (26). *Then $\mathrm{SOL}_M(q)$ must*

be nonempty. Moreover, if this solution set is bounded, then there exists a constant $L' > 0$ *such that for all* k *sufficiently large,*

$$d(x^k, \mathrm{SOL}_M(q)) \leq L' \| \min(x^k, q + Mx^k) \|,$$

where $d(x, S)$ *denotes the distance from the vector* x *to the set* S*; in particular, the sequence* $\{x^k\}$ *is bounded.*

Proof. Let $y^k = \min(x^k, q + Mx^k)$. Then Theorem 5.1 implies that the sequence $\{y^k\}$ converges to zero. The definition of y^k implies that the vector $z^k = x^k - y^k \in \mathrm{SOL}_M(q^k)$ where

$$q^k = q + My^k - y^k$$

which clearly converges to q. Hence, the desired conclusions follow easily from the aforementioned consequences of the locally upper Lipschitzian property of the solution set of an LCP. □

Remark. The boundedness conclusion of the sequence $\{x^k\}$ in Corollary 5.3 does not follow from Corollary 5.2. The reason is that the solution set $\mathrm{SOL}_M(q)$ is equal to the level set

$$\{x \in R^n_+ : \theta(x) \leq 0\} = \{x \in R^n_+ : \theta(x) = 0\},$$

which is different from any of the sets L_t with $t \geq 0$. Clearly, the polyhedrality nature of the LCP has much to do with the validity of Corollary 5.3.

Infinite update of c. We return to the NCP (f) and analyze the other case of the positive algorithm, namely, when $\|w_{c_k}(x^k)\| > \delta$ fails for infinitely many k. Our goal here is to demonstrate that if x^* is the limit of any subsequence $\{x^k : k \in \kappa\}$ for which

$$\|w^k\| \leq \delta \quad \text{for all } k \in \kappa,$$

and if x^* is an s-regular vector in the sense defined in [27], then x^* solves the NCP (f). For the sake of clarity, we review this concept in the definition below.

DEFINITION. *A vector* $x \geq 0$ *is said to be* s-regular *if the following system of linear inequalities has a solution in the variable* d*:*

$$\begin{aligned}
x_i + d_i &= 0 \quad \text{for } i \in I_x(x) \cup I^0_e(x), \\
f_i(x) + \nabla f_i(x)^T d &= 0 \quad \text{for } i \in I^+_f(x), \\
x_i + d_i &\geq 0 \quad \text{for } i \in I^0_f(x), \\
f_i(x) + \nabla f_i(x)^T d &\geq 0 \quad \text{for } i \in I^0_f(x), \\
x_i + d_i &\leq 0 \quad \text{for } i \in I^+_e(x), \\
f_i(x) + \nabla f_i(x)^T d &\leq 0 \quad \text{for } i \in I^+_e(x),
\end{aligned}$$

where

$$\begin{aligned}
I^+_f(x) &= \{i : f_i(x) < x_i > 0\}, \qquad I^+_e(x) = \{i : f_i(x) = x_i > 0\}, \\
I^0_f(x) &= \{i : f_i(x) < x_i = 0\}, \qquad I^0_e(x) = \{i : f_i(x) = x_i = 0\}.
\end{aligned}$$

In [27, Prop. 3], a sufficient condition for a nonnegative vector x to be s-regular was established in terms of certain matrix-theoretic properties of the Jacobian matrix

$\nabla f(x)$. We will postpone the discussion of this condition until the next subsection. In what follows, we proceed to establish a convergence result that complements Theorem 5.1. Given a subset $X \subseteq R^n$ and a point $x \in X$, we recall the set of feasible directions at x with respect to X:

$$\mathcal{F}(x, X) \equiv \{d \in R^n \mid \exists \bar{\alpha} > 0 \text{ such that } x + \alpha d \in X \text{ for all } \alpha \in [0, \bar{\alpha}]\}.$$

LEMMA 5.4. *Suppose that the penalty sequence $\{c_k\}$ is updated infinitely often and that x^* is the limit of a subsequence $\{x^k : k \in \kappa\}$ for which*

$$\|w^k\| \leq \delta \quad \text{for all } k \in \kappa.$$

Then, the sequence $\{u^k : k \in \kappa\}$, where $u^k \equiv A(x^k)H(x^k)$ has an accumulation point and any accumulation point u^∞ of this sequence satisfies

$$d^T u^\infty \geq 0 \quad \forall d \in \mathcal{F}(x^*, R^n_+). \tag{27}$$

Proof. Since $x^k \xrightarrow{k\in\kappa} x^*$ and there is only a finite number of distinct index sets $I_f(x^k)$, it is easy to see that the sequence $\{u^k : k \in \kappa\}$ has an accumulation point. Assume that u^∞ is one such accumulation point and let us show that (27) holds. Indeed, let $B \equiv \{i \mid x_i^* > 0\}$ and $N \equiv \{i \mid x_i^* = 0\}$. Noting that $\mathcal{F}(x^*, R^n_+) = \{d \in R^n \mid d_N \geq 0\}$, we conclude that (27) is equivalent to the condition

$$u_B^\infty = 0, \qquad u_N^\infty \geq 0. \tag{28}$$

We now show that (28) holds. Indeed, using the assumption that $\|w^k\| \leq \delta$ for all $k \in \kappa$ and the definition of w^k, we obtain

$$\left\| u^k + v^k - \frac{\theta(x^k)}{c_k}(x^k)^{-1} \right\| \leq \frac{\delta\theta(x^k)}{c_k} \quad \text{for all } k \in \kappa, \tag{29}$$

where

$$v^k \equiv \frac{\xi\theta(x^k)}{c_k(\theta(x^k) + e^T x^k)}(e + u^k) \quad \text{for all } k.$$

Observe that the sequence $v^k \xrightarrow{k\in\kappa} 0$ since $c_k \to \infty$. Hence, from the fact that $x_B^k \xrightarrow{k\in\kappa} x_B^* > 0$ and $c_k \to \infty$, and relation (29), we obtain that $u_B^\infty = \lim_{k\in\kappa} u_B^k = 0$. Moreover, from (29) and the fact that $x_N^k \xrightarrow{k\in\kappa} 0$, we obtain that $u_N^k + v_N^k \geq 0$ for all $k \in \kappa$ sufficiently large. Hence,

$$u_N^\infty = \lim_{k\in\kappa}(u_N^k + v_N^k) \geq 0,$$

and the result follows. □

Observe that condition (27) can be viewed as a weak stationarity condition for the point x^*. If the accumulation point x^* is nondegenerate; that is, it satisfies $x_i^* \neq f_i(x^*)$ for all $i = 1, \ldots, n$, then we can conclude that x^* is a stationary point for the function $\theta(x)$ as the following corollary states.

COROLLARY 5.5. *Let the assumptions of Lemma 5.4 hold and assume further that x^* is nondegenerate. Then, we have*

$$\theta'(x^*, d) \geq 0 \quad \text{for all } d \in \mathcal{F}(x^*, R^n_+). \tag{30}$$

Proof. Using the fact that x^* is nondegenerate, we conclude that $I_x(x^k) = I_x(x^*)$ and $I_f(x^k) = I_f(x^*)$ for all k sufficiently large. It is now straightforward to see that the sequence $\{u^k : k \in \kappa\}$ converges and that its limit point u^∞ satisfies $d^T u^\infty = \theta'(x^*, d)$ for all $d \in R^n$. The result now follows from Lemma 5.4. □

The above two results say nothing about x^* being a solution of NCP (f) or, equivalently, that x^* satisfies $\theta(x^*) = 0$. To conclude that x^* is a solution of NCP (f), we need to assume that x^* is an s-regular vector. Before showing this fact, we state a preliminary result that gives several conditions that are related to s-regularity.

LEMMA 5.6. *Let $x \in R^n_+$ be given and for all $d \in R^n$, define*

$$g_x^{\min}(d) \equiv \theta'(x,d) = \sum_{i \in I_f(x)} f_i(x) \nabla f_i(x)^T d + \sum_{i \in I_e(x)} \min\{f_i(x) \nabla f_i(x)^T d, x_i d_i\} + \sum_{i \in I_x(x)} x_i d_i,$$

$$g_x^{\max}(d) \equiv \sum_{i \in I_f(x)} f_i(x) \nabla f_i(x)^T d + \sum_{i \in I_e(x)} \max\{f_i(x) \nabla f_i(x)^T d, x_i d_i\} + \sum_{i \in I_x(x)} x_i d_i.$$

Let $g_x : \mathcal{F}(x, R^n_+) \to R$ be any homogeneous function of degree 1 (i.e., $g_x(\lambda d) = \lambda g_x(d)$, for all $\lambda > 0$ and $d \in \mathcal{F}(x, R^n_+)$) satisfying

$$g_x^{\min}(d) \le g_x(d) \le g_x^{\max}(d) \quad \text{for all } d \in \mathcal{F}(x, R^n_+),$$

and consider the following conditions on x:

(a) *x is an s-regular vector;*

(b) *there exists a $d \in \mathcal{F}(x, R^n_+)$ such that*

$$\begin{aligned} f_i(x)\left[f_i(x) + \nabla f_i(x)^T d\right] &\le 0 \quad \text{for all } i \in I_f(x), \\ \max\left\{x_i[x_i + d_i]\,,\, f_i(x)[f_i(x) + \nabla f_i(x)^T d]\right\} &\le 0 \quad \text{for all } i \in I_e(x), \\ x_i[x_i + d_i] &\le 0 \quad \text{for all } i \in I_x(x); \end{aligned} \tag{31}$$

(c) *there exist scalar $\gamma > 0$ and vector $d \in \mathcal{F}(x, R^n_+)$ such that $\gamma\theta(x) + g_x(d) \le 0$;*

(d) *if $\theta(x) > 0$ then there exists $d \in \mathcal{F}(x, R^n_+)$ such that $g_x(d) < 0$.*

Then, the following implications hold:

$$(a) \Longrightarrow (b) \Longrightarrow (c) \Longleftrightarrow (d).$$

Proof. The implication $(a) \Longrightarrow (b)$ follows from the definition of s-regularity and the fact that $d \in \mathcal{F}(x, R^n_+)$ if and only if $d_i \ge 0$ for all i such that $x_i = 0$. To show the second implication, assume that (b) holds. Summing the relations in (31) over all $i = 1, \ldots, n$, we obtain that

$$2\theta(x) + g_x^{\max}(d) \le 0,$$

which obviously implies (c) with $\gamma = 2$. The equivalence of (c) and (d) can be easily proved using the fact that g_x is homogeneous of degree 1. □

We are now ready to state the main result of this subsection.

THEOREM 5.7. *Suppose that the penalty sequence $\{c_k\}$ is updated infinitely often. If x^* is the limit of a subsequence $\{x^k : k \in \kappa\}$ for which*

$$\|w^k\| \le \delta \quad \text{for all } k \in \kappa$$

and x^* is s-regular, then $\theta(x^*) = 0$.

Proof. Assume for contradiction that $\theta(x^*) \neq 0$. Let u^∞ be an accumulation point of the sequence $\{u^k : k \in \kappa\}$ as defined in the statement of Lemma 5.4. Then, it is easily seen that the function $g_x(d) \equiv d^T u^\infty$ satisfies the assumptions of Lemma 5.6. Hence, since x^* is an s-regular vector and $\theta(x^*) \neq 0$, it follows from Lemma 5.6(d) that there exists a vector $d \in \mathcal{F}(x^*, R^n_+)$ such that $d^T u^\infty < 0$. Since this contradicts Lemma 5.4, we must have $\theta(x^*) = 0$. □

The class of s-regular functions. We may combine Theorems 5.1 and 5.7 to give a unifying convergence result for the positive algorithm. For this purpose, we define the following class of functions.

DEFINITION. *A function $f : R^n_+ \to R^n$ is said to be* s-regular *if every nonnegative vector is s-regular.*

A function f with the property that $\nabla f(x)$ is a P-matrix for all $x \geq 0$ must be s-regular; in particular, any uniform P-function f on R^n (and hence, any strongly monotone function) is s-regular. The proof of these observations hinges on the following result, which is a paraphrase of Proposition 3 in [27]. (The reader may want to consult [1] for discussion of the various matrix classes involved here.)

PROPOSITION 5.8. *Let x be a nonnegative vector. Suppose that* (i) *the principal submatrix*

$$\nabla_{I_f^+ f_{I_f^+}}(x) \tag{32}$$

is nonsingular, and (ii) *the Schur complement of this matrix in*

$$\begin{bmatrix} \nabla_{I_f^+ f_{I_f^+}}(x) & \nabla_{I_f^0 f_{I_f^+}}(x) & -\nabla_{I_e^+ f_{I_f^+}}(x) \\ \nabla_{I_f^+ f_{I_f^0}}(x) & \nabla_{I_f^0 f_{I_f^0}}(x) & -\nabla_{I_e^+ f_{I_f^0}}(x) \\ -\nabla_{I_f^+ f_{I_e^+}}(x) & -\nabla_{I_f^0 f_{I_e^+}}(x) & \nabla_{I_e^+ f_{I_e^+}}(x) \end{bmatrix} \tag{33}$$

is an S-matrix.

(Here, the index sets are all evaluated at the given x.) Then x is an s-regular vector.

We observe that if $x \geq 0$ is such that $\nabla f(x)$ is a P-matrix, then conditions (i) and (ii) of Proposition 5.8 are satisfied. Indeed, if $\nabla f(x)$ is a P-matrix then the matrices (32) and (33) are both P-matrices. In particular, (32) is nonsingular, showing that condition (i) holds. Moreover, since the Schur complement of any principal submatrix of a P-matrix is a P-matrix and since every P-matrix is an S-matrix, condition (ii) follows.

An interesting example of an s-regular function that is neither P nor monotone is the negative identity function. Indeed, if $f(x) = -x$, then $I_e^+(x) = I_f^0(x) = \emptyset$ for all $x \geq 0$. Consequently, any nonnegative vector is s-regular and f is an s-regular function. More generally, if f is a function for which these two index sets are empty for all nonnegative vectors and whose Jacobian matrix $\nabla f(x)$ is nondegenerate for all $x \geq 0$, then f must be s-regular.

The following theorem is immediate from the previous results.

THEOREM 5.9. *If f is an s-regular function, then every accumulation point of a sequence of iterates produced by the positive algorithm is a solution of the* NCP (f).

A referee of this paper correctly points out that the above theorem does not provide conditions under which a sequence of iterates produced by the positive algorithm will have at least one accumulation point. The difficulty with this deficiency of the

theorem lies in the case where the scalar c is updated infinitely often; indeed, in the present version of the algorithm, whenever c_k is updated, we do not change the iterate x^k, and essentially, do nothing. It might be necessary to modify the algorithm to yield a more desirable result.

If the function f has the property that the function θ has bounded level sets, then the only way for the sequence $\{x^k\}$ to have no accumulation point is that $\lim_{k\to\infty} \theta(x^k) = \infty$. Although we cannot rule out this possibility when $\{c_k\}$ is unbounded, it does seem rather unlikely to occur in practice. To substantiate this statement, we have implemented the positive algorithm for solving a variety of complementarity problems. The results are reported in the next section. In all the numerical tests we have conducted, the θ-values at termination were consistently substantially smaller than the θ-values at initiation, even in cases when the positive algorithm failed to solve a particular problem.

6. Numerical results. We have carried out some numerical experiments with the positive algorithm applied to two sets of complementarity problems: one experiment consists of NCPs arising from various equilibrium models that are documented in detail in [27]; the other is a set of randomly generated LCPs. For the first set of NCPs, we also compared the positive algorithm with the NE/SQP method described in [27] because the latter method was also based on the formulation (1) of the NCP and was highly successful in terms of robustness and speed. Our experiments show that the positive algorithm is faster than the NE/SQP method on the test problems but less robust. The improved speed is not surprising since in calculating each search direction, the positive algorithm solves only one system of linear equations whereas the NE/SQP method solves a convex quadratic programming with lower bound constraints; indeed, the simplicity of the direction generation is the single most important feature of the positive algorithm.

The positive algorithm was implemented in a FORTRAN-77 computer code and the experiments were conducted on a SUN SPARCStation IPX with 16 megabytes of memory and one CPU processor. Double precision arithmetic was employed in the calculations. In each iteration, we solved the system of linear equations (9) for the search direction by using the LU decomposition; the subroutines in [29] were used. We terminated the algorithm when the θ-value was less than 10^{-12}. The parameters of the algorithm were set as follows:

$$\zeta = n + 1, \quad \delta = 10^{-6}, \quad \sigma = \rho = 0.5, \quad \text{and } \alpha = 0.95.$$

We set the initial penalty parameter $c_0 = 10^3$. The numerical results for the positive algorithm applied to the equilibrium problems are summarized in Table 1.

In Table 1, n denotes the dimension of the NCP, niter the number of iterations, and aver.nls the average number of steps needed in the Armijo line search. In the column of niter, the numerators are the numbers of systems of linear equations solved by the positive algorithm, and the denominators are the numbers of quadratic programs solved by the NE/SQP method as reported in [27]. The column of θ-values gives some indication of the speed of the positive algorithm at the tail of the iterations. In these runs, all the starting points were chosen to be the vector of ones, except for the Hansen–Koopmans problem where the NE/SQP method started from $(x_1, \ldots, x_{10}, y_1, y_2, z_1, z_2) = (0.3, \ldots, 0.3, 0, 0, 0, 0)$ while the positive algorithm started from $(x_1, \ldots, x_{10}, y_1, y_2, z_1, z_2) = (0.3, \ldots, 0.3, 0.1, 0.1, 0.1, 0.1)$; the reason for this deviation is that the positive algorithm must start from a positive vector.

TABLE 1

Problem	n	niter	aver.nls	last three θ–values
Kojima–Shindo	4	8/7	2.000	8.01D−08/2.10D−12/5.05D−17
Mathieson	4	6/3	1.167	3.59D−08/8.24D−11/1.91D−13
Nash–Cournot	10	9/9	1.000	4.62D−02/2.23D−06/6.07D−15
Hansen–Koopmans	14	22/10	1.667	1.18D−08/1.85D−12/2.01D−16
Spatial price	42	21/20	1.050	1.14D−11/1.57D−12/2.16D−13
Traffic equi.	50	28/20	3.357	2.36D−09/6.10D−12/8.34D−15

The column of niter demonstrates the relative efficiency of the positive algorithm versus the NE/SQP method. Since one single system of linear equations was solved in the former algorithm, versus a quadratic program in the latter, the advantage of the positive algorithm in terms of speed should be evident. Indeed, we have compared these two algorithms on the largest of these problems, the traffic equilibrium problem, on the VAX 6000 computer at the Homewood Computing Facility Center at The Johns Hopkins University. The positive algorithm and the NE/SQP method used 3.94 and 8.35 CPU seconds, respectively. For the NE/SQP method, the FORTRAN package QPSOL [3] was used to solve each quadratic subprogram.

There are three test problems reported in [27] that were not included in the present experimentation. These are the PIES model, the Walrasian equilibrium problem with production, and the generalized von Thünen model. These problems are mixed NCPs as defined in [1]; the present version of the positive algorithm needs to be modified to deal with this class of problems. This is a topic for further study.

The set of LCPs to which the positive algorithm was applied can be classified into four types, each according to the properties of the matrix M. Specifically, the generation of M was as follows. In each case, M was completely dense.

1. $M = A^T A$. Each entry of the matrix A was generated uniformly from the interval (0,50), with a probability of 0.5 for the entry to be given a negative sign. This matrix M is symmetric positive semidefinite.

2. M is diagonally dominant. We set $M = A$, where A was generated in the same way as above except that each diagonal entry was set to be one plus the sum of the absolute values of the off-diagonal entries in the same row.

3. $M = A^2$. Here A was generated as before. This matrix M is indefinite.

4. M is a positive matrix. Each entries of M was generated uniformly from the interval (1,50).

For each matrix M generated, we constructed a solvable LCP as follows. First we generated a random number from the interval (0,1); if this number was greater than 0.5, then we set $x_i^* = 0.0$, otherwise we generated a number from the interval (0,50) and set x_i^* to be that number. Therefore, roughly half of the components of x^* were zeros. We then formed the vector q as follows. If $x_i^* > 0$, then $q_i = -M_{i,.}x^*$, where $M_{i,.}$ denotes the ith row of M; otherwise $q_i = -M_{i,.}x^* + r$, where r is a random number in (0,50) scaled by 0.3. Clearly the resulting LCP (q, M) has at least one solution, namely, x^*. For M of the first two types, x^* is the unique solution.

After each LCP was generated, we scaled the problem in the following way: let s be the sum of all entries of M and q, we scaled M and q by the factor $50/s$. The positive algorithm was then applied to the scaled LCP. It turned out that such a scaling was quite useful in ensuring the effectiveness of the algorithm.

The parameters of the algorithm were given the same values as before. We summarize the numerical results in Tables 2 and 3 for the cases $M = A^T A$ and M

TABLE 2
$M = A^TA$, $n = 100$.

Problem	1	2	3	4	5
niter	28	33	30	32	26
aver. nls	1.037	1.031	1.034	1.032	1.040
Initial θ	40308.4	35554.7	36078.4	43694.5	32757.5
Final θ	2.0D−14	3.8D−14	3.9D−14	1.5D−14	3.9D−13
Problem	6	7	8	9	10
niter	41	35	23	31	25
aver. nls	1.025	1.029	1.045	1.033	1.041
Initial θ	31538.1	35382.4	29677.3	45676.6	31909.8
Final θ	1.0D−13	4.1D−14	6.5D−13	9.2D−15	7.0D−13

TABLE 3
M is diagonally dominant, $n = 100$.

Problem	1	2	3	4	5
niter	11	9	10	10	11
aver. nls	1.300	1.125	1.222	1.444	1.300
Initial θ	46418.3	42027.2	47092.5	53837.2	36841.5
Final θ	1.6D−14	3.4D−13	2.6D−13	1.9D−14	3.1D−14
Problem	6	7	8	9	10
niter	13	10	10	10	9
aver. nls	1.583	1.333	1.111	1.111	1.125
Initial θ	45390.3	44541.4	36948.6	48601.9	43534.8
Final θ	180D−14	2.3D−13	1.0D−14	2.5D−14	8.3D−13

diagonally dominant, respectively. The entries in the tables are self-explanatory (n is the dimension of M). As we have mentioned, the matrix M is completely dense; this is the reason why we have not attempted to solve problems of larger size in these two cases; in other words, data storage has imposed a restriction on our ability to use the Sun workstation for solving larger problems of this kind.

Observe that in the above two cases, all 10 problems in each group were successfully solved; more importantly, the computational statistics were very encouraging. In contrast, the results for the remaining two cases were not as good. In each of these cases, we ran LCPs of size $n = 10, 20, 30$. Ten problems were tested in each category. When $M = A^2$, the following results were obtained. For $n = 10$, seven problems were solved to satisfaction; for $n = 20$, one; and for $n = 30$, two. When M is positive, we obtained the following results. For $n = 10$, six problems were solved to satisfaction; for $n = 20$, four; and for $n = 30$, three. Invariably, when a successful run occurred, the results were good (i.e., small number of iterations, good speed at the tail, and small number of line searches). When an unsuccessful run occurred, it was due to the excessive number of iterations (80 was the maximum we set) and the small magnitude of the search directions; for these failed runs, the θ-values were consistently in the range of 10^{-4} and 10^{-8}, which were small but not enough for successful termination according to our rule. There was good reason to believe that the iterates at termination of these unsuccessful runs were not s-regular vectors for the functions.

In summary, our computational results suggest that the positive algorithm holds promise in practice for solving complementarity problems that satisfy certain regularity conditions. For problems that do not necessarily satisfy the latter conditions, the algorithm requires further study and modification is needed.

7. Some concluding remarks. In this paper, we have presented and tested a positive algorithm for solving the general nonlinear complementarity problem. Some limiting properties of this algorithm were derived. At this time, although some theoretical issues remain with the algorithm, the computational experience we have gathered suggests that this algorithm is quite competitive with a previous algorithm on a set of equilibrium problems and has the potential for solving certain nonmonotone NCPs effectively.

REFERENCES

[1] R. COTTLE, J. S. PANG, AND R. E. STONE, *The Linear Complementarity Problem*, Academic Press, Boston, 1992.

[2] S. GABRIEL AND J. PANG, *A trust region method for constrained nonsmooth equations with applications*, in Large-Scale Optimization: State of the Art, W. Hager, D. Hearn, and P. Pardalos, eds., Academic Publishers B.V., Boston, 1994, pp. 159–186.

[3] P. GILL, W. MURRAY, M. SAUNDERS, AND M. WRIGHT, *User's guide for qpsol (version 3.2): A fortran package for quadratic programming*, Tech. Report SOL 84-6, Systems Optimization Laboratory, Dept. of Operations Research, Stanford University, Stanford, CA, 1984.

[4] O. GÜLER, *Existence of interior points and interior paths in nonlinear monotone complementarity problems*, Math. Oper. Res., 18 (1993), pp. 128–147.

[5] P. HARKER AND J. PANG, *Finite-dimensional variational inequality and complementarity problems: a survey of theory, algorithms, and applications*, Math. Programming, Series B, 48 (1990), pp. 161–220.

[6] J. JI, F. POTRA, AND S. HUANG, *A predictor-corrector method for linear complementarity problems with polynomial complexity and superlinear convergence*, Tech. Report 18, Dept. of Mathematics, The University of Iowa, Iowa City, August 1991.

[7] J. JI, F. POTRA, R. A. TAPIA, AND Y. ZHANG, *An interior-point method with polynomial complexity and superlinear convergence for linear complementarity problems*, Tech. Report TR–91–23, Dept. of Mathematical Sciences, Rice University, Houston, TX, July 1991.

[8] M. KOJIMA, N. MEGIDDO, AND T. NOMA, *Homotopy continuation methods for nonlinear complementarity problems*, Math. Oper. Res., 16 (1991), pp. 754–774.

[9] M. KOJIMA, N. MEGIDDO, T. NOMA, AND A. YOSHISE, *A unified approach to interior point algorithms for linear complementarity problems*, Vol. 538, Lecture Notes in Computer Science, Springer-Verlag, Berlin, 1991.

[10] M. KOJIMA, N. MEGIDDO, AND Y. YE, *An interior point potential reduction algorithm for the linear complementarity problem*, Math. Programming, 54 (1992), pp. 267–279.

[11] M. KOJIMA, S. MIZUNO, AND T. NOMA, *A new continuation method for complementarity problems with uniform P-functions*, Math. Programming, 43 (1989), pp. 107–113.

[12] ———, *Limiting behavior of trajectories by a continuation method for monotone complementarity problems*, Math. Oper. Res., 15 (1990), pp. 662–675.

[13] 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.

[14] 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.

[15] M. KOJIMA, T. NOMA, AND A. YOSHISE, *Global convergence and detecting infeasibility in interior-point algorithms*, Research Reports on Information Sciences, Ser. B: Operations Research B-257, Dept. of Information Sciences, Tokyo Institute of Technology, Oh-Okayama, Meguro-ku, Tokyo, September 1992.

[16] I. J. LUSTIG, *Feasibility issues in a primal-dual interior point method for linear programming*, Math. Programming, 49 (1990/91), pp. 145–162.

[17] 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.

[18] L. MCLINDEN, *The complementarity problem for maximal monotone multifunctions*, in Variational Inequalities and Complementarity Problems, R. Cottle, F. Giannessi, and J.-L. Lions, eds., Wiley, New York, 1980, pp. 251–270.

[19] S. MIZUNO, *A new polynomial time method for a linear complementarity problem*, Math. Programming, 56 (1992), pp. 31–43.

[20] S. MIZUNO AND M. J. TODD, *An $O(n^3L)$ adaptive path following algorithm for a linear complementarity problem*, Math. Programming, 52 (1991), pp. 587–595.

[21] R. D. C. MONTEIRO, *The global convergence of a class of primal potential reduction algorithms for convex programming*, Tech. Report 91–024, Dept. of Systems and Industrial Engineering, University of Arizona, Tucson, August 1991.

[22] ———, *A globally convergent primal dual interior point algorithm for convex programming*, Tech. Report 91–021, Dept. of Systems and Industrial Engineering, University of Arizona, Tucson, July 1991.

[23] R. D. C. MONTEIRO AND S. WRIGHT, *Superlinear primal-dual affine scaling algorithms for* LCP, Tech. Report 93–9, Dept. of Systems and Industrial Engineering, University of Arizona, Tucson, June 1993.

[24] J. S. PANG, *Newton's method for B-differentiable equations*, Math. Oper. Res., 15 (1990), pp. 311–341.

[25] ———, *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.

[26] ———, *Iterative descent algorithms for a row sufficient linear complementarity problem*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 611–624.

[27] J. S. PANG AND S. A. GABRIEL, NE/SQP: *a robust algorithm for the nonlinear complementarity problem*, Math. Programming, 60 (1993), pp. 295–338.

[28] F. A. POTRA AND Y. YE, *Interior point methods for nonlinear complementarity problems*, Working Paper, Dept. of Management Sciences, University of Iowa, Iowa City, July 1991.

[29] W. H. PRESS, B. P. FLANNERY, S. A. TEUKOLSKY, AND W. T. VETTERLING, *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, Cambridge, NY, 1989.

[30] M. J. TODD AND Y. YE, *A centered projective algorithm for linear programming*, Math. Oper. Res., 15 (1990), pp. 508–529.

[31] S. J. WRIGHT, *An infeasible interior point algorithm for linear complementarity problems*, Tech. Report MCS–P331–1092, Mathematical and Computer Science Division, Argonne National Laboratory, Argonne, IL, October 1992.

[32] Y. YE AND K. ANSTREICHER, *On quadratic and* $O(\sqrt{n}L)$ *convergence of a predictor-corrector algorithm for* LCP, Math. Programming, 62 (1993), pp. 537–551.

[33] Y. YE AND P. M. PARDALOS, *A class of linear complementarity problems solvable in polynomial time*, Linear Algebra Appl., 152 (1991), pp. 3–17.

[34] Y. ZHANG, *On the convergence of a class of infeasible interior–point methods for the horizontal linear complementarity problem*, SIAM J. Optim., 4 (1994), pp. 208–227.

[35] 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. OPTIMIZATION
Vol. 5, No. 1, pp. 149–171, February 1995

008

A PRACTICAL INTERIOR-POINT METHOD FOR CONVEX PROGRAMMING*

FLORIAN JARRE† AND MICHAEL A. SAUNDERS‡

Abstract. The authors present a primal interior-point algorithm for solving convex programs with nonlinear constraints. The algorithm uses a predictor-corrector strategy to follow a smooth path that leads from a given starting point to an optimal solution. A convergence analysis is given showing that under mild assumptions the algorithm simultaneously iterates towards feasibility and optimality. The matrices involved can be kept sparse if the nonlinear functions are separable or depend on only a few variables.

A preliminary implementation has been developed. Some promising numerical results indicate that the algorithm may be efficient in practice, and that it can deal in a single phase with infeasible starting points without relying on some "big M" parameter.

Key words. convex program, interior-point method, implementation, sparsity

AMS subject classifications. 65F05, 65F10, 65F50, 65K05, 90C05

1. Introduction.

Soon after Karmarkar's proof of polynomial-time complexity of an interior-point method for solving linear programs [15], many related methods for linear programs were presented and analyzed. Several numerical implementations were also developed and shown to be efficient on large real-life problems (e.g., [19], [20], [23]).

Concurrently, Karmarkar's method was modified and applied to nonlinear convex programs by Sonnevend [35], and detailed complexity analyses were given in [8], [10], [11], [21], [25], [29], [30], showing that for certain classes of nonlinear constraints essentially the same speed of convergence can be expected as for a linear program. However, the conversion of these theoretical results into numerical algorithms has been very slow so far.

In this paper, we first outline a predictor-corrector method and analyze its convergence under mild conditions. In particular, we do not assume that there exists an optimal solution or that the Karush–Kuhn–Tucker (KKT) conditions hold at an optimal solution. We then consider practical aspects of this method. Our main aim is to provide some assurance, supported by numerical experiments, that interior-point methods are in practice—not just in theory—an efficient tool for solving certain classes of convex (and possibly nonconvex) problems.

This claim regarding practical efficiency stands in contrast to the experience obtained from implementations of the (closely related) sequential unconstrained minimization technique (SUMT) of Fiacco and McCormick [2] in the 1960s. We therefore outline some of the new theoretical developments that may be used to stabilize the performance of interior-point methods. Our main argument is that a careful application of the *theoretical* results also gives rise to a reevaluation of the *practical* relevance of interior-point methods for solving nonlinear (convex) programs.

* Received by the editors July 31, 1991; accepted for publication (in revised form) August 5, 1993.

† Institut für Angewandte Mathematik, University of Würzburg, 8700 Würzburg, Germany (jarre@vax.rz.uni-wuerzburg.d400.de). This work was supported by a research grant from the Deutsche Forschungsgemeinschaft.

‡ Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, California 94305-4022 (mike@sol-michael.stanford.edu). This work was partially supported by Department of Energy grant DE-FG03-92ER25117, National Science Foundation grant DDM-9204208, and Office of Naval Research grant N00014-90-J-1242.

1.1. The problem and assumptions. The problem under study is to find an optimal solution x^* for the convex program

$$\text{(CP)} \qquad \min\{ f_0(x) \mid x \in P \},$$

where the feasible domain P is given by

$$P := \{ x \in \mathcal{A} \mid f_i(x) \leq 0,\ 1 \leq i \leq m \}, \tag{1}$$

$$\mathcal{A} := \{ x \in \mathbb{R}^n \mid Ax = b \}, \tag{2}$$

where the matrix $A \in \mathbb{R}^{m_2 \times n}$ and the vector $b \in \mathbb{R}^{m_2}$ are constant data. The distinction between A and $\mathcal{A}$ will be clear from the context. The following assumptions are made.

Assumption 1. The functions $f_i : S \to \mathbb{R}$ $(0 \leq i \leq m)$ are continuous and convex on a common closed set $S \supset P$ and are twice continuously differentiable in the interior S^o of S.

Assumption 2. The first and second derivatives are known. We denote them by column vectors and square matrices

$$g_i(x) = \nabla f_i(x), \qquad H_i(x) = \nabla^2 f_i(x).$$

Assumption 3. The matrix $\sum H_i(x) + g_i(x) g_i(x)^T$ is positive definite for all $x \in S^o$. (This assumption is for convenience; it may be relaxed to assuming positive definiteness on the null space of A.)

Assumption 4. We are given a starting point $x^0 \in S^o$. We do not assume that x^0 is feasible (i.e., we do not require $x^0 \in P$), and we allow P to be empty or unbounded. We also allow the objective to be unbounded for $x \in P$.

Assumption 5. Nonnegative quantities β_i are known such that $f_i(x^0) < \beta_i$ for all i and such that S contains the "enlarged feasible set" $\{ x \mid f_i(x) \leq \beta_i \}$.

2. A simple barrier method. We start by presenting a simplified version of our algorithm where we assume that the relative interior of the feasible set P is nonempty and bounded. (By continuity of $f_0(x)$, this means that the objective is also bounded for $x \in P$.) These assumptions are dropped in §3 and later.

2.1. Outline of a predictor-corrector method. The principle of a barrier method for approximating the solution of problem (CP) is based on the ideas in [2] and [3] and can be outlined as follows:

For $\mu = \mu^0,\ \mu^1,\ \mu^2,\ \ldots$ (where $\mu^0 > \mu^1 > \mu^2 \ldots$ and $\mu^k \to 0$), find

$$x(\mu) := \arg\min_{x \in \mathcal{A}} f_0(x) + \mu\phi(x),$$

i.e., minimize the true objective function f_0 perturbed by $\mu\phi$, where $\phi(x)$ is a smooth convex barrier function for the set P tending to infinity as x approaches the boundary of P and being finite in P^o.

Here, and in the remainder of this paper, we refer to the *relative* interior of the feasible set P (relative to the affine manifold $\mathcal{A}$) and to the boundary of the *relative* interior. The boundary is therefore given by those points $x \in P$ for which $f_i(x) = 0$ for at least one i.

Throughout, we use the logarithmic barrier function

$$\phi(x) = -\sum_{i=1}^{m} \log(-f_i(x)). \tag{3}$$

It is straightforward to see that ϕ is smooth and convex if the f_i are also.

It is well known (see, e.g., [2]) that the minimizers $x(\mu)$ are unique if, for example, P is bounded, ϕ is strictly convex, and f_0 is convex. For any $\mu > 0$ the barrier term $\mu\phi(x)$ ensures that $x(\mu)$ is feasible for (CP). Fiacco and McCormick showed under weak assumptions that the minimizers $x(\mu)$ form a smooth curve that leads to an optimal solution of (CP) as the perturbation $\mu\phi(x)$ of the objective function is "phased out," i.e., as $\mu \to 0$.

A general outline for a predictor-corrector barrier method can be stated as follows. We assume $\mu^0 > 0$ and $x(\mu^0)$ are given.

Set $k = 0$.
Do until convergence
- Compute the tangent $x'(\mu^k)$.
- Select $\mu^{k+1} < \mu^k$.
- (Predictor step) Estimate $x(\mu^{k+1})$ by linear extrapolation:

$$\hat{x}(\mu^{k+1}) := x(\mu^k) + (\mu^{k+1} - \mu^k)x'(\mu^k).$$

- (Corrector step) Estimate $x(\mu^{k+1})$ by approximately minimizing $f_0(x) + \mu^{k+1}\phi(x)$ in $\mathcal{A}$, using Newton's method with starting point $\hat{x}(\mu^{k+1})$.
- Set $k = k + 1$.

End

In this outline we suppressed a few details that we mention briefly, postponing fuller discussion to §3.

1. We must choose μ^{k+1} such that the prediction $\hat{x}(\mu^{k+1})$ is guaranteed to be feasible.

2. Newton's method must be secured by a linesearch (since the function ϕ that defines $x(\mu)$ is not defined outside P).

3. We must find an initial point $x(\mu^0)$ minimizing $f_0(x) + \mu^0\phi(x)$ in $\mathcal{A}$.

2.2. Theoretical results. Under certain conditions, the strong theoretical results that were proved for interior-point methods for linear programs can be extended to interior-point methods for nonlinear convex programs; see, e.g., [10], [11], [17], [21], [29], [30], [35], [36]. The most general framework for convex optimization via interior-point methods is presented in [30]. We reproduce some of the results in [30], since they explain important features of our method and clarify its relationship to the SUMT method of Fiacco and McCormick [2]. The results in [30] hold for programs (CP) with *self-concordant* barrier functions.

Let the logarithmic barrier function $\phi(x)$ for problem (1) be defined as in (3) and let $\psi : \mathbb{R} \to \mathbb{R}$ satisfy $\psi(\beta) := \phi(x + \beta h)$. Thus, ψ depends on two given parameters $x \in P^o$ and $h \in \mathbb{R}^n$.

DEFINITION. *The function ϕ is self-concordant if ψ satisfies*

$$\psi'''(0) \leq 2\psi''(0)^{3/2} \tag{4}$$

for all strictly feasible x and all $h \in \mathbb{R}^n$.

This condition is satisfied for example if the functions f_i are linear or quadratic, or if $f_i(X) = -\det X$ and X is a positive definite matrix. Condition (4) bounds the *relative* change of $\nabla^2\phi(x)$ in a small neighborhood of x (since ψ''', which describes the (absolute) change of ψ'', is bounded by a suitable power of ψ'').

Self-concordance of the logarithmic barrier function then implies the following results (see [30], [12]).

1. (Inner ellipsoid for P) The canonical norm associated with a point $x \in P^o$ is given by $\|h\|_H = (h^T H h)^{1/2}$, where $H = \nabla^2\phi(x)$ and $h \in \mathbb{R}^n$. The unit ball of the H-norm describes an inner ellipsoid for P: If $Ah = 0$ and $\|h\|_H \leq 1$ then $x + h \in P$.

2. (Outer ellipsoid for P around $\bar{x}$) If $\bar{x} = \arg\min \phi(x)$ and $\|h\|_H > 1 + 3m$ then $\bar{x} + h \notin P$. Here, m is the number of convex constraints for P. The point $\bar{x}$ is the analytic center of P [35].

3. (Newton's method) Let Δx be the Newton step for minimizing ϕ, so that

$$\begin{pmatrix} \nabla^2\phi(x) & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ y \end{pmatrix} = -\begin{pmatrix} \nabla\phi(x) \\ 0 \end{pmatrix}$$

for some multiplier $y \in \mathbb{R}^{m_2}$. For any $x \in P^o$ the H-norm of the Newton step satisfies $\|\Delta x\|_H \leq \sqrt{m}$ (the square root of the number of constraints). If $\|\Delta x\|_H \leq 1$ then ϕ has a minimum (i.e., P is bounded), and if $\|\Delta x\|_H \leq 0.2$ then Newton's method for minimizing ϕ is quadratically convergent with constant at most 2. More precisely, if $\Delta\tilde{x}$ is the (following) Newton step starting at $x + \Delta x$ and $\tilde{H}$ is the Hessian of ϕ at $x + \Delta x$, then $\|\Delta\tilde{x}\|_{\tilde{H}} \leq 2\|\Delta x\|_H^2$.

4. (Distance to optimality) Finally, $f_0(x(\mu)) - f_0(x^*) \leq m\mu$. (This result also holds for nonself-concordant barrier functions, merely using convexity of f_i, $0 \leq i \leq m$; see Appendix A.)

With the above results one can show that for a linearly decreasing sequence $\{\mu^k\}$, one step of Newton's method suffices to approximate the next center to sufficient accuracy. More precisely, assume $x(\mu^0)$ is given. If $\mu^{k+1} = (1 - 1/8\sqrt{m})\mu^k$, the iterates x^k remain in a close neighborhood of the points $x(\mu^k)$, and the method is linearly convergent.

This result is somewhat surprising, since as $\mu \to 0$ the iterates $x(\mu)$ approach the optimal solution x^*. In general, this lies at the boundary of P, and the logarithmic barrier function has a singularity at the boundary (it approaches infinity). Moreover, we will see that the Hessian of ϕ is usually rank deficient in the limit. Nevertheless (assuming exact arithmetic) the subproblems of approximating $x(\mu^{k+1})$ are all of the "same difficulty" in that they all require just one Newton step.

The latter property depends on the particular choice of the barrier function (the logarithmic barrier function) and in general does not hold for other barrier functions (see also [8]). However, the stepsize $(1 - 1/8\sqrt{m})$ to guarantee the property is much too small to be useful in an implementation. Of practical interest are the results for the norms $\|\,.\,\|_H$, which are used below.

3. Modifications. We now drop the temporary assumption of the last section and allow P to be empty or unbounded, and the objective to be unbounded for $x \in P$. The analytic center then no longer exists, and it is necessary to redefine the points $x(\mu)$ below. For convenience we assume that $\mathcal{A} = \mathbb{R}^n$, i.e., m_2 of §1.1 is zero. The results also hold for $\mathcal{A} \subsetneq \mathbb{R}^n$; the necessary modifications are straightforward but tedious.

3.1. Shifted constraints $f_i(x, \mu)$. A given initial point $x^0 \in S^o$ might not be feasible for (CP). Moreover, the (relative) interior of (CP) may be empty, as is occasionally the case for (poorly formulated) linear programs. In both cases, the barrier approach presented in the last section is not possible. To define a barrier function at x^0 we "enlarge" the feasible set P by subtracting certain nonnegative quantities β_i from f_i such that $f_i(x^0) - \beta_i < 0$, i.e., such that x^0 is in the relative interior of the "enlarged" feasible domain $\{\, x \in \mathcal{A} \mid f_i(x) - \beta_i \leq 0 \,\}$. Both the

"enlargement" of the feasible set and the "weight" of the barrier function (relative to the objective function f_0) will be parameterized by the same parameter μ. Because of this double role of μ—as a parameter for optimality and for infeasibility—it is convenient to *restrict* μ to the interval $(0, 1]$, and to fix $\mu^0 = 1$. To eliminate again the restriction on the initial weight of the objective function, we introduce another parameter ρ below.

Define $t = \max_{1 \le i \le m}\{1, f_i(x^0)\}$, $\beta_i = \max\{f_i(x^0) + t, 0\}$ for $1 \le i \le m$, and

$$f_i(x, \mu) = f_i(x) - \mu\beta_i. \tag{5}$$

The number 1 in the definition of t is arbitrary; it may be replaced by some other number; see §5.2. Note that $f_i(x, \mu)$ depends implicitly on β_i, and hence on the initial point. However, to keep the notation short we do not list β_i as an extra parameter of f_i. The shift implies that

$$f_i(x^0, 1) \le -t, \qquad 1 \le i \le m,$$

so that for $\mu^0 = 1$ the initial point x^0 is at least $t \ge 1$ away from each constraint.

The above computation of β_i is not affine invariant. Neither is it invariant under multiplication of f_i by a positive constant. To reduce the latter dependence we suggest multiplying f_i by

$$\frac{1}{\|g_i(x^0)\|_2 + 1} \quad \text{or} \quad \frac{1}{\|g_i(x^0)\|_2 + \|H_i(x^0)\|}$$

(with some matrix norm $\|\,.\,\|$) as the very first step of the algorithm. Note that $\nabla_x f_i(x, \mu) = g_i(x)$.

3.2. Shifted sets P_μ. For $\mu \in [0, 1]$ we consider feasible domains P_μ defined by

$$P_\mu := \{\, x \in \mathcal{A} \mid f_i(x, \mu) \le 0, \quad 1 \le i \le m \,\}. \tag{6}$$

Note that $P_0 = P$ and that x^0 is in the interior of P_1 (i.e., $x^0 \in P_1^o$). The algorithm below follows a path of points $x(\mu) \in P_\mu$ from $\mu = 1$ to $\mu = 0^+$. In contrast to the simple outline given earlier, the feasible sets P_μ for the subproblems of finding $x(\mu)$ will not remain constant. For the sets P_μ we define a barrier function ϕ of the variables x and the parameter μ by

$$\phi(x, \mu) = -\sum_{i=1}^{m} \log(-f_i(x, \mu)).$$

The change in concept by considering a shifted barrier function $\phi(x, \mu)$ is not substantial, and in some cases the theoretical results for $\phi(x)$ that were outlined in §2.2 also hold for $\phi(x, \mu)$ with $\mu \in [0, 1]$ fixed. For example, if the functions $f_i(x)$ are linear or convex quadratic, then so are the functions $f_i(x, \mu)$, and the self-concordance properties of the logarithmic barrier function also hold for $\phi(x, \mu)$. (More general "compatible" functions, as defined and analyzed in [30], also allow a shift of the above form.) We point out that for other convex functions this is not always true and if the domain of the constraint functions is not all of $\mathbb{R}^n$ we need to use special care in applying such a shift β. A certain function f_i may be convex (or have a self-concordant logarithmic barrier function) in the domain $\{\, x \mid f_i(x) \le 0 \,\}$ but not in $\{\, x \mid f_i(x) \le \beta_i \,\}$ if $\beta_i > 0$. In particular, for constraints that have a singularity

at their boundary, the shifts β_i should be kept at zero. The following brief example illustrates this situation.

Consider the pair of convex inequalities $y - \sqrt{x} \leq 0$, $-y \leq 0$ with initial point $(1,1)$. In this case, the constraints $y - \sqrt{x} \leq 1$, $-y \leq 0$ shifted as above do not properly define the enlarged set; the point $x = 0$, $y = 0.5$ lies at the boundary of the domain of a constraint function, but none of the constraints is active. Henceforth, we assume that the shifts β_i satisfy our assumption in §1.1.

The following lemma relates the feasible domains P_μ to P.

LEMMA 3.1. *The following relations hold*:

1. $P \subset P_{\mu^1} \subset P_{\mu^2}$ *for* $0 \leq \mu^1 \leq \mu^2 \leq 1$;
2. $P = \cap_{\mu>0} P_\mu$;
3. *If* P *is not empty, the interior* P_μ^o *is not empty for all* $\mu \in (0,1]$;
4. *If* P *is empty, there is a* δ *in* $[0,1)$ *such that* P_μ *is empty for all* $\mu \in [0,\delta)$ *and the interior* P_μ^o *is not empty for all* $\mu \in (\delta,1]$.

Proof. The proof is straightforward. □

For the case of linear constraints, much stronger results hold than those presented here; see, e.g., [4]. In an analysis by Gill et al.[7] a different shift strategy is examined, and it is shown that an interior-point algorithm based on a shifted barrier function is superlinearly convergent if the objective and the constraints are linear. In this paper, however, the shifts are merely used to define a barrier function at the initial point.

A slightly more complicated construction to achieve "feasibility" of the initial point is as follows. If points $\bar{x}^i$ are known such that $f_i(\bar{x}^i) < 0$, we may consider the functions $f_i(x,\mu) := f_i(x + \mu(\bar{x}^i - x^0))$. In this case, Lemma 3.1, Parts 1 and 2 no longer hold but the sets P_μ still converge to P and Parts 3 and 4 still hold. We will not discuss this modification further but concentrate on the case (5).

3.3. The perturbed center. Section 3.2 dealt with a shift of the constraints such that the initial point becomes feasible. In this section we aim at making the initial point a "center" of the shifted set by introducing a perturbation to the barrier function.

Note that the set of linear perturbations of ϕ,

$$\{\, \tilde{\phi} \mid \tilde{\phi}(x,\mu) := \phi(x,\mu) + w^T x,\ w \in \mathbb{R}^n \text{ fixed} \,\},$$

forms a family of strictly convex barrier functions for P_μ. Under mild conditions (see Lemma 3.2 below) each barrier function defines a smooth path of minimizers $\arg\min\{f_0(x) + \mu\tilde{\phi}(x,\mu)\}$ leading from some point in P_1^0 to an optimal solution of (CP). Also, for any $\mu^0 > 0$ and any $x^0 \in P_{\mu^0}^o$ there is a unique w such that the path starts at x^0 when μ runs from μ^0 to 0. Therefore, the functions $\tilde{\phi}$ define a vector field that flows to the optimal set, a fact that is used extensively later on and is well described for the case of linear constraints in [22]. The minimizers of the perturbed barrier functions are called *perturbed centers*, and the paths of the vector field are referred to as perturbed center paths.

The "perturbed center" without perturbation (i.e., with $w = 0$) is the *analytic center* introduced above. It exists if, for example, the set of optimal solutions is bounded and ϕ is self-concordant. For the analytic center a number of nice properties can be shown; in particular are the following.

1. The two-sided ellipsoidal approximation of the set P around its analytic center $\bar{x}$ carries over to the level sets $P_{\mu,\lambda} := \{\, x \in P_\mu \mid f_0(x) \leq \lambda \,\}$ centered at the point $x(\mu)$. Here, $\lambda := f_0(x(\mu)) + \mu$. The sub level set $P_{\mu,\lambda}$ has the barrier function

$-\log(\lambda - f_0(x)) - \sum \log(-f_i(x,\mu))$. Its derivative is zero at the point $\bar{x} = x(\mu)$, and hence the ellipsoidal approximation holds with a similarity ratio of at most $1+3(m+1)$.

2. The numbers $\mu/f_i(x(\mu),\mu)$ define dual feasible variables that can be used in a test for optimality (see also Appendix A.2).

Both properties hold in a somewhat weaker form if the perturbation w is small [11]. We emphasize, however, that the name perturbed "center" is somewhat misleading if the perturbation w is large. In this case, the perturbed center may be arbitrarily close to the boundary, and the name "center" is no longer justified. As we see below, our algorithm may also become inefficient when it follows a perturbed center curve that is too close to the boundary. Unfortunately, no reliable norm is known to determine when w is large, and in §5.1 we can only give a sometimes overly pessimistic criterion for judging when w may be considered sufficiently small.

For our barrier method we consider the functions $\varphi_\mu : P_\mu^o \to \mathbb{R}$,

$$\varphi_\mu(x) := \frac{\rho}{\mu} f_0(x) - \sum_{i=1}^{m} \log(-f_i(x,\mu)) - w^T x, \tag{7}$$

that combine a multiple of the objective function and the perturbed barrier function. For $\mu \in (0,1]$ we define $x(\mu)$ as follows:

$$x(\mu) \text{ is a } \textit{perturbed center} \text{ if } x(\mu) \in P_\mu^o \text{ and if it is a minimum of } \varphi_\mu.$$

(If P is empty then the definition is valid only for $\mu \in (\delta, 1]$, where δ is as in Lemma 3.1.) We note that for certain degenerate cases (e.g., when the set of optimal solutions is unbounded) the analytic center may not exist, while the perturbed center is well-defined. The gradient and Hessian of φ_μ are denoted by $g(x,\mu)$ and $H(x,\mu)$ as follows:

$$g(x,\mu) := \nabla\varphi_\mu(x) \;\; = \frac{\rho}{\mu} g_0(x) + \sum_{i=1}^{m} \frac{g_i(x)}{-f_i(x,\mu)} - w, \tag{8}$$

$$H(x,\mu) := \nabla^2\varphi_\mu(x) = \frac{\rho}{\mu} H_0(x) + \sum_{i=1}^{m} \frac{H_i(x)}{-f_i(x,\mu)} + \frac{g_i(x) g_i(x)^T}{f_i(x,\mu)^2}. \tag{9}$$

Clearly, $H(x,\mu)$ is positive semidefinite if the f_i are convex. From our assumptions it follows that H is positive definite. Note that $x = x(\mu)$ is a perturbed center if and only if it is a zero of the following characteristic equation:

$$g(x,\mu) = 0. \tag{10}$$

(If x satisfies $g(x,\mu) = 0$, then by convexity it is a minimum of φ_μ. Conversely, if x is a minimum of φ_μ in the open set P_μ^o, then $g(x,\mu) = 0$.)

The particular perturbation we choose is

$$w := \rho g_0(x^0) + \sum_{i=1}^{m} \frac{1}{-f_i(x^0,1)} g_i(x^0), \tag{11}$$

where $\rho > 0$ determines the initial weight of the objective function. In theory, any value of $\rho > 0$ is possible to obtain convergence; a practical choice is described in §5.3. (Both w and ρ are fixed throughout the algorithm.) Thus, by definition of w,

the point x^0 is the first center: $x^0 = x(1)$. It was our goal to define a path for all problems that have an optimal solution, with the path leading to an optimal solution. However, if the problem is solvable but "highly degenerate" (in that it has neither an interior nor a bounded set of optimal solutions), then the perturbed center might not exist as the example following Lemma 8.1 shows. For less degenerate cases we prove the following lemma.

LEMMA 3.2. *Let* (CP) *have an optimal solution. The following statements hold if the interior P^o of P is not empty, or if the set of optimal solutions is bounded.*

1. *A unique perturbed center exists for all $\mu \in (0, 1]$.*
2. *The perturbed center $x(\mu)$ is bounded for all $\mu \in (0, 1]$.*
3. *It becomes feasible in the limit:* $\lim_{\mu \to 0} \min_{x \in P} \|x(\mu) - x\|_2 = 0$.
4. $\lim_{\mu \to 0} f_0(x(\mu)) - f_0(x^*) = 0$ *if x^* is an optimal solution to* (CP).

Proof. See Appendix B. □

We note that the analytic center exists only if P^o is nonempty *and* the set of optimal solutions is bounded. Our assumptions for Lemma 3.2 are weak in the sense that they do not require a constraint qualification. The centers $x(\mu)$ still converge to an optimal solution x^* (under our assumptions) even if the KKT conditions do not hold at x^*. To our knowledge, neither the limit of the perturbed center nor the limit of the analytic center of shifted sets P_μ has been analyzed so far for nonlinear convex constraints.

The function $\mu\varphi_\mu(x)$ is (at least for $\rho = 1$) just the objective function $f_0(x)$ to which a multiple (μ) of the barrier function is added, as in the outline of §2.1. Our choice of φ_μ in (7), rather than the seemingly more natural choice $\mu\varphi_\mu$, was more convenient since the property of self-concordance is not invariant under multiplication of the barrier function by μ.

4. A modified barrier method. The general idea of the method is as follows. Starting from $\mu = 1$ and $x(1) = x^0$, a sequence of iterates is generated in some neighborhood of the path of perturbed centers $x(\mu)$. The iterates x^k are regarded as approximations to points $x(\mu^k)$, where $\mu^0 = 1$, $\mu^k > 0$, $\mu^k \to 0$. The algorithm proceeds in three steps per iteration.

Step 1. Compute the tangent x' to the perturbed curve passing through the current iterate x^k at $\mu = \mu^k$.

Step 2. Choose adaptively a steplength $\alpha \in (0, 1)$ to follow the tangent starting from x^k. Let the resulting point $\hat{x}^k = x^k - \alpha\mu^k x'$ be a prediction for $x(\mu^{k+1})$. Set $\mu^{k+1} = \mu^k(1 - \alpha)$. The steplength α is chosen such that $\hat{x}^k \in P^o_{\mu^{k+1}}$ and such that Newton iterations starting from $\hat{x}^k$ for finding $x(\mu^{k+1})$ can be expected to converge rapidly.

Step 3. Perform a small number of Newton steps with linesearch to bring the iterate closer to the path of perturbed centers. The result of this "corrector step" is x^{k+1}.

It is in Step 3 where our method differs from most implementations of interior-point algorithms for linear programming [6], [19], [20]. For primal-dual methods for solving linear programs it appears that the extra effort taken in Step 3 to move away from the boundary towards the center does not pay [34]. For nonlinear problems our results indicate that "centering" stabilizes the algorithm. Furthermore, since our algorithm works in primal space only, the stopping test is reliable only in a neighborhood of the analytic center $x(\mu)$.

4.1. The tangent. Let $v(x,\mu)$ be the derivative of $g(x,\mu)$ with respect to μ:

$$v(x,\mu) := \frac{d}{d\mu} g(x,\mu) = -\frac{\rho}{\mu^2} g_0(x) - \sum_{i=1}^{m} \frac{\beta_i}{f_i(x,\mu)^2} g_i(x). \tag{12}$$

If $H(x(\mu),\mu)$ is positive definite, the perturbed center is unique and the tangent to the curve of perturbed centers at $x(\mu)$ is defined by the linear system

$$H(x(\mu),\mu)x'(\mu) = -v(x(\mu),\mu). \tag{13}$$

Verification of (13) is straightforward by differentiating $g(x(\mu),\mu) \equiv 0$ in (8) with respect to μ. In general (if there is at least one active constraint with a nonzero Lagrange multiplier at the optimal solution) it holds that $\lim_{\mu\to 0} \mu^2 H(x(\mu),\mu)$ exists and is nonzero. In our implementation we therefore use $\mu^2 H(x,\mu)$ and $\mu^2 v(x,\mu)$ instead of the unbounded quantities $H(x,\mu)$ and $v(x,\mu)$.

Note that w does not occur in the definition of the tangent. If the current iterate is some point x^k that is not on the path of perturbed centers, then the above quantity is the tangent to some other perturbed center curve that also leads to an optimal solution x^*.

In Step 1 above we determine x' from (13) with x^k in place of $x(\mu)$ and $\mu = \mu^k$, i.e. from the system $H(x^k,\mu^k)x' = -v(x^k,\mu^k)$. The steplength α in Step 2 depends on how well Newton's method converges. We focus on the Newton step first.

4.2. The Newton step. The Newton step Δx for finding $x(\mu)$ starting from $x \in P_\mu$ is given by the system

$$H(x,\mu)\Delta x = -g(x,\mu). \tag{14}$$

From [30] (see §2.2) we know that Newton's method (without linesearch) for finding the center $x(\mu)$ converges quadratically if φ_μ is self-concordant and if

$$\gamma := (\Delta x^T H(x,\mu)\Delta x)^{1/2} = (-g(x,\mu)^T \Delta x)^{1/2} = \|\Delta x\|_H < 0.2. \tag{15}$$

In our program we used $\gamma < 0.5$ as a stopping test for Newton's method. This choice of γ guarantees that we are reasonably close to the center in the sense that if we continue iterating Newton's method without linesearch it is guaranteed to converge. (Compare also with §4.5.) Unfortunately, for $\gamma \approx 1$ or $\gamma > 1$ the H-norm of the Newton step does not contain any information about the closeness of the iterate to the boundary of P_μ, since it is easy to construct examples for which $\gamma \to 1$ while x approaches the boundary of P_μ, or conversely for which $\gamma = O(\sqrt{m})$ even if P_μ is a ball and x is the center of this ball.

We note that a Newton step for finding $x(\mu^{k+1})$ may not be necessary for convergence, since as mentioned above, all the "perturbed center curves" end in the optimal set. Hence one could continue by following the tangents of different curves. However, the step along the tangent may bring the point $\hat{x}^k$ close to the boundary of P_{μ^k} (and iterating too close to the boundary of P_μ slows down convergence; see §7.1), so that a Newton correction is indeed useful. (The proofs that methods without any centering converge to an optimal solution are quite involved, even in the case of linear constraints; see, e.g., [37] and others. We will not discuss this issue here.) In our program we compute an inexact Newton correction by using the same factorization of H as already used for the computation of x'. The linesearch during the Newton step is controlled by the merit function $\varphi_{\mu^{k+1}}(x)$.

4.3. The steplength α during extrapolation. Our goal is to choose $\alpha \in (0,1)$ as large as possible such that the extrapolation $\hat{x}^k = x^k - \alpha\mu^k x'$ is strictly feasible, i.e., $\hat{x}^k \in P^o_{\mu^{k+1}}$ with $\mu^{k+1} = \mu^k(1-\alpha)$, and such that Newton's method for finding the (next) center converges rapidly.

An obvious possibility to guarantee the second condition is to choose the steplength during extrapolation small enough so that the first Newton step Δx starting at the predicted point satisfies a relation of the type (15). However, this generally results in very short steps α. In our implementation we constructed a "trust region" for the choice of α and first approximated the maximum possible steplength $\alpha_{\max}$ such that $x + \alpha_{\max}x'$ is still feasible and then took r percent of $\alpha_{\max}$. If it turned out that Newton iteration converged quickly we increased r for the next extrapolation and, conversely, if Newton iteration was slow we decreased r. More precisely, we initialized $r = 0.7$ for example, and if Newton's method took more than, say, four iterations we set $r = \max\{r/2,\ 2r-2\}$; if it took less than three iterations we set $r = (r+1)/2$. We allowed a rather large number of (inexact) Newton iterations, since we used the old Hessian, which made each Newton step cheap but less effective.

4.4. Stopping test. Let ϵ be the desired final accuracy in the objective function. A possible stopping criterion is $\mu \le \epsilon\rho(1+\|\nabla f_0(x)\|)/m$. This stopping test is "exact" at points on the path of analytic centers in that $f_0(x) - f_0(x^*) \le \epsilon(1 + \|\nabla f_0(x)\|)$ (see Appendix A.2). Here, we included a factor $\|\nabla f_0(x)\|$ to make the test invariant under multiplication of f_0 by some positive scalar, and added 1 for the (unlikely) case that $\|\nabla f_0(x)\| = 0$. Thus, ϵ is essentially the relative accuracy in the objective function. However, since the constraints have been shifted, the final iterate is not always feasible; it is only guaranteed that $f_i(x) \le \mu\beta_i$. We therefore included factors $(1 + \|\nabla f_i(x)\|)$ and stopped the algorithm as soon as

$$\mu \le \bar{\mu} := \min\{\ \epsilon\rho(1 + \|\nabla f_0(x^0)\|)/m,\ \min_i \epsilon(1 + \|\nabla f_i(x^0)\|)/(\beta_i + \epsilon)\ \},$$

which guarantees a relative accuracy of ϵ for the constraint violation as well as for the objective value.

4.5. Convergence. Before concluding this description we briefly state some convergence results of our algorithm under the further assumption that the barrier function φ_μ is self-concordant for all $\mu > 0$.

If the stopping criterion at each iteration for Newton's method is $\|h\|_H \le c < 1$ (see also (15)), and if the quantity r in §4.3 is chosen such that it is bounded away from zero, then the following are true.

1. If (CP) satisfies the assumptions of Lemma 3.2, then as $\mu \to 0$ the iterates x^k satisfy the same limit relations (for $k \to \infty$) as stated for $x(\mu)$ in Lemma 3.2.

2. If (CP) has no optimal solution, either $x^k \to \infty$, or we find that $\mu \to \delta > 0$, with δ as in Lemma 3.1 Part 4. (Both cases are hard to identify in an implementation and need special attention.)

Proof. See Appendix B. □

The importance of these results is that we believe they yield a very reliable heuristic for our algorithm, even if we do not know that the barrier function is self-concordant.

5. Important details for the initialization. The following additions to our algorithm do not change the convergence results, but they may be essential for the efficiency of the algorithm.

5.1. Decreasing the perturbation. Recall that w defines the perturbed center. In this section we are concerned with the influence of the magnitude of w on the rate of convergence of our method. Ideally we would like the perturbation w to be zero, in which case the points $x(\mu)$ are the analytic centers. Also if w is close to zero, one can prove for self-concordant barriers that the tangent to the curve $x(\mu)$ closely approximates the curve in some interval $[\mu^1, \mu^2]$ that does not depend on the problem data. For large w (measured in the norm given by $H(x,\mu)^{-1}$) this is no longer true.

Our numerical experiments suggest that in practice also, a large w slows down convergence. It is therefore important to initialize the method such that w is moderate in size. The perturbation w may be considered moderate if $\|w\|_H^2 = w^T \nabla^2 \varphi_1(x^0, 1)^{-1} w$ is strictly less than 1. "Usually," e.g., if the barrier function is self-concordant and if there is a unique optimal solution, one can show that $H(x(\mu),\mu) \to \infty$ as $\mu \to 0$, and that w measured in the above norm tends to zero. If there is more than one optimal solution, this is generally not true, but in this case, the eigenvectors that belong to finite eigenvalues of $H(x(\mu),\mu)$ are—in the limit—parallel to the set of optimal solutions, i.e., the component of w orthogonal to the set of optimal solutions tends to zero as well. Unfortunately, the condition $\|w\|_H < 1$ is only a sufficient condition, and it may be overly pessimistic in some cases.

The size of w depends on the choice of starting point and on the shifts β_i and, in some examples, the initialization outlined in this paper does yield large perturbations w. In these cases it is necessary (to obtain a reasonable rate of convergence) to reduce the size of w before starting the algorithm. We used the following procedure.

1. Before starting the predictor-corrector iterations, set $w = 0$ and introduce additional constraints of the form $(x_i - x_i^0)^2 \le 10^{12} t^2$ (with t as in §3.1).

2. Perform a number of Newton steps for finding the (analytic) center of P_1 with the additional constraints and stop when $\|\Delta x\|_H = (\Delta x^T H(x,1) \Delta x)^{1/2}$ satisfies a given bound. Let the result be $\bar{x}^0$.

3. Remove the additional constraints again and redefine w for the new starting point $\bar{x}^0$ as outlined in §3.3.

This process can be motivated as follows. Suppose for the moment that the set P_1 is bounded. In this case the *analytic* center $\bar{x}$ of P_1 exists ($w = 0$). As a set P_1' we define the set P_1 with some additional bounds that are much larger than the size of P_1. (More precisely, the sets P_1 and P_1' are the same; only the constraint functions defining them are not identical. We note here that it is not quite exact to talk about a center of a set P, but rather about a center of a set of (nonlinear) inequalities describing P. The center depends on the constraint functions.) Let $\tilde{x}$ be the *analytic* center of P_1'. After removing the additional constraints, $\tilde{x}$ is a *perturbed* center of P_1, and the norm of the perturbation is indirectly proportional to the distance of $\tilde{x}$ from the (removed) additional constraints. The additional constraints were necessary since we do not know whether P_1 is bounded. In particular, this procedure does not assume that a bound of the form "$\|x - x^0\|_\infty \le 10^6 t$ for all feasible x" is known a priori; the additional constraints are used only while decreasing w to guarantee that Newton's method converges, and they are later removed.

5.2. Warm start. If we expect that the initial point x^0 is "almost" optimal, we may apply the following "warm start" procedure. Define the quantity t preceding (5) as $\max\{10^{-4}, f_i(x^0)\}$, for example, rather than $\max\{1, f_i(x^0)\}$. (The number 10^{-4} is arbitrary.) Then fix $\rho \ge 1$ to minimize the norm of the gradient of $\varphi_1(x^0)$.

Our motivation for this warm start is the following. From the given x^0 we determine approximate Lagrange multipliers as $1/(\beta_i - f_i(x^0))$. The definition of t

implies that these are positive and bounded by 10^4. The initial parameter ρ is then determined as the "best" weight factor for the objective function.

The reason why we do not suggest this start as standard is that with the fairly large bound of 10^4 for the approximate multipliers, the perturbation w may become quite large.

5.3. The weight factor ρ. In the definition of the center (7) we did not elaborate on the choice of the weight factor ρ in the objective function. However, a good choice of ρ is very important. For one-dimensional examples it is easy to see that a poor choice can result in the curve of centers passing the same point x twice (which is not attractive when following the curve numerically). We applied the following heuristic in our implementation.

1. If the constraints have not been shifted, i.e., if $\beta = 0$, then we choose ρ such that the H-norm $\|x'\|_H$ of the tangent x' at the first iteration is 1.

2. If the constraints have been shifted, i.e., if $\beta \neq 0$, we compute a "tangent" $x^{1'}$ for the case $\rho = 0$ (this is the component of the tangent resulting from the shift β) and a "tangent" $x^{2'}$ by setting $\rho = 1$ and $\beta_i = 0$ for all i in (4.1) (this is the component of the tangent resulting from the objective function). We then choose $\rho = \|x^{2'}\|/\|x^{1'}\|$ so that both components are of the same magnitude.

6. Further comments.

6.1. Solving the linear systems. The search directions in our algorithm (extrapolation and Newton step) are given by linear systems (13)–(14) involving the Hessian $H(x, \mu)$. We are concerned with the stability and the sparsity of these systems. It is well known that the Hessian becomes ill-conditioned if x approaches some point on the boundary of P_μ at which less than n linearly independent constraints are active. Also note the inherent sparsity of H if the functions f_i each depend on few variables only or if there is a small number of separable functions f_i. (For separable f_i the Hessian is a diagonal matrix, but the gradient g_i and thus also $g_i g_i^T$ could be full, so that rank-one update techniques could be used for example.)

Let $\rho = 1$ and $\beta = 0$ for the moment. To illustrate how we would deal with H in a large-scale implementation, note that H (9) can be written in the form

$$H(x, \mu) = \hat{H}(x, \mu) + J^T D^{-2} J,$$

where $\hat{H}(x, \mu) = \frac{1}{\mu} H_0(x) + \sum_{i=1}^m \frac{1}{-f_i(x)} H_i(x)$, $J = (\, g_1(x) \ \dots \ g_m(x)\,)^T$ is the Jacobian of the constraints and $D = \operatorname{diag}(f_i(x))$. Solving a system with H is equivalent to solving a system with

$$K = \begin{pmatrix} -D^2 & J \\ J^T & \hat{H} \end{pmatrix},$$

which can be seen when taking the Schur complement of $-D^2$ within K. This system in turn is equivalent to a system involving

$$K' = \begin{pmatrix} D^{-2} & 0 & I \\ 0 & \hat{H} & J^T \\ I & J & 0 \end{pmatrix}.$$

(Take the Schur complement of D^{-2}.) Systems of the form K' are considered in Gill et al. [6]. The basic idea is that it is better either to factorize K' directly or to

take the Schur complement of just certain parts of the diagonal matrices, such that the Schur complement does not become excessively ill-conditioned and does not suffer excessive fill-in caused by dense rows or columns of J.

6.2. Inherent stability. In view of the ill-conditioning of the linear systems to be solved during the algorithm, one must question the numerical stability of the method. In our numerical examples (see, also, [13]) we did not encounter any difficulties. In particular, we were always able to compute the solution of the problems to the full accuracy given in the literature [27], [28], [33], [9]. The observed robustness may result from two facts that were addressed in [13].

First, we observe that our method will converge to an optimal solution even if the linear systems are solved inexactly. Clearly, inexact Newton's method with linesearch is known to converge eventually at each iteration when minimizing the strictly convex function φ_μ as long as the Newton steps are approximated by using uniformly positive definite approximations $\tilde{H}$ to H. (For the (very poor) approximation $\tilde{H} = I$, for example, the inexact Newton method reduces to the method of steepest descent, which does converge, though very slowly.) Similary, the tangent direction need not be computed exactly, since the predicted point is corrected by Newton's method in the next step.

Second, the corrector step brings the iterate "away from the boundary" towards the center $x(\mu)$. Since $x(\mu)$ is in some sense approximately "equally far away" from all active constraints, we anticipate (and for nondegenerate linear programs this can actually be proven [12]) that the Hessian is better conditioned near $x(\mu)$ than near the boundary of P_μ. Thus the corrector step is likely to reduce the condition number of the linear systems.

We believe that these facts imply an "inherent stability" of our method that would not arise, for example, if the method did not use a centering (corrector) step.

6.3. Note on primal-dual methods. The method outlined above works in the primal space only. We briefly mention the relationship to primal-dual methods.

It is straightforward to convert the KKT conditions of a convex optimization problem into a nonlinear complementarity problem (NCP) that involves primal and dual variables. The functions defining the NCP are monotone (they are the gradients of convex functions), and interior-point algorithms for solving NCPs with monotone functions have been proposed in [17], [18]. Implementations of such primal-dual methods proved to very effective when applied to linear programs [19], [20], and it may be expected that the same also holds for nonlinear problems.

We give a brief comparison of our method to a primal-dual method and consider the search directions first. For simplicity we assume that the objective function $f_0(x) = c^T x$ is linear and that $\rho = 1$. In all interior-point methods, the main focus is on two directions, the centering direction and the affine scaling direction. Assume we are given a current iterate x and a parameter value μ. We construct a dual variable y by setting $y_i = \mu/(\mu\beta_i - f_i(x))$. This construction implies that (x, y) is the analytic center in primal-dual space if x is the analytic center in the primal space. Furthermore, let $J(x)$ be the Jacobian of $(f_1, \ldots, f_m)$. The Newton direction Δx for finding the analytic center, and the tangent direction x' of our algorithm, are given by

$$H(x,\mu)\Delta x = -(c + J(x)^T y), \qquad H(x,\mu)x' = -(c + J(x)^T Y^2 \beta),$$

where $Y = \operatorname{diag}(y_i)$. It turns out that if y is given as above, the centering direction

of the primal-dual method (after taking the Schur complement) is given by

$$H(x,\mu)\Delta x = -(c + J(x)^T y + J(x)^T Y^2 \beta),$$

i.e., it combines the Newton and tangent directions, and the affine scaling direction is exactly the tangent direction above. If the dual variables are obtained in a different way, then the search directions will also be different, but nevertheless we may recognize a close relationship between the two methods. The additional degree of freedom in the choice of y may be an advantage for the primal-dual method. On the other hand, the primal method offers a simple heuristic for the control of the steplength during extrapolation, and a natural merit function for the linesearch during Newton's method. Moreover, the strong theoretical results proved in [12], [30] about the convergence of primal methods for convex programs have not been proved (yet) for primal-dual methods.

6.4. Relationship to classical barrier methods. Our method has much in common with the traditional barrier methods suggested by [3], [2] and implemented in SUMT in 1964. It is natural to ask why these methods did not retain their initial popularity.

As mentioned earlier, it is very important which barrier function is chosen. Only the logarithmic barrier function combines all the theoretical properties that were listed in §2.2; they do not hold for other barriers like $1/f_i(x)$.

Some of the early barrier methods also used logarithmic barrier functions by following some perturbed path as above. The smoothness of the perturbed path was well known then, but no theory existed to explain the importance of the *central* path. In particular, the two-sided ellipsoidal approximation around the analytic center was not known. This approximation implies that the centers $x(\mu)$ are approximately equally far away from all active constraints—measured in the canonical norm $\|\,.\,\|_H$. As we illustrate with some examples below, it may be of great importance to find a point on the central path first and to follow the central path rather than some random perturbed path that depends on the initial point.

Another difficulty that arises during the implementation of a barrier method is how to choose the steplength once the search direction is known. Where is the break-even point between progress in the objective function and staying far away from all constraints, i.e., what is the appropriate merit function for a linesearch? In the approach above, for the linesearch during the centering step, a natural merit function is given (the log-barrier function), and for the extrapolation the heuristic outlined above is fast and robust, at least for all examples that were tested.

As pointed out in [26], [38], the Hessians of the barrier functions become increasingly ill conditioned as the iterates approach an optimal solution x^* if there are less than n linearly independent constraints active at x^*. This difficulty also occurs when solving degenerate linear programs by interior-point methods. The large number of numerical experiments carried out to date suggests, however, that with a careful choice of algebra for solving the linear equations this difficulty can be overcome. We may hope that this is also the case when solving nonlinear problems. In addition, the fact that computers today use a much higher arithmetic precision than was typical 25 years ago makes current codes less sensitive to ill conditioning.

Finally, as pointed out in [5], an approach used in early barrier methods of enforcing equality constraints by a quadratic penalty function (rather than linearizing them at each step) might have introduced further numerical instability. The theoretical results of §2.2, such as the ellipsoidal approximations, do not hold for quadratic penalty

functions. It is not yet clear whether barrier methods are effective for nonconvex problems (such as ones with nonlinear equality constraints).

Many implementations of interior-point methods for the conceptually simpler problem of solving linear programs have been tested in the recent past. These implementations documented the great importance of good sparse-matrix techniques. Without the latter, interior-point methods for large linear programs are completely unattractive, and the same may be true for nonlinear problems. It may be anticipated, however, that interior-point methods applied to certain classes of "inherently sparse" (e.g., separable) nonlinear problems with cheap first and second derivatives will be able to exploit the additional structure and yield fast special-purpose solvers. For a simple characterization of "inherently sparse" programs based on the notion of "partially separable" functions, we refer to [1].

7. Numerical experiments. The method of §4 was implemented in MATLAB [24] and tested on a few problems with up to 300 unknowns without exploiting sparsity or special structure of the problem. As mentioned before, the use of sparse-matrix techniques will be crucial for the efficiency of this method. The development of efficient interior-point methods for linear programs took several years and similar efforts may be needed for developing an interior-point method for nonlinearly constrained problems. The goal of the implementation here was merely to illustrate the behavior of the method in terms of number of iterations and Newton corrections, and to test various parameters (such as β and ρ) that define the barrier function.

The statistics gathered read as follows. Each iteration involves computation of the tangent and a small number (1–10) of inexact Newton steps. The tangent and the Newton steps are computed from a linear system that involves the Hessian of φ. We used each Hessian (or rather its factorization) for five inexact Newton steps before we recomputed a new Hessian, so that sometimes more than one Hessian was computed in an iteration. Each Hessian is used for several inexact Newton steps; its computation and factorization dominates the overall computation.

7.1. Problem Manne. The following two problems are taken from [27]. Problem Manne1 involves 300 variables, a logarithmic objective function, 100 nonlinear constraints, 100 linear inequalities, and 400 simple bounds. Problem Manne2 is identical except that it has only 300 simple bounds.

Results are given in [27] for MINOS. Manne1 took 7 major iterations, 183 minor iterations, 497 function evaluations, and 12 seconds on an IBM 370/168, while Manne2 required 11 major iterations, 355 minor iterations, 859 function evaluations, and 34 seconds. MINOS performs best if a high number of linear constraints or bounds are active at the optimal solution, thus reducing the size of the (dense) systems that are solved in each iteration. For Manne1, the size of the dense systems grew to 25, and for Manne2 they grew to 99 (since some of the active bounds in Manne1 were removed).

In contrast, the size of the systems to be solved in each iteration of the interior-point algorithm is always 300, i.e., the number of variables. For both Manne1 and Manne2, these systems are sparse and of diagonal structure, with at most seven nonzeros per row.

In Table 1 we report the results of our method for both examples. As a starting point we chose the (infeasible) vector of all ones. (The objective and some of the constraints are not defined for $x = 0$.) Unlike MINOS, the interior-point method performed slightly better on Manne2 than on Manne1, giving hope that for certain problems in which the active constraints do not significantly reduce the dimension of

the MINOS subproblems, interior-point algorithms may become an attractive alternative.

TABLE 1
Results for problem Manne.

Problem	Large w	Manne1	Manne2	Small problem
Iterations	38	19	15	11
Hessians	40	24	20	13
Newton steps	115	76	61	38
$\max \lvert\beta_i\rvert$	2.05	2.05	2.05	2.05
ρ	4.1	4.1	3.8	0.29
Final μ	3.4e-9	5.7e-9	2.2e-8	8.7e-9
Objective	9.287556	9.287556	9.330183	2.670098
Constraint violation	4.9e-9	8.1e-9	3.1e-8	1.1e-8

Problem Manne was one for which our initialization in §3.3 resulted in a vector w of norm $(w^T H^{-1} w)^{1/2} \approx 15$. By the procedure in §5.1 we decreased the norm of w to about one before starting the iterations. For comparison we also list the results for Manne1 without reducing the size of w. In this case convergence was very slow, and for many iterations the maximum steplength $\alpha_{\max}$ during extrapolation was less than 0.25. (At each iteration we anticipate a steplength of say $\alpha_{\max} \geq 0.9$ that would reduce μ by at least 90%.)

We also give results for a smaller version of problem Manne1 with only 30 unknowns, to show that the number of iterations grows only moderately with the number of variables for this particular problem.

7.2. Problem 385, Schittkowski. This is a problem with 15 unknowns, 10 convex quadratic constraints, and a linear objective function. It is taken from [33], where it was solved with NLPQL [32] using 693 function evaluations and 242 gradient evaluations. Running times or numbers of arithmetic operations are not reported in [33]. The starting point (zero) was strictly feasible and β was zero. (Hence, P_μ was constant and also the final point was strictly feasible.) Our implementation took 11 iterations to solve the problem, a total of 11 evaluations of the Hessian, 27 Newton steps (each of which requires the evaluation of the gradient of φ), and 64 additional gradient evaluations for the linesearch steps. The steplength α was 0.80 on average, ranging from 0.70–0.92. The Hessians of the constraints are diagonal, but to preserve the sparsity of the Hessians of φ, the dense outer products of the gradients in (9) must be treated separately (for example as in §4.4).

7.3. Problem 386, Schittkowski. This is the same as problem 385 above (except that two entries in the coefficients of the constraints are changed) with an additional *concave* constraint. In [33], 900 function evaluations and 327 gradient evaluations were required to solve the problem to 6 digits of accuracy. To explore the limit of applicability of our method we tested this problem with different parameter settings.

1. Using the standard method, the Hessian of φ became indefinite in the eighth iteration and our algorithm failed.

2. In a second run we set the Hessian of the *concave* constraint equal to zero (this corresponds to a linearization of this constraint at each iteration) but kept all other second-derivative information. The method converged to the true solution in 10 iterations using 10 evaluations of the Hessian of φ and 28 Newton steps.

3. In a third run we set the Hessians of all constraints to zero. In this case, the Hessian of φ was indefinite to begin with (since there were only 11 linearized constraints in a 15-dimensional space) and the method failed again.

7.4. A linear problem. Here we briefly describe the results of applying our method to a linear program. In [9], Iri and Imai considered a diet problem with 17 variables, 11 constraints, and 17 simple bounds, which they solved with a primal-dual interior-point algorithm in 18 iterations to 7 digits of accuracy. (More recent interior-point algorithms for linear programs may be more efficient.) Our code is not primarily intended for linear programs; we are merely interested in its behavior compared with nonlinear programs. For this purpose, we tested our method with different starting points and with small and large w. Table 2 gives results for $x^0 = \lambda(1, \ldots, 1)^T$ with different values of λ. (The optimal solution has nonnegative entries of size ≤ 6.5.) We list the initial $\|w\|_H$ and $\max|\beta_i|$, as well as the number of iterations, Hessian evaluations, and inexact Newton steps. In all cases we set $\epsilon = 10^{-7}$ in the stopping test and obtained the same accuracy as in [9]. We see that varying w and x^0 did not have a great effect.

TABLE 2
Results for a linear program.

λ	$\|w\|_H$	$\max\|\beta_i\|$	Iter.	Hess.	Newt.
0	0.63	4.27	12	15	33
0	8.6	4.27	14	14	33
1	0.73	0.74	12	15	34
1	4.4	0.74	12	13	29
10	0.45	0	13	16	35
10	6.1	0	15	15	32
−100	0.65	588	13	16	36
−100	9.9	588	17	17	40

8. Conclusions. The design of fast and stable implementations of interior-point algorithms is marked by conflicting principles.

1. It is desirable to maintain some polynomiality results, since they limit the dependence of the method on the data of a particular problem.

2. It is preferable to perform only few (if any) centering steps, since they do not give much progress towards optimality.

3. The linear systems involved should be kept well conditioned.

4. Given a search direction (the tangent at $x(\mu)$), it is desirable to take a large step (close to the boundary of the feasible set) to make fast progress towards optimality.

A typical example of how these concepts conflict with each other in the above method is the number of centering steps and the steplength along the tangent. Without centering there exist no polynomiality results and the Hessians become very ill conditioned, yet the best theoretical complexity can be proven for methods that use only centering, and those methods are completely unattractive in practice. The closer the extrapolation along the tangent to the boundary, the more ill conditioned the Hessian of the barrier function and the worse the theoretical complexity, suggesting that one should not take too large steplengths. The concept of numerical stability based on the condition number of a matrix however is not perfect. For example, interior-point methods for linear programs evidently perform best when taking steps of 99.995% to the boundary [19], in spite of the ill conditioning introduced by these large steps. It

is an open question how far the extrapolation should go for nonlinear programs, but in view of the nonlinearity in the constraints we believe it is more efficient to take shorter steps in our algorithm.

Appendix A. Some results on the sets P_μ.

A.1. Uniform convergence in K_r. Part 2 of Lemma 3.1 also holds in the following stronger form.

LEMMA 8.1. *Let* $K_r := \{ x \mid \|x\|_2 \le r \}$ *and* $\mathbf{P}_\epsilon := \{ x \mid x = y + z,\ y \in P,\ \|z\|_2 \le \epsilon \}$. *Then for all finite* r *and all positive* ϵ *there is a positive* μ *such that*

$$P_\mu \cap K_r \subset \mathbf{P}_\epsilon,$$

which implies that P_μ *converges uniformly to* P *in any ball* K_r. *The restriction to a bounded set* K_r *is necessary, since there are examples for which* $P_\mu \not\subset \mathbf{P}_1$ *for any* $\mu > 0$.

Proof. We prove the first statement by contradiction. Suppose there was a finite r and a positive ϵ such that for all $\mu > 0$, $P_\mu \cap K_r \not\subset \mathbf{P}_\epsilon$. Let $\mu^k \to 0$, $\mu^k \in (0,1)$ be a sequence and $x^k \in (P_{\mu^k} \cap K_r) \backslash \mathbf{P}_\epsilon$. Since $\|x^k\| \le r$ there is an accumulation point $\bar{x}$. Clearly $\bar{x} \in P$ (otherwise there exists $i_0 :\ f_{i_0}(\bar{x}) > 0$ and then by continuity of f_{i_0}, there exists $\delta, \sigma > 0 :\ f_{i_0}(\bar{x} + K_\delta) > \sigma$, contradicting the definition of $\bar{x}$). By construction we also know that $\bar{x} \notin \mathbf{P}_\epsilon^o$, in contradiction to $P \subset \mathbf{P}_\epsilon^o$.

That the more general statement, "$P_\mu \subset \mathbf{P}_\epsilon$ for small enough μ," is not true can be seen from a simple counterexample. Take the convex function $f(x,y) := y^2/x - 1$ with domain $S := \{ (x,y) \mid x \ge 1 \}$, defining $P := \{ (x,y) \in S \mid y^2/x \le 1 \}$. It is easy to verify that $P_\mu = \{ (x,y) \mid x \ge 1,\ |y| \le \sqrt{x + \mu x} \} \not\subset \mathbf{P}_1$ for any $\mu > 0$.

We may modify the above example such that P^o is empty by using the function $y^2/x \le 0$. Thus, P shrinks to the set $\{ (x,0) \mid x \ge 1 \}$. Note that $\min\{ y \mid (x,y) \in P \}$ exists in this case, but not so $\min\{ y \mid (x,y) \in P_\mu \}$ for $\mu > 0$, and neither does a perturbed center exist for $0 < \mu < 1$. (This example shows another surprising property. If f_1 and f_2 are convex functions that each have a minimum on a common closed set S, then $f_1 + f_2$ may not have a minimum on S; e.g., take y^2/x and $(y-1)^2/x$ on the above set S.) □

Despite these counterexamples, the vague intuition that P_1 might not be much "bigger" than P can be formalized in the following simple lemma.

LEMMA 8.2. *If* P_δ *is nonempty and bounded for some* $\delta \in [0,1)$, *then so is* P_1.

Proof. The proof is straightforward. □

A.2. Proof of near-optimality of the analytic center The following result justifies the stopping test used in our program. We prove the inequality $f_0(x(\mu)) - f_0(x^*) \le m\mu$ for the case that $x(\mu)$ is the analytic center and $\rho = 1$ (modification of Lemma 3.8 in [10]).

Let $\bar{x} = x(\mu)$ and $z = x^* - x(\mu)$. Set

$$\Psi(t) := \frac{f_0(\bar{x} + tz)}{\mu} - \sum \log(\mu\beta_i - f_i(\bar{x} + tz)).$$

By definition of $x(\mu)$,

$$\Psi'(0) = \frac{\nabla f_0(\bar{x})^T z}{\mu} - \sum \frac{\nabla f_i(\bar{x})^T z}{f_i(\bar{x}) - \mu\beta_i} = 0.$$

By the convexity of $f_i(x)$, $0 \le i \le m$, it holds that

$$f_i(\bar{x}) - f_i(\bar{x} + z) \le -\nabla f_i(\bar{x})^T z.$$

Combining the last two relations it follows that

$$f_0(x(\mu)) - f_0(x^*) \le -\nabla f_0(\bar{x})^T z = -\mu \sum \frac{\nabla f_i(\bar{x})^T z}{f_i(\bar{x}) - \mu\beta_i}$$

$$\le \mu \sum_{i=1}^{m} \frac{f_i(\bar{x}) - f_i(\bar{x} + z)}{f_i(\bar{x})} \le m\mu.$$

The last inequality follows from the feasibility of $\bar{x} + z$: $f_i(\bar{x} + z) \le 0$ (and of course $f_i(\bar{x}) < 0$).

If the barrier function is self-concordant, a similar estimate follows from the ellipsoidal approximations in [12] for the perturbed center if $\|w\|_H \le 0.2$.

Appendix B. Convergence of the centers. We are now ready to show the results stated in Lemma 3.2.

B.1. Proof of Lemma 3.2.

Part 1. From our assumption "$\sum H_i + g_i g_i^T > 0$" it follows that φ_μ is strictly convex for $\mu > 0$. Hence, if the center exists, it is unique. Assume the perturbed center $x(\mu)$ does not exist for some $\mu \in (0, 1)$, but $x(1)$ does; i.e.,

$$\varphi_1(x) = -\sum \ln(\beta_i - f_i(x)) + f_0(x) + w^T x$$

has a minimum $x(1)$ (without loss of generality let $x(1) = 0$ for the moment), and

$$\varphi_\mu(x) = -\sum \ln(\mu\beta_i - f_i(x)) + \frac{f_0(x)}{\mu} + w^T x$$

does not. For $R \ge 0$ let $x(R) = \arg\min_{\|x\| \le R} \varphi_\mu(x)$. From convexity of φ_μ it follows that $\|x(R)\| = R$. From strict convexity of φ_1 it follows that $\varphi_1(x(R)) \ge \varphi_1(0) + \epsilon R$ for $R \ge 1$ and some fixed $\epsilon > 0$. (Since 0 is the unique minimum, $\varphi_1(x) \ge \epsilon$ for $\|x\| = 1$.) Hence,

$$\varphi_1(0) + \epsilon R \le \varphi_1(x(R)) \le \varphi_\mu(x(R)) - \left(\frac{1}{\mu} - 1\right) f_0(x(R)),$$

using the monotonicity of ln and $\beta_i \ge 0$ in the last inequality. Since $\varphi_\mu(x(R)) \le \varphi_\mu(0) < \infty$, this implies that $f_0(x(R)) \to -\infty$ at least linearly.

Case 1. Suppose P has an interior point $\bar{x}$ and $f_0(x) > -M$ for $x \in P$. Without loss of generality, now let $\bar{x} = 0$ and $f_0(\bar{x}) = 0$. (Note that by this change in coordinates we still have $\|x(R)\| = R$, where $x(R)$ is as above.) Let $\delta > 0$ be such that $f_i(\bar{x}) < -\delta$ for all $i \in 1, \ldots, m$. Let $\theta > \max_i \beta_i + \delta$, $\lambda = \delta/\theta < 1$ and take R large enough such that $f_0(x(R)) < -M/\lambda$. Since $f_i(x(R)) < \mu\beta_i$ for all i, it follows from convexity of f_i that

$$f_i(\lambda x(R)) \le -\delta + \lambda(\mu\beta_i + \delta) \le 0,$$

i.e., $\lambda x(R) \in P$. On the other hand,

$$f_0(\lambda x(R)) \le \lambda f_0(x(R)) < -\lambda \frac{\theta}{\delta} M = -M.$$

This is a contradiction.

Case 2. Suppose that the set of optimal solutions is nonempty and bounded. Without loss of generality, let $x^* = 0$ be an optimal solution with $f_0(0) = 0$, and let all other optimal solutions satisfy $\|x^*\| \leq 1$ and $f_0(x^*) = 0$. Thus, $f_i(0) \leq 0$ for all $i = 0, \ldots, m$ and there exists $\epsilon > 0$ such that for any $\|x\| = 2$ we have $\max_{0 \leq i \leq m} f_i(x) \geq \epsilon$. ($x$ is either not feasible or not optimal.) It follows that $\max_i f_i(x(R)) > \epsilon R/2$ for $R \geq 2$, by convexity of f_i. Let us fix R large enough and consider the index i for which $f_i(x(R)) > \epsilon R/2$. If $i \geq 1$ and $R > 2\mu\beta_{\max}/\epsilon$, where $\beta_{\max} = \max_{1 \leq i \leq m} \beta_i$ we obtain a contradiction, since $f_i(x(R)) > \mu\beta_i$, i.e., $x(R)$ is not in P_μ. Hence, $i = 0$. Again, this is in contradiction since $f_0(x(R))$ goes to $-\infty$.

Part 2. First, we briefly sketch a proof that guarantees that $x(\mu)$ is bounded if the set of optimal solutions for (CP) is nonempty and bounded.

Assume $x(\mu)$ is unbounded for $\mu \in (0, 1]$. By continuity of $x(\mu)$ this implies that there exists a sequence $\mu^k \to 0$ such that $\|x(\mu^k)\| \to \infty$. Without loss of generality let 0 be an optimal solution with $f_0(0) = 0$, and suppose $\|x^*\| \leq 1$ for all optimal solutions. As in Case 2 above, we conclude that there is an $\epsilon > 0$ such that $f_i(x) \geq \epsilon$ for some $i \in 0, 1, \ldots, m$ if $\|x\| = 2$. If $R > 2\mu\beta_{\max}/\epsilon$ and $\|x(\mu)\| \geq R$, then $f_0(x(\mu)) \geq R\epsilon$ (since $x(\mu) \in P_\mu$). Now we may choose M sufficiently large and $\mu < 1/M$ such that $\|x(\mu)\| = R > M$. We obtain

$$\varphi_\mu(x(\mu)) = \frac{f_0(x(\mu))}{\mu} + w^T x(\mu) - \sum \log(\mu\beta_i - f_i(x(\mu))) = a + b - \sum \log c_i.$$

Here $a \geq \epsilon M R$, $|b| \leq R\|w\|$, and $0 < c_i \leq \mu\beta_i - \nabla f_i(0)^T x(\mu) \leq \mu\beta_i + \|\nabla f_i(0)\| R$. From convexity we also obtain that

$$\varphi_\mu(\tfrac{1}{2}x(\mu)) \leq \tfrac{1}{2}a + \tfrac{1}{2}b - \sum \log(\tfrac{1}{2}c_i).$$

By choosing M large enough, a becomes the dominant term in φ_μ and $\varphi_\mu(\frac{1}{2}x(\mu)) < \varphi_\mu(x(\mu))$, which is a contradiction.

Now we consider the case where the interior of P is nonempty and $\min\{ f_0(x) \mid x \in P \}$ exists. In the sequel we assume again that 0 is an optimal solution of (CP) with $f_0(0) = 0$. Let $\tilde{P}_\mu = P_\mu \cap \{ x \mid f_0(x) \leq 0 \}$ and $P^* = \{ x \in P \mid f_0(x) = 0 \}$. We proceed in two steps.

In step one we show that there exists a finite constant $c > 0$ such that $\inf\{ f_0(x) \mid x \in P_\mu \} \geq -c\mu$ for $\mu \in (0, 1]$. Without loss of generality suppose $\inf\{ f_0(x) \mid x \in P_1 \} < 0$ (else step one follows trivially). Since $P^o \neq \emptyset$, this implies that both $\tilde{P}_1$ and $P_1 \backslash \tilde{P}_1$ have interior points. Hence the relative interior $(P^*)^i$ of P^* is contained in the interior P_1^o of P_1. In particular, the derivatives of all functions f_i exist in P^*. (By assumption, the shifts β_i are chosen such that $S \subset P_1$.) Assume the functions $f_1, \ldots, f_k$ for some $k \geq 0$ are active for all $x \in P^*$, $f_i(x) = 0$ for $1 \leq i \leq k$, and $x \in (P^*)^i$. The KKT condition implies $\sum_{i=1}^k \lambda_i \nabla f_i(0) + \nabla f_0(0) = 0$ with some $\lambda_i \geq 0$. (This is where we use the existence of an interior point.) Now by convexity, the linearized problem satisfies

$$l(\mu) := \min\{ \nabla f_0(0)^T x \mid \nabla f_i(0)^T x \leq \mu,\ 1 \leq i \leq k \} \leq \min\{ f_0(x) \mid x \in P_\mu \}.$$

Clearly, $l(\mu)$ is linear in μ and the claim of step one follows.

In step two we show that $x(\mu)$ is bounded. Assume it is not, and there exists an infinite direction $x(\infty)$. (By this we mean an accumulation point of $x(\mu^k)/\|x(\mu^k)\|$,

where $\mu^k \to 0$ is a sequence such that $\|x(\mu^k)\|$ is unbounded.) We may conclude that $w^T x(\infty) \leq 0$ using an analogous argument as above and the fact that $f_0(x)/\mu$ is bounded for $x \in P_\mu$ and the barrier term is sublinear. By construction (w defined a first center), $w = -\sum_{i=0}^m \nabla f_i(x^0)\kappa_i$ with some positive constants κ_i. Hence, $\sum \kappa_i \nabla f_i(x^0)^T x(\infty) \geq 0$, i.e., either $\nabla f_i(x^0)^T x(\infty) = 0$ for all i, in which case φ_μ must be constant along $x^0 + \lambda x(\infty)$, $\lambda \geq 0$, and x^0 was not the unique minimum of φ_1, or there exists i such that $\nabla f_i(x^0)^T x(\infty) > 0$, which again leads to a contradiction.

Part 3. The third part of Lemma 3.2 follows immediately from Lemma 8.1 since $x(\mu)$ is bounded for $\mu \in (0, 1]$.

Part 4. Note that when multiplying (7) by μ we obtain

$$x(\mu) = \arg\min_{x \in P_\mu} \mu\varphi_\mu(x) = \arg\min_{x \in P_\mu} \; f_0(x) - \mu \sum_{i=1}^m \ln(-f_i(x, \mu)) - \mu w^T x.$$

Let $x^\mu := x^* + \mu(x^0 - x^*)$. By convexity, $f_i(x^\mu, \mu) \leq -\mu$ for all i. (In §3.1, the constraints were shifted such that $f_i(x^0) \leq 1$ for all $i = 1, \ldots, m$.) This implies that

$$-\mu \sum_{i=1}^m \ln(-f_i(x^\mu, \mu)) - \mu w^T x^\mu \leq -m\mu \ln(\mu) - \mu w^T x^\mu \to 0$$

as $\mu \to 0$. (Note that $w^T x^\mu$ is bounded for $\mu \in [0, 1]$.) By continuity of f_0 it also follows that $f_0(x^\mu) \to f_0(x^*)$, so that $\limsup_{\mu \to 0} \mu\varphi_\mu(x^\mu) \leq f_0(x^*)$. From Part 2 above and continuity of f_0 it follows that $\liminf_{\mu \to 0} f_0(x(\mu)) \geq f_0(x^*)$. Furthermore, $\mu\varphi_\mu(x(\mu)) \leq \mu\varphi_\mu(x^\mu)$, and since the logarithmic barrier terms $-\ln(-f_i(x(\mu), \mu))$ and $w^T x(\mu)$ are bounded below for $\mu \in (0, 1]$, the claim follows from the above inequalities.

B.2. Substantiation of the convergence results. We do not give a complete proof for our convergence results here—we did not exactly specify an algorithm either—but will explain the reasoning behind the convergence statements in §4.5.

Assume that (CP) satisfies the conditions of Lemma 3.2.

1. First we note that the extrapolation yields a strictly feasible point for φ_μ, so that Newton's method with linesearch will eventually converge at each iteration.

2. Furthermore, $x(\mu)$ is a smooth curve in μ (by the implicit function theorem; the f_i are twice continously differentiable and φ_μ is strictly convex). This implies that the extrapolation will not "stagnate" at some positive value of μ—the assumption that $\alpha_{\max} \to 0$ while $\mu \geq \delta > 0$ leads to a contradiction to the continous differentiability of $x(\mu)$ (straightforward).

3. Finally, from Theorems 1.1 and 1.4 in [30] it follows that x^k satisfies the same limit relation as $x(\mu)$. (The sets $\tilde{P}_\mu = P_\mu \cap \{ x \mid f_0(x) \leq f_0(x^*) + \mu \}$ converge to the optimal set, and the above theorems imply that the iterates x^k are in a fixed multiple (depending on $c < 1$) of inner ellipsoids of the sets $\tilde{P}_\mu$.)

These observations lead to the first statement in §4.5. For the second statement, not the following.

1. If P is empty, clearly the method will stagnate ($\alpha_{\max} \to 0$) while $\mu > \delta > 0$, or Newton's method will diverge (if $x(\mu)$ does not exist).

2. If (CP) is feasible but does not have an optimal solution, it follows that either $x(\mu)$ does not exist, or $x(\mu) \to \infty$ as $\mu \to 0$. In the first case, Newton's method diverges. (The stopping criterion is never satisfied, since $\|\Delta x\|_H < 1$ would imply that the function does have a minimum.) In the latter case, $x^k \to \infty$.

Acknowledgments. The authors wish to thank Dr. Margaret Wright and the unknown referees for their highly constructive criticisms.

REFERENCES

[1] A. R. CONN, N. I. M. GOULD, AND PH. L. TOINT, *An introduction to the structure of large-scale nonlinear optimization problems and the LANCELOT project*, in Computing Methods in Applied Sciences and Engineering, R. Glowinski and A. Lichnewsky, eds., Society for Industrial and Applied Mathematics, Philadelphia, 1990, pp. 42–51.

[2] A. FIACCO AND G. P. MCCORMICK, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, Wiley, New York, 1968.

[3] K. R. FRISCH, *The logarithmic potential method of convex programming*, Technical Report, University Institute of Economics, Oslo, Norway, 1955.

[4] R. M. FREUND, *A potential-reduction algorithm for solving a linear program directly from an infeasible "warm start,"* Math. Programming, 52 (1992), pp. 441–466.

[5] P. E. GILL, Private communication, 1991.

[6] P. E. GILL, W. MURRAY, D. B. PONCELEÓN, AND M. A. SAUNDERS, *Solving reduced* KKT *systems in barrier methods for linear and quadratic programming*, Report SOL 91-7, Department of Operations Research, Stanford University, Stanford, CA, 1991.

[7] P. E. GILL, W. MURRAY, M. A. SAUNDERS, AND M. H. WRIGHT, *Shifted barrier methods for linear programming*, Report SOL 88-9, Department of Operations Research, Stanford University, Stanford, CA, 1988.

[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] M. IRI AND H. IMAI, *A multiplicative barrier function method for linear programming*, Algorithmica, 1 (1986), pp. 455–482.

[10] F. JARRE, *On the convergence of the method of analytic centers when applied to convex quadratic programs*, Math. Programming, 49 (1991), pp. 341–358.

[11] ———, *The method of analytic centers for solving smooth convex programs*, in Optimization, S. Dolecki, ed., Lecture Notes in Mathematics, Vol. 1405, Springer-Verlag, New York, 1989, pp. 69–85.

[12] ———, *Interior-point methods for convex programming*, Appl. Math. Optim., 26 (1992), pp. 287–311.

[13] ———, *An interior-point method for minimizing the largest eigenvalue of an affine combination of symmetric matrices*, SIAM J. Control Optim., 31 (1993), pp. 1360–1377.

[14] F. JARRE, G. SONNEVEND, AND J. STOER, *An implementation of the method of analytic centers*, in Lecture Notes in Control and Information Sciences, Vol. 111, A. Benoussan and J. L. Lions, eds., Springer-Verlag, New York, 1988, pp. 297–307.

[15] N. K. KARMARKAR, *A new polynomial-time algorithm for linear programming*, Combinatorica, 4 (1984), pp. 373–395.

[16] M. KOJIMA, N. MEGIDDO, T. NOMA, AND A. YOSHISE, *A unified approach to interior-point algorithms for linear complementarity problems*, G. Goos and J. Hartmanis, eds., Lecture Notes in Computer Science, Vol. 538, Springer-Verlag, New York, 1991.

[17] M. KOJIMA, S. MIZUNO, AND T. NOMA, *A new continuation method for complementarity problems with uniform P-functions*, Math. Programming, 43 (1989), pp. 107–130.

[18] M. KOJIMA, S. MIZUNO, AND A. YOSHISE, *A little theorem on the big M in interior-point algorithms*, Math. Programming, 59 (1993), pp. 361–375.

[19] I. J. LUSTIG, R. E. MARSTEN, AND D. F. SHANNO, *On implementing Mehrotra's predictor-corrector interior-point method for linear programming*, SIAM J. Optim., 2 (1992), pp. 435–449.

[20] S. MEHROTRA, *On the implementation of a (primal-dual) interior-point method*, SIAM J. Optim., 2 (1992), pp. 575–601.

[21] S. MEHROTRA AND J. SUN, *An interior-point algorithm for solving smooth convex programs based on Newton's method*, Mathematical developments arising from linear programming, Contemporary Mathematics, Vol. 114, Amer. Math. Soc., Providence, RI, 1990, pp. 265–284.

[22] 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.

[23] J. MENNICKEN, *Implementation of a first order central path following algorithm for solving large*

linear programs, Report No. 202, Schwerpunktprogramm der DFG Anwendungsbezogene Optimierung und Steuerung, Institut für Ang. Math. und Statistik, Universität Würzburg, Am Hubland, 1990.

[24] C. MOLER, J. LITTLE, AND S. BANGERT, *PRO-MATLAB User's Guide*, The MathWorks, Inc., Sherborn, MA, 1987.

[25] R. MONTEIRO AND I. ADLER, *An extension of Karmarkar type algorithm to a class of convex separable programming problems with global linear rate of convergence*, Math. Oper. Res., 15 (1990), pp. 408–422.

[26] W. MURRAY, *Analytical expressions for the eigenvalues and eigenvectors of the Hessian matrices of barrier and penalty functions*, J. Optim. Theory Appl., 7 (1971), pp. 189–196.

[27] B. A. MURTAGH AND M. A. SAUNDERS, *A projected Lagrangian algorithm and its implementation for sparse nonlinear constraints*, Math. Programming Study, 16 (1982), pp. 84–117.

[28] ———, *MINOS 5.1 user's guide*, Report SOL 83-20R, Department of Operations Research, Stanford University, Stanford, CA, 1983, 1987.

[29] J. E. NESTEROV AND A. S. NEMIROVSKY, *A general approach to polynomial-time algorithms design for convex programming*, Report, Central Economical and Mathematical Institute, USSR Acad. Sci., Moscow, USSR, 1988.

[30] ———, *Self-concordant functions and polynomial-time methods in convex programming*, Report, Central Economical and Mathematical Institute, USSR Acad. Sci., Moscow, USSR, 1989.

[31] J. RENEGAR, *A polynomial-time algorithm based on Newton's method for linear programming*, Math. Programming, 40 (1988), pp. 59–93.

[32] K. SCHITTKOWSKI, *NLPQL: A FORTRAN subroutine solving constrained nonlinear programming problems*, Ann. Oper. Res., 5 (1985), pp. 485–500.

[33] ———, *More test examples for nonlinear programming codes*, in Lecture Notes in Economics and Mathematical Systems, Vol. 282, M. Beckmann and W. Krelle, eds., Springer-Verlag, New York, 1987.

[34] D. F. SHANNO, Private communication, 1991.

[35] G. SONNEVEND, *An 'analytical centre' for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming*, Lecture Notes in Control and Information Sciences, 84 (1986) pp. 866–878.

[36] G. SONNEVEND AND J. STOER, *Global ellipsoidal approximations and homotopy methods for solving convex analytic programs*, Appl. Math. Optim., 21 (1989), pp. 139–166.

[37] T. TSUCHIYA AND M. MURAMATSU, *Global convergence of a long-step affine scaling algorithm for degenerate linear progrmming problems*, Report No. 423, The Institute of Statistical Mathematics, Minami-Azabu, Tokyo 106, Japan, 1992.

[38] M. H. WRIGHT, *Numerical Methods for Nonlinearly Constrained Optimization*, Ph.D. thesis, Computer Science Dept., Stanford University, Stanford, CA, 1976.

SIAM J. OPTIMIZATION
Vol. 5, No. 1, pp. 172–191, February 1995

009

AN ALL-INCLUSIVE EFFICIENT REGION OF UPDATES FOR LEAST CHANGE SECANT METHODS*

HENRY WOLKOWICZ† AND QING ZHAO†

Abstract. Least change secant methods, for function minimization, depend on finding a "good" symmetric positive definite update to approximate the Hessian. This update contains new curvature information while simultaneously preserving, as much as possible, the built-up information from the previous update. Updates are generally derived using measures of least change based on some function of the eigenvalues of the (scaled) Hessian. A new approach for finding good least change updates is the multicriteria problem of Byrd, which uses the deviation from unity, of the n eigenvalues of the scaled update, as measures of least change. The efficient (multicriteria optimal) class for this problem is the Broyden class on the "good" side of the symmetric rank one (SR1) update called the *Broyden efficient class*. This paper uses the framework of multicriteria optimization and the eigenvalues of the scaled (sized) and inverse scaled updates to study the question of what is a good update. In particular, it is shown that the basic multicriteria notions of efficiency and proper efficiency yield a region of updates that contains the well-known updates studied to date. This provides a unified framework for deriving updates. First, the inverse efficient class is found. It is then shown that the Broyden efficient class and inverse efficient class are in fact also proper efficient classes. Then, allowing sizing and an additional function in the multicriteria problem, results in a two parameter efficient region of updates that includes many of the updates studied to date, e.g., it includes the Oren–Luenberger self-scaling updates, as well as the Broyden efficient class. This efficient region, called the *self-scaling efficient region*, is proper efficient and lies between two curves, where the first curve is determined by the sized SR1 updates while the second curve consists of the optimal conditioned updates. Numerical tests are included that compare updates inside and outside the efficient region.

Key words. least change secant methods, unconstrained minimization, multicriteria efficiency, proper efficiency, quasi-Newton methods, eigenvalues, sizing, scaling

AMS subject classifications. 90C30, 49M37, 49M15

1. Introduction.

We consider a unified multicriteria framework for deriving updates for least change secant methods (also called quasi-Newton methods) for the unconstrained minimization problem

$$\text{(P)} \qquad \min_{x \in \Re^n} f(x),$$

where f is twice continuously differentiable. Starting with a current approximation to a local minimum for (P) (denoted x_c) and a symmetric positive definite (spd) approximation (denoted B_c) for the current Hessian, these methods perform an inexact line search in the Newton direction $d = -H_c g_c$ to find a new point x_+. (Here $g_c = \nabla f(x_c)$ and $H = B^{-1}$.)

Under the assumption that B_c is spd and that the line search satisfies some Wolfe-type conditions, the success of these methods depends on finding an updated spd Hessian approximation B_+, which satisfies the secant equation and preserves current built-up curvature information in B_c. Various update formulae have been proposed. The updates usually arise from some proposed measure of least change, which generally depends on the eigenvalues of the scaled update $B = H_c^{1/2} B_+ H_c^{1/2}$. These updates include the well-known Broyden class of updates, which then includes the

* Received by the editors September 24, 1992; accepted for publication (in revised form) October 28, 1993.

† University of Waterloo, Department of Combinatorics and Optimization, Waterloo, Ontario N2L 3G1, Canada (henry@orion.uwaterloo.ca and combopt@math.uwaterloo.ca). This research was supported by the Natural Sciences Engineering Research Council of Canada grant.

Davidon–Fletcher–Powell (DFP) and Broyden–Fletcher, Goldfarb, Shanno (BFGS) updates and their convex hull, termed the convex class. The BFGS is currently the most popular update. (See e.g., [11] for details on various measures and updates.)

The fundamental concept in multicriteria decisionmaking is that of an *efficient point*, sometimes called a *nondominated solution* or *Pareto optimum*; see, e.g., [9], [22]. This refers to optimal solutions in the presence of multiple objectives. A decision-maker (DM) can then choose a "best" efficient solution based on some utility function; see, e.g., [19]. The framework of least change secant methods has the, possibly conflicting, objectives of minimizing the distances between 1 and each of the n eigenvalues of the scaled update. Currently, only single objectives dealing with some function of the eigenvalues have been used to derive updates.

The starting point of this paper is the result of Byrd [2]. This result states that the efficient class of updates, with respect to the multiobjective optimization problem consisting of the objective functions $|\lambda_i(H_c^{1/2}B_+H_c^{1/2}) - 1|$, $i = 1, \ldots, n$, is the Broyden class on the good side of the SR1 update. This efficient class, called the *Broyden efficient class*, includes the convex class of updates. In this paper we extend the results of Byrd; we use the framework of multicriteria optimization to find a region of efficient updates that includes the Broyden efficient class found by Byrd, as well as other important updates. We first find the efficient class of inverse updates and then show that both the Broyden efficient class and inverse efficient class are proper efficient classes. However, there are many important updates that are not in the Broyden class. In fact, selective sized updates (e.g., [6]) have outperformed updates in the Broyden class. By replacing the 1 in the above functions by t and adding the additional objective function $|t - 1|$, we obtain a new multicriteria problem that results in a two parameter efficient region of updates. This region, called the *self-scaling efficient region* (SSER), contains the sized updates.

In particular, the following measures are functions of the eigenvalues that lead to updates that are contained in the SSER: the weighted Frobenius norm measure that results in the BFGS and DFP updates; see, e.g., [11]; the measure $\psi(A) = \text{trace}(A) - \log(\det(A))$, which is used in the convergence analysis in [4] and also results in the BFGS and DFP updates, see [13]; the standard condition number measure $\kappa(A) = \lambda_1(A)/\lambda_n(A)$, which results in a curve of sized updates, see [2], [25], [18]; the uniform condition number $\omega(A) = (\text{trace}(A)/n)/(\det(A)^{\frac{1}{n}})$ which results in the sized DFP and inverse-sized BFGS updates, also called the Oren–Luenberger self-scaling updates, see [10]; the optimal volume measure $\sigma(A) = \lambda_1(A)/(\det(A)^{\frac{1}{n}})$ and the resulting optimally conditioned, sized, SR1 updates; see [25]; and the dual optimal volume measure $\tau(A) = \text{trace}(A)/\lambda_n(A)$, (see [24]) which leads to the same optimally conditioned, sized, SR1 updates, see [26]. (See [26] for results on these and other condition numbers, their relations to measures for least change, and also their relations to potential functions.)

The recent introduction and popularity of automatic differentiation [16], [12] raises questions on the importance of quasi-Newton methods in the future and, in particular, on whether it is worthwhile expending a lot of energy on finding better and improved updates. However, there are many problems where automatic differentiation is not suitable, e.g., where function evaluations may require an unknown number of iterations. Moreover, though the area of least change secant methods has been intensively studied for a long time (since [7]), there are still many fundamental unanswered questions. In fact, understanding what makes one update better than another, and whether scaling or inverse scaling is better, are still open questions.

(See, e.g., [3].) In addition, proper implementation of quasi-Newton methods to large sparse problems also remains an open problem.

The paper is organized as follows. We first conclude this section with some preliminary notations and results. Section 2 presents the preliminary definitions and notation for multicriteria problems. We then present and prove the above mentioned result of Byrd as well as present the inverse efficient class of updates and show that both classes are proper efficient. Section 3 presents our main result, the efficient region of updates. This region lies between two curves determined by the sized SR1 and the optimally conditioned sized updates. Moreover, by allowing sizing by t but not including the extra function $|t-1|$ in the multicriteria problem, we get an efficient curve of updates corresponding to optimally conditioned sized updates. This region and curve are also proper efficient sets. We conclude with some numerical tests illustrating the efficient region.

1.1. Preliminaries. The update at the new point x_+ (denoted B_+) satisfies the *secant equation*

$$B_+ s = y,$$

where the change in x is $s = x_+ - x_c$ and the change in the gradient is $y = g_+ - g_c$. We let H denote B^{-1} and define the curvature formulas

$$a = y^t H_c y, \quad b = y^t s, \quad c = s^t B_c s.$$

We assume that the current update B_c is spd and the curvature $b > 0$.

The *Broyden class* of updates is

$$B_\phi = B_c - \frac{1}{s^t B_c s} B_c s s^t B_c + \frac{1}{y^t s} y y^t + (1-\phi) s^t B_c s w w^t, \tag{1.1}$$

where

$$w = \frac{1}{y^t s} y - \frac{1}{s^t B_c s} B_c s.$$

Since B_c spd and $b > 0$ are assumed, we have B_ϕ is spd if and only if $\phi < ac/(ac-b^2)$. Choosing $\phi = 0, 1$ yields the well-known DFP and BFGS updates, respectively; the set of updates with $\phi \in [0,1]$ is called the *convex class*; while the *symmetric rank-one* update, denoted SR1, corresponds to the ϕ value $\phi_{SR1} = -c/(b-c)$. (See, e.g., [11], [21] for details.) The current update is *sized* by the positive scalar t, which means it is changed to tB_c; see, e.g., [10].

We work in the space of symmetric matrices equipped with the *trace inner product* $\langle A, B \rangle = \text{trace} AB$. For a symmetric matrix A, we let $\lambda_1(A) \geq \lambda_2(A) \geq \cdots \geq \lambda_n(A)$ denote its ordered eigenvalues. By abuse of notation, we let $\lambda_i(\phi)$ denote $\lambda_i(B(\phi))$ when $B(\phi)$ is a symmetric matrix dependent on the parameter ϕ.

2. Multicriteria optimal updates. In this section we introduce our first multicriteria problem and show that the efficient updates are the Broyden class updates on the good side of the SR1. We also derive the efficient class of inverse updates and show that both classes of updates are properly efficient. We first introduce the multicriteria problems and give some definitions and preliminary results.

Consider the multiobjective optimization problem

$$\begin{aligned} \min \quad & g_i(z), \ i = 1, \ldots, m, \\ \text{subject to } & z \in \Omega, \end{aligned} \tag{2.1}$$

where $g_i : \Re^n \to \Re$, are m real valued objective functions. A point z^1 is dominated by z^2, denoted $z^2 \prec z^1$, if

$$g_i(z^2) \leq g_i(z^1), \quad i = 1, \ldots, m, \quad \text{and} \quad g_j(z^2) < g_j(z^1) \quad \text{for some } j.$$

(We use $z^2 \preceq z^1$ if z^1 is weakly dominated by z^2, i.e., if $\leq$ holds for all i without the possible strict $<$.) The efficient set is the set of points $z \in \Omega$ that are not dominated by any points in Ω. z^0 is said to be a properly efficient solution of (2.1) if it is in the efficient set and if there exists a scalar $M > 0$ such that, for each i, we have

$$g_i(z) < g_i(z^0) \Rightarrow \left\{ \exists j \text{ s.t. } g_j(z) > g_j(z^0) \text{ and } \frac{g_i(z^0) - g_i(z)}{g_j(z) - g_j(z^0)} \leq M \right\}$$

(see [15]). Thus, for any point z, we can find a constant M, such that for each gain there exists a loss where the gain over the loss is bounded above by M. Thus, the marginal improvement is bounded. Note that an efficient point that is not proper efficient allows for an arbitrarily large gain in one objective function at the expense of only a small loss in another objective function. Therefore, efficient points that are not proper are not desirable.

We now present some preliminary results on the parametrization of the Broyden class; see (1.1).

LEMMA 2.1. [21, pg. 111] *The matrix* $B_c^{-1/2} B_\phi B_c^{-1/2}$ *has* $n-2$ *unit eigenvalues and the two remaining eigenvalues are*

$$\lambda_\pm(\phi) = f_1(\phi) \pm (f_1(\phi)^2 - f_2(\phi))^{\frac{1}{2}}, \tag{2.2}$$

where

$$f_1(\phi) = \frac{a(b+c) - \phi(ac - b^2)}{2b^2}, \qquad f_2(\phi) = \frac{a}{b} - \frac{\phi(ac - b^2)}{bc}. \tag{2.3}$$

Every member of the Broyden class satisfies the secant equation. The following result of Davidon gives conditions for the converse.

PROPOSITION 2.1. [8] *Suppose that* B_+ *is* spd *and satisfies the secant equation. Then the columns of* $B_+ - B_c$ *are in the span of* $\{B_c s, y\}$ *if and only if* B_+ *is in the Broyden class.*

COROLLARY 2.1. *Suppose that* B_+ *satisfies the secant equation. Then* B_+ *is in the Broyden class if and only if the matrix* $B = H_c^{1/2} B_+ H_c^{1/2}$ *is an update of* I *of rank at most two and it has* $n-2$ *unit eigenvalues with corresponding* $n-2$ *dimensional eigenspace orthogonal to span*$\{B_c^{1/2} s, H_c^{1/2} y\}$.

Proof. Proposition 2.1 above implies that B_+ is in the Broyden class if and only if the columns of $B - I$ are in span$\{B_c^{1/2} s, H_c^{1/2} y\}$, i.e., $B - I$ is at most rank two and has eigenvectors corresponding to nonzero eigenvalues in the above span. □

We will also need the following lemma which we call the *perturbation lemma.*

LEMMA 2.2. *Let* B *be a symmetric matrix with eigenvalues* $\lambda_1 \geq \cdots \geq \lambda_n$. *Suppose that* B *satisfies the scaled secant equation* $B(B_c^{1/2} s) = H_c^{1/2} y$. *Then the following are true.*

(i) *For any two eigenvalues* $\lambda_k \geq \lambda_j > 1$, *we can find a symmetric matrix* $\bar{B}$ *satisfying the secant equation* $\bar{B}(B_c^{1/2} s) = H_c^{1/2} y$, *such that its* n *eigenvalues* $\bar{\lambda}_1, \ldots, \bar{\lambda}_n$

satisfy

$$\lambda_i \geq \bar{\lambda}_i \geq 1 \quad \text{for } i = k, j$$
$$\lambda_i = \bar{\lambda}_i \quad \text{for } i = 1, \ldots, n, \quad i \neq j, k,$$

with at least one of $\bar{\lambda}_k < \lambda_k$ *or* $\bar{\lambda}_j < \lambda_j$;

(ii) *For any two eigenvalues* $\lambda_k \leq \lambda_j < 1$, *we can find a symmetric matrix* $\bar{B}$ *satisfying the scaled secant equation* $\bar{B}(B_c^{1/2}s) = H_c^{1/2}y$, *such that its n eigenvalues* $\bar{\lambda}_1, \ldots, \bar{\lambda}_n$ *satisfy*

$$\lambda_i \leq \bar{\lambda}_i \leq 1 \quad \text{for } i = k, j$$
$$\lambda_i = \bar{\lambda}_i \quad \text{for } i = 1, \ldots, n, \quad i \neq j, k,$$

with at least one of $\bar{\lambda}_k > \lambda_k$ *or* $\bar{\lambda}_j > \lambda_j$.

Proof. To prove (i), let $x_1, \ldots, x_n$ be a set of orthonormal eigenvectors corresponding to $\lambda_1, \ldots, \lambda_n$. Find α, $\beta \in \Re$, not both 0, such that

$$(s^t B_c^{1/2})[x_k \; x_j] \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = 0,$$

i.e., $0 \neq \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \in z^{\perp}$, $z^t = (s^t B_c^{1/2})[x_k \; x_j]$. Set

$$0 \neq w = [x_k \; x_j] \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha x_k + \beta x_j.$$

Let

$$\bar{B} = B - \epsilon w w^t \tag{2.4}$$

and denote its eigenvalues by $\bar{\lambda}_i$ with corresponding eigenvectors $\bar{x}_i$, $i = 1, \ldots, n$. We can see that $\bar{B}$ satisfies the secant equation. Moreover,

$$\bar{\lambda}_i = \lambda_i, \quad i = 1, \ldots, n, \quad i \neq k, j$$

and for $\epsilon > 0$, since w is in the span of $\{x_k, x_j\}$ and $w \neq 0$,

$$\bar{\lambda}_i \leq \lambda_i, \quad i = k, j,$$

with at least one of $\bar{\lambda}_k < \lambda_k$ or $\bar{\lambda}_j < \lambda_j$.

By a similar argument with $\epsilon < 0$, we can show (ii). □

2.1. Broyden efficient class. The general notion of least change is that we want B_+ spd such that the secant equation $B_+ s = y$ holds and B_+ is "close" to B_c. If we view B_c and B_+ as quadratic forms, then we can define close as satisfying

$$\frac{u^t B_+ u}{u^t B_c u} \text{ is close to 1} \quad \text{for all } u \in \Re^n. \tag{2.5}$$

Equivalently,

$$\frac{x^t H_c^{\frac{1}{2}} B_+ H_c^{\frac{1}{2}} x}{x^t x} \text{ is close to 1 for all } x \in \Re^n. \tag{2.6}$$

This implies that all of the eigenvalues of $H_c^{1/2}B_+H_c^{1/2}$ satisfy

$$\lambda_i \text{ is close to } 1, \qquad i = 1, \ldots, n. \tag{2.7}$$

(If we view the inverse updates instead, we would consider the eigenvalues of $B_c^{1/2}H_+B_c^{1/2}$.) Motivated by this, Byrd [2] considers the following multiobjective least change problem, where the functions in (2.1) are $g_i(x) = |x - 1|$:

$$\begin{aligned} &\min_{B_+} \quad |\lambda_i(H_c^{\frac{1}{2}}B_+H_c^{\frac{1}{2}}) - 1|, \qquad i = 1, \ldots, n, \\ &\text{subject to } B_+s = y, \qquad B_+ = B_+^t. \end{aligned} \tag{2.8}$$

Let

$$\Phi = \begin{cases} [\phi_{SR1}, \infty) & \text{if } b > c, \\ (-\infty, \phi_{SR1}] & \text{if } b < c, \\ (-\infty, \infty) & \text{if } b = c. \end{cases} \tag{2.9}$$

We let Φ represent the subset of the Broyden class of updates $\{B_\phi,\ \phi \in \Phi\}$, and call it the *Broyden efficient class.* This is the set of Broyden class updates on the good side, or convex class side, of the SR1. We now state the result presented by Byrd [2]. We provide our own proof for completeness.

THEOREM 2.1. *The efficient set for problem* (2.8) *consists of the Broyden efficient class, i.e., the Broyden class with* $\phi \in \Phi$.

Before proving the theorem, we first note that the efficient class for our multicriteria problem has the following dominating property.

LEMMA 2.3. *Suppose that* $B = B^t$ *satisfies the secant equation. Then there exists an efficient update* $\bar{B}$ *for* (2.8) *such that* $\bar{B} \preceq B$.

Proof. The proof follows by an application of Zorn's Lemma upon noting that the set of updates $\{\bar{B} : \bar{B} \preceq B\}$ is compact; see, e.g., Theorem 1 in [1]. □

We now use the perturbation lemma and the above lemma to prove the theorem.

Proof. Let B_+ be in the efficient set for problem (2.8) and let $B = H_c^{1/2}B_+H_c^{1/2}$. Then B satisfies the scaled secant equation

$$B(B_c^{\frac{1}{2}}s) = (H_c^{\frac{1}{2}}y). \tag{2.10}$$

Let $\lambda_1 \geq \cdots \geq \lambda_n$ be the eigenvalues of B with $x_1, \ldots, x_n$ a set of corresponding orthonormal eigenvectors. First we show that

$$\lambda_2 = \lambda_{n-1} = 1. \tag{2.11}$$

Suppose not and suppose that

$$\lambda_2 > 1. \tag{2.12}$$

By Lemma 2.2 we can find an update $\bar{B}$ such that $\bar{B} \prec B$. This contradicts the fact that B_+ is an efficient point. So (2.12) fails. Similarly, we cannot have $1 > \lambda_{n-1}$, i.e., (2.11) must hold and B is an update of I of at most rank-two. It remains to show that B is in the Broyden class with parameter $\phi \in \Phi$. We do this by first showing that it solves the following problem

$$\begin{aligned} \min_A \quad & \lambda_1(A) \\ \text{subject to} \quad & AB_c^{\frac{1}{2}}s = H_c^{\frac{1}{2}}y, \\ & \lambda_n(A) \geq \lambda_n(B), \\ & A = A^t. \end{aligned} \tag{2.13}$$

Since λ_1 is a convex function while λ_n is a concave function, the above is a convex programming problem. Since $||A|| = \max\{\lambda_1(A), -\lambda_n(A)\}$, we can assume that the feasible set is bounded. Therefore, the minimum is attained say at $\bar{A}$. Then, since B is feasible, we have

$$\lambda_1(B) \geq \lambda_1(\bar{A}) \geq \lambda_n(\bar{A}) \geq \lambda_n(B). \tag{2.14}$$

By Lemma 2.3, there must be an efficient update that weakly dominates $\bar{A}$ and, since any efficient update satisfies (2.11), it weakly dominates B as well. The efficiency of B now implies that $B = \bar{A}$ solves (2.13).

We now use the optimality conditions for (2.13) with Lemma 2.3 to show that the update is in the Broyden class. For some Lagrange multiplier vector u and nonnegative scalars v, w, the Lagrangian for problem (2.13) is

$$L(u, v, w, \bar{B}) = w\bar{\lambda}_1 + u^t(\bar{B}B_c^{\frac{1}{2}}s - H_c^{\frac{1}{2}}y) - v(\bar{\lambda}_n - \lambda_n),$$

where u, v, w cannot all be 0 and $\bar{\lambda}_i = \lambda_i(\bar{A})$. We have added the multiplier w to avoid assuming a constraint qualification, i.e., in the absence of a constraint qualification we necessarily have $w = 0$; while if a constraint qualification holds for (2.13), then we can assume that $w = 1$. We can differentiate the Lagrangian and set it equal to zero to get the Lagrange equation (or Fritz John stationarity condition)

$$w\bar{\lambda}_1' + us^tB_c^{\frac{1}{2}} + B_c^{\frac{1}{2}}su^t + v\bar{\lambda}_n' = 0. \tag{2.15}$$

Here $\bar{\lambda}_i'$ are subgradients; see, e.g., [20]. First suppose that a constraint qualification holds, so $w = 1$, and suppose that $v > 0$. Since the rank of $B_c^{1/2}su^t + us^tB_c^{1/2}$ is at most 2, we must have the ranks of the subdifferentials equal to 1, i.e.,

$$\bar{\lambda}_1' = \bar{x}_1\bar{x}_1^t \quad \text{and} \quad \bar{\lambda}_n' = \bar{x}_n\bar{x}_n^t.$$

(Note that $u = 0$ implies that $\bar{\lambda}_1 = \bar{\lambda}_n$, which would uniquely define the trivial identity update. In fact, the entire Broyden class must reduce to this trivial update in this case, since necessarily $a = b = c$.) Now (2.15) implies that $B_c^{1/2}s \in \text{span}\{x_1, x_n\}$. The secant equation (2.10) now implies that $H_c^{1/2}y$ is in this span as well. Therefore (2.11) and Corollary 2.1 implies that B is in the Broyden class.

If $v = 0$, then the Lagrange equation now implies that either $\bar{\lambda}_1'$ is rank-one, so $B_c^{1/2}s$ and $H_c^{1/2}y$ are in $\text{span}\{x_1\}$, i.e., are linearly dependent, or it is rank-two and the eigenvalue $\bar{\lambda}_1 = \bar{\lambda}_2 = 1$. In either case we can still apply Corollary 2.1 and we have the SR1 update. The same argument holds if $w = 0$.

Therefore, it only remains to show that B is in the efficient part of the Broyden class. If $b = c$ then there is nothing to show. If $b < c$, then from Lemma 2.1, the eigenvalues for the SR1 update satisfy $0 < \lambda_-(\phi_{SR1}) < 1 = \lambda_+(\phi_{SR1})$. If $\lambda_n < \lambda_-(\phi_{SR1})$, then we have $B_{\phi_{SR1}} \prec B$, which is a contradiction. Therefore

$$\lambda_n \geq \lambda_-(\phi_{SR1}), \tag{2.16}$$

i.e., $B = B_\phi$ with $\phi \in \Phi$, since $\lambda_-(\phi)$ is isotonic with $-\phi$, by Lemma 2.1.

The case $b > c$ follows similarly except we use the fact that B is optimal for the following problem

$$\begin{aligned} \max \quad & \lambda_n(\bar{B}) \\ \text{subject to} \quad & \bar{B}B_c^{\frac{1}{2}}s = H_c^{\frac{1}{2}}y, \\ & \lambda_1(\bar{B}) \leq \lambda_1(B), \\ & \bar{B} = \bar{B}^t. \end{aligned}$$

Now we have shown that any efficient point B must be in the Broyden efficient class, Φ. Conversely, for any update B_ϕ in the Broyden efficient class, $\lambda_\pm(\phi)$ are isotonic with $-\phi$ and (2.11) holds. Thus B_ϕ cannot be dominated by any other updates in the Broyden efficient class, Φ. Moreover, we can restrict ourselves to Φ since any update can be dominated by an efficient update by Lemma 2.3. □

Note that (2.7) is a relaxation of (2.6), which involves an infinite number of functions. One way of handling (2.6) directly is to use the volume of the ellipsoids corresponding to the quadratic forms. This is the approach in [25].

2.2. Inverse Broyden efficient class. The inverse Broyden class updates can be parametrized by

$$H_{\hat\phi} = H_c - \frac{1}{y^t H_c y} H_c y y^t H_c + \frac{1}{y^t s} s s^t + (1-\hat\phi) y^t H_c y v v^t,$$

where

$$v = \frac{1}{y^t s} s - \frac{1}{y^t H_c y} H_c y.$$

The BFGS and DFP updates correspond to $\hat\phi = 0, 1$, respectively; while the SR1 update is $\hat\phi_{SR1} = -a/b - a$. In general,

$$\hat\phi = \iota(\phi) = \frac{1-\phi}{1+\phi[\frac{b^2}{ac} - 1]} \tag{2.17}$$

is a 1–1 and onto mapping (c.f. [23]) that relates ϕ and $\hat\phi$ for which B_ϕ is spd and $B_\phi^{-1} = H_{\hat\phi}$. Now consider the inverse multi-objective least change problem

$$\begin{aligned} \min \quad & |\hat\lambda_i(B_c^{\frac12} H_+ B_c^{\frac12}) - 1|,\ i = 1, \ldots, n; \\ & \text{subject to } B_+ s = y,\ B_+ = B_+^t. \end{aligned} \tag{2.18}$$

Let

$$\hat\Phi = \begin{cases} [\hat\phi_{SR1}, \infty) & \text{if } b > a, \\ (-\infty, \hat\phi_{SR1}] & \text{if } b < a, \\ (-\infty, \infty) & \text{if } b = a. \end{cases} \tag{2.19}$$

We let $\hat\Phi$ represent the subset of the Broyden class of updates $\{B_{\hat\phi},\ \hat\phi \in \hat\Phi\}$, and call it the *inverse Broyden efficient class.* From the previous section, after exchanging the roles of B_+, B_c and s with H_+, H_c and y, respectively, we get the following corollary.

COROLLARY 2.2. *The efficient class for problem* (2.18) *consists of the inverse Broyden class with* $\hat\phi \in \hat\Phi$.

2.3. Proper efficiency. As mentioned above, an efficient point that is not proper efficient is not desirable, since we can obtain an arbitrarily large marginal improvement. However, based on the results of Theorem 2.1, we can present the following theorem.

THEOREM 2.2. *The Broyden efficient class* Φ *for problem* (2.8) *is also the properly efficient class for problem* (2.8). *The same holds true for the inverse efficient class* $\hat\Phi$ *and the problem* (2.18).

Proof. Let B_{ϕ_0} be an update in the Broyden efficient class. Without loss of generality and for simplicity of notation, we can assume that $B_c = I$. By abuse of notation, let

$$h_i(\phi_0) = |\lambda_i(B_{\phi_0}) - 1|, \qquad i = 1, \dots, n,$$

and let $h_i(+)$ denote $|\lambda_i(B_+) - 1|$. Then

$$h_i(\phi_0) = |\lambda_i(B_{\phi_0}) - 1| = 0 \quad \text{for } i = 2, \dots, n-1,$$

and so, to show that B_{ϕ_0} is proper efficient, it is sufficient to only consider the two cases $i = 1$ and $i = n$. In addition, if $ac = b^2$, the entire Broyden class reduces to the SR1 update. Therefore every update is dominated by the SR1 update which must therefore be proper efficient. So we can also assume that $ac > b^2$.

Let B_+ be a symmetric update satisfying the secant equation such that

$$h_1(+) < h_1(\phi_0). \tag{2.20}$$

As in the proof of Theorem 2.1, we can use Lemma 2.2 and find a Broyden efficient class update B_{ϕ_1} such that

$$h_i(\phi_1) \le h_i(+), \quad i = 1, n. \tag{2.21}$$

Therefore, from (2.20) we have

$$h_1(\phi_1) < h_1(\phi_0),$$

which implies that $\lambda_1(B_{\phi_1}) < \lambda_1(B_{\phi_0})$. Since Lemma 2.1 implies that $\lambda_\pm(\phi)$ are isotonic with $-\phi$, we have $\phi_1 > \phi_0$ and hence

$$\lambda_n(B_{\phi_1}) < \lambda_n(B_{\phi_0}),$$

which is equivalent to

$$h_n(\phi_1) > h_n(\phi_0). \tag{2.22}$$

Therefore, from (2.20), (2.21), and (2.22) we have

$$\begin{aligned} h_1(\phi_1) &\le h_1(+) < h_1(\phi_0), \\ h_n(+) &\ge h_n(\phi_1) > h_n(\phi_0). \end{aligned}$$

This together with Cauchy's mean value theorem yields

$$\frac{h_1(\phi_0) - h_1(+)}{h_n(+) - h_n(\phi_0)} \le \frac{h_1(\phi_0) - h_1(\phi_1)}{h_n(\phi_1) - h_n(\phi_0)} = \frac{\lambda_+(\phi_0) - \lambda_+(\phi_1)}{\lambda_-(\phi_0) - \lambda_-(\phi_1)} = \frac{\lambda'_+(\bar\phi)}{\lambda'_-(\bar\phi)}, \tag{2.23}$$

for some $\phi_0 < \bar\phi < \phi_1$ with $\lambda'_-(\bar\phi) \neq 0$.

From Chapter 7 in [21] or Lemma 2.1, we have

$$\lambda'_\pm(\phi) = \frac{-(ac - b^2)}{2b^2}(1 \pm g(\phi)),$$

where

$$g(\phi) = \frac{(f_1(\phi) - \frac{b}{c})}{[(f_1(\phi) - \frac{b}{c})^2 + \frac{ac-b^2}{c^2}]^{\frac{1}{2}}} < 1.$$

Let

$$h(\phi) = \frac{\lambda'_+(\phi)}{\lambda'_-(\phi)} = \frac{1+g(\phi)}{1-g(\phi)}. \tag{2.24}$$

Then

$$h'(\phi) = \frac{\partial h}{\partial g}\frac{\partial g}{\partial f_1}\frac{\partial f_1}{\partial \phi} = \frac{2}{(1-g)^2}\,\frac{ac-b^2}{((f_1(\phi)-\frac{b}{c})^2+\frac{ac-b^2}{c^2})^{\frac{3}{2}}}\,\frac{-(ac-b^2)}{2b^2} < 0,$$

i.e., we obtain the upper bound for gain over loss

$$\frac{\lambda'_+(\bar{\phi})}{\lambda'_-(\bar{\phi})} \le \frac{\lambda'_+(\phi_0)}{\lambda'_-(\phi_0)} = \frac{1+g(\phi_0)}{1-g(\phi_0)}. \tag{2.25}$$

For the case of $i = n$ in the definition of proper efficient, a similar argument yields the upper bound $1 - g(\phi_0)/1 + g(\phi_0)$. Therefore, B_{ϕ_0} is a properly efficient solution of problem (2.8). The proof for the inverse efficient class follows similarly. □

3. Self-scaling efficient region. The above Broyden efficient class Φ of updates does not contain many important updates that have been studied in the literature, e.g., the Oren–Luenberger self-scaling updates to which we refer as sized updates. We now relax the multicriteria problem (2.8) by allowing sizing of B_c by t and adding the function $|t-1|$. The relaxation attempts to have all the eigenvalues close to a constant, where the constant is close to 1. (The constant was equal to 1 in the first multicriteria problem.) We then see that we get an efficient region that contains all the classes mentioned so far. The relaxation yields the following problem.

$$\min_{t,B_+} \begin{cases} |\lambda_i((tB_c)^{-\frac{1}{2}}B_+(tB_c)^{-\frac{1}{2}}) - 1|, & i = 1,\dots,n, \\ |t-1| \end{cases} \tag{3.1}$$
$$\text{subject to } B_+ s = y, \quad B_+ = B_+^t, \quad t > 0.$$

The self-scaling Broyden class is

$$B_+(t,\phi) = t\left(B_c - \frac{1}{s^tB_cs}B_css^tB_c + (1-\phi)s^tB_csww^t\right) + \frac{1}{y^ts}yy^t,$$

where $w = (1/y^ts)y - (1/s^tB_cs)B_cs$ and $t \in \Re$. This includes the sized DFP with $(t,\phi) = (\frac{b}{c}, 0)$ and the inverse sized BFGS with $(t,\phi) = (\frac{a}{b}, 1)$. If $ac > b^2$, then we define the region

$$\Phi_R^{\geq} = \left\{ (t,\phi): \quad \begin{matrix} \min(1,\alpha_-) \le t \\ t \le \max(1,\alpha_+) \end{matrix} \quad \text{and} \quad \begin{pmatrix} \frac{ac}{ac-b^2}\frac{t-\frac{b}{c}}{t} \le \phi & \text{if } 1 < t \\ \phi \le \frac{ac}{ac-b^2}\frac{t-\frac{b}{c}}{t} & \text{if } t < 1 \\ \phi \le \frac{t}{t-\frac{b}{c}} & \text{if } \frac{b}{c} < t \\ \frac{t}{t-\frac{b}{c}} \le \phi & \text{if } t < \frac{b}{c} \\ \phi \text{ arbitrary} & \text{if } t = \frac{b}{c} = 1 \end{pmatrix} \right\}, \tag{3.2}$$

where

$$\alpha_\pm = \frac{a}{b} \pm \left\{\frac{a^2}{b^2} - \frac{a}{c}\right\}^{\frac{1}{2}}. \tag{3.3}$$

If $ac = b^2$, then we define the region

$$\Phi_R^{=} = \left\{(t,\phi) : \min\left(1, \frac{b}{c}\right) \le t \le \max\left(1, \frac{b}{c}\right),\ \phi \text{ arbitrary}\right\}. \tag{3.4}$$

We let Φ_R represent the appropriate region determined by the value of $ac - b^2$, i.e., it represents the sized Broyden class updates $B_+(t,\phi)$, $(t,\phi) \in \Phi_R$,

$$\Phi_R = \begin{cases} \Phi_R^{>} & \text{if } ac > b^2, \\ \Phi_R^{=} & \text{if } ac = b^2. \end{cases}$$

We call Φ_R the SSER. The above representation illustrates that the efficient region lies between the two curves:

$$\phi = \frac{ac}{ac - b^2}\frac{t - \frac{b}{c}}{t}; \qquad \phi = \frac{t}{t - \frac{b}{c}}.$$

We also define the curve

$$\Phi_C = \left\{(t,\phi) :\ \alpha_- \le t \le \alpha_+ \text{ and } \begin{pmatrix} \phi = \frac{ac}{ac-b^2}\frac{t-\frac{b}{c}}{t} & \text{if } ac - b^2 > 0 \\ \phi \text{ arbitrary} & \text{if } ac - b^2 = 0 \end{pmatrix}\right\}, \tag{3.5}$$

and call Φ_C the *self-scaling efficient curve.* Note that this curve contains optimally conditioned updates, i.e., updates optimal for the κ measure; see [2], [25], [18]. (See Figs. 1 and 2 for illustrations of the various efficient sets.) We now state and prove our main results.

THEOREM 3.1. *The efficient updates for problem* (3.1) *are the* SSER *updates* $B_+(t,\phi)$, *with* $(t,\phi) \in \Phi_R$.

Before we prove the above theorem, we present the following preliminary results.

LEMMA 3.1. *The matrix* $B = (tB_c)^{-1/2}B_+(t,\phi)(tB_c)^{-1/2}$ *has* $n-2$ *unit eigenvalues and the two remaining eigenvalues are*

$$\lambda_\pm(t,\phi) = f_1(t,\phi) \pm (f_1(t,\phi)^2 - f_2(t,\phi))^{\frac{1}{2}},$$

where

$$f_1(t,\phi) = \tfrac{a}{2bt} + \tfrac{ac - \phi(ac-b^2)}{2b^2}$$
$$f_2(t,\phi) = \tfrac{1}{t}\left(\tfrac{a}{b} - \tfrac{\phi(ac-b^2)}{bc}\right).$$

Furthermore,

(i) $\lambda_\pm(t,\phi)$ *are isotonic with* $-\phi$ *for any fixed* t, *and*

$$\frac{\partial\lambda_\pm(t,\phi)}{\partial\phi} = \frac{-(ac-b^2)}{2b^2}[1 \pm g(t,\phi)],$$

where

$$g(t,\phi) = \frac{(f_1(t,\phi) - \frac{b}{tc})}{[(f_1(t,\phi) - \frac{b}{tc})^2 + \frac{ac-b^2}{t^2c^2}]^{\frac{1}{2}}},$$

and $|g(t,\phi)| \le 1$;

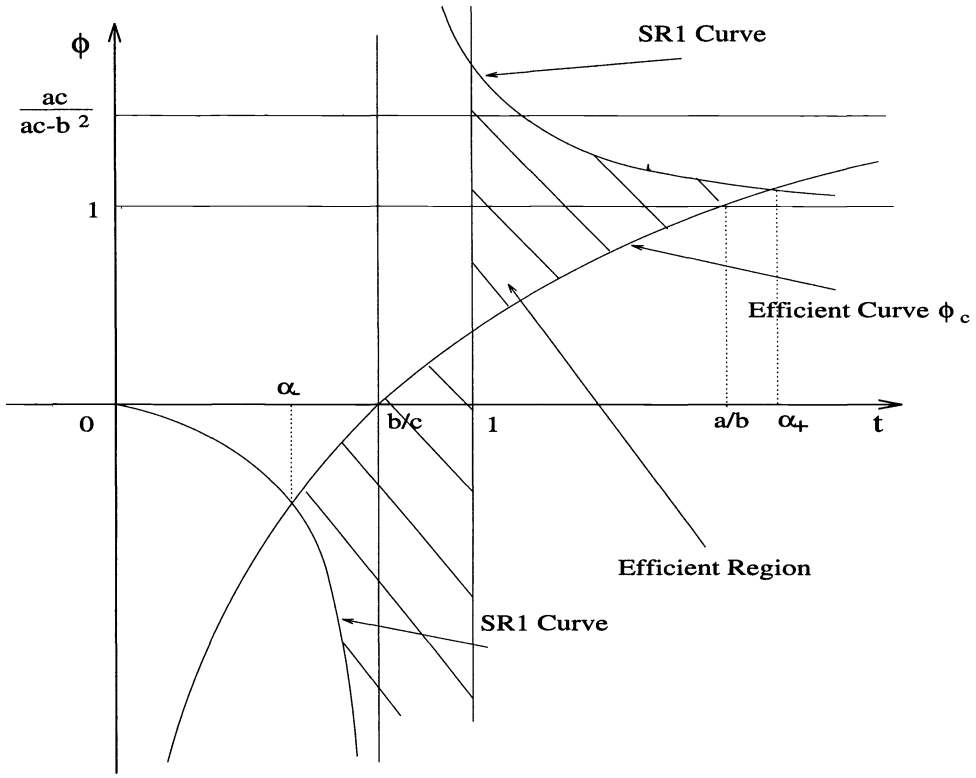


FIG. 1. *Efficient region for* $\frac{b}{c} < 1$.

(ii) $\lambda_{\pm}(t,\phi)$ *are isotonic with* $-t$ *for any fixed* ϕ, *and*

$$\frac{\partial \lambda_{\pm}(t,\phi)}{\partial t} = -\frac{a}{2bt^2}(1 \pm d(t,\phi)), \tag{3.6}$$

where

$$d(t,\phi) = \frac{f_1(t,\phi) - \frac{tb}{a} f_2(t,\phi)}{(f_1^2(t,\phi) - f_2(t,\phi))^{\frac{1}{2}}},$$

and $|d(t,\phi)| \leq 1$.

Proof. Note that a becomes a/t and c becomes ct when B_c is sized with t. The results in (i) are straightforward extensions of those in [21]; see, also, Lemma 2.1. Here we only prove (ii). Differentiating $\lambda_{\pm}(t,\phi)$ yields (3.6). Now, to show that $\lambda_{\pm}(t,\phi)$ are isotonic with $-t$, we need only show that $|d(t,\phi)| \leq 1$. By squaring both sides, we have

$$\left(f_1(t,\phi) - \frac{b}{a} f_2(t,\phi)\right)^2 \leq f_1^2(t,\phi) - f_2(t,\phi),$$

which is equivalent to

$$\left(1 - \frac{\phi(ac - b^2)}{ac}\right)^2 \left(1 - \frac{ac}{b^2}\right) \leq 0.$$

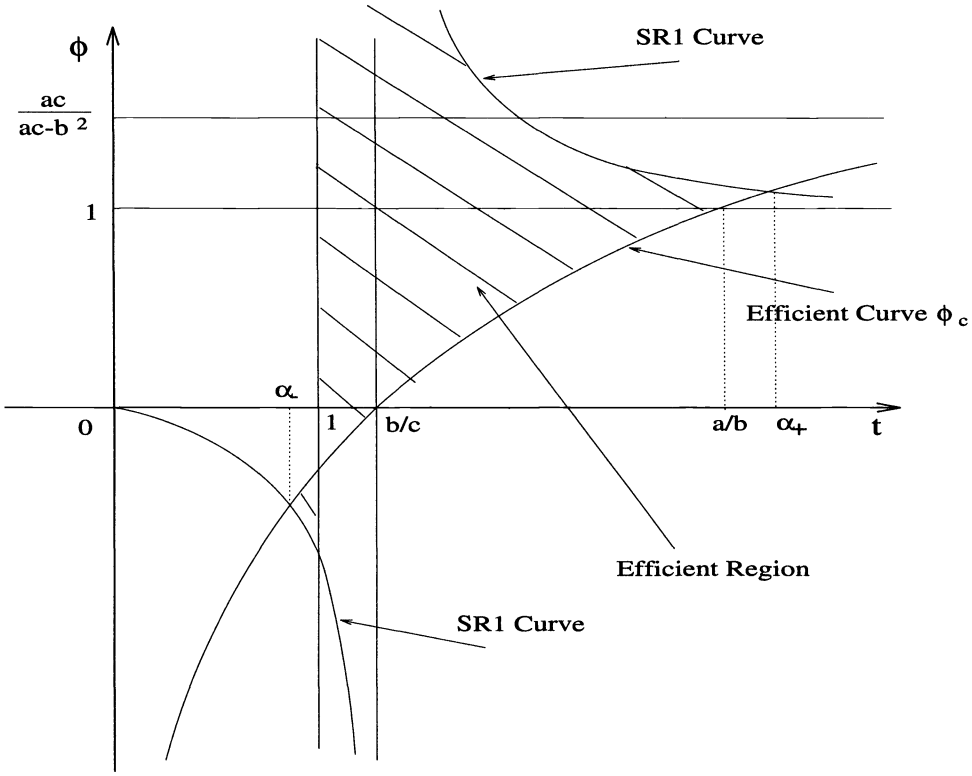


FIG. 2. *Efficient region for* $\frac{b}{c} > 1$.

Therefore, we have

$$\begin{aligned} d(t,\phi) &= 1 \quad \text{if } \phi = \tfrac{ac}{ac-b^2}, \\ d(t,\phi) &< 1 \quad \text{otherwise.} \end{aligned} \qquad \square$$

Note that the derivatives may not exist in the case of multiple eigenvalues. In this case, they are subdifferentiable, since the scaled update $B(t,\phi)$ is a linear function of ϕ and $1/t$ and so the largest and smallest eigenvalues are convex and concave functions, respectively, of ϕ and $1/t$.

LEMMA 3.2. *Suppose that* $ac - b^2 > 0$ *and* $(t_1, \phi_1) \in \Phi_R$. *Then the level curves for* $\lambda_\pm(t,\phi)$ *passing through* (t_1,ϕ_1) *are*

$$\phi = \frac{\lambda_+(t_1,\phi_1)b}{\lambda_+(t_1,\phi_1)ct - b} + \frac{ac - \lambda_+(t_1,\phi_1)b^2}{ac - b^2} \quad \textit{for } t > \frac{b}{\lambda_+(t_1,\phi_1)c}$$

and

$$\phi = \frac{\lambda_-(t_1,\phi_1)b}{\lambda_-(t_1,\phi_1)ct - b} + \frac{ac - \lambda_-(t_1,\phi_1)b^2}{ac - b^2} \quad \textit{for } t < \frac{b}{\lambda_-(t_1,\phi_1)c},$$

respectively.

Proof. Let (t,ϕ) be on the level curve for $\lambda_+(t,\phi)$ passing through (t_1,ϕ_1). Then

$$(\lambda_+(t_1,\phi_1) - f_1(t,\phi))^2 = f_1^2(t,\phi) - f_2(t,\phi),$$

which is equivalent to

$$\lambda_+^2(t_1,\phi_1) - 2\lambda_+(t_1,\phi_1)f_1(t,\phi) = -f_2(t,\phi).$$

By putting the expressions for $f_1(t,\phi)$ and $f_2(t,\phi)$ into the above equality, we have

$$\begin{aligned}
&\phi(ac-b^2)(\lambda_+(t_1,\phi_1)ct-b)\\
&= abc\lambda_+(t_1,\phi_1) + \lambda_+(t_1,\phi_1)ac^2t - \lambda_+^2(t_1,\phi_1)b^2ct - abc\\
&= (ac-b^2)\lambda_+(t_1,\phi_1)b + \lambda_+(t_1,\phi_1)ac^2t - \lambda_+^2(t_1,\phi_1)b^2ct - abc + \lambda_+(t_1,\phi_1)b^3\\
&= (ac-b^2)\lambda_+(t_1,\phi_1)b + (\lambda_+(t_1,\phi_1)ct-b)(ac-\lambda_+(t_1,\phi_1)b^2),
\end{aligned}$$

i.e.,

$$\phi = \frac{\lambda_+(t_1,\phi_1)b}{\lambda_+(t_1,\phi_1)ct-b} + \frac{ac-\lambda_+(t_1,\phi_1)b^2}{ac-b^2}.$$

Moreover, the Raleigh quotient implies that

$$\lambda_+(t_1,\phi_1) \geq \frac{((tB_c)^{\frac{1}{2}}s)^t(tB_c)^{-\frac{1}{2}}B_+(t,\phi)(tB_c)^{-\frac{1}{2}})(tB_c)^{\frac{1}{2}}s}{||(tB_c)^{\frac{1}{2}}s)||^2} = \frac{b}{tc}.$$

Therefore, the level curve is the branch for $t > (b/\lambda_+(t_1,\phi_1)c)$.

Conversely, for any (t,ϕ) on the curve

$$\phi = \frac{\lambda_+(t_1,\phi_1)b}{\lambda_+(t_1,\phi_1)ct-b} + \frac{ac-\lambda_+(t_1,\phi_1)b^2}{ac-b^2}, \tag{3.7}$$

where $t > \frac{b}{\lambda_+(t_1,\phi_1)c}$, we have

$$|\lambda_+(t_1,\phi_1) - f_1(t,\phi)| = (f_1^2(t,\phi) - f_2(t,\phi))^{\frac{1}{2}}. \tag{3.8}$$

Now $\lambda_+(t_1,\phi_1) - f_1(t,\phi) > 0$ is equivalent to

$$\frac{\frac{ab}{t} + ac - \phi(ac-b^2)}{2b^2} < \lambda_+(t_1,\phi_1). \tag{3.9}$$

After substituting for ϕ using (3.7), and noting that $t > (b/\lambda_+(t_1,\phi_1)c)$, this is equivalent to $\lambda_+^2(t_1,\phi_1)ct^2 - 2\lambda_+(t_1,\phi_1)bt + a > 0$. This further reduces to $4\lambda_+^2(t_1,\phi_1)(b^2 - ac) < 0$, which clearly holds. Therefore (3.8) becomes

$$\lambda_+(t_1,\phi_1) - f_1(t,\phi) = (f_1^2(t,\phi) - f_2(t,\phi))^{\frac{1}{2}},$$

which yields $\lambda_+(t,\phi) = \lambda_+(t_1,\phi_1)$.

By a similar argument we obtain the level curve for λ_-. □

We now prove the above theorem.

Proof. From Theorem 2.1, for any fixed $t > 0$, we need only consider sized Broyden class updates in the set

$$\Phi_t = \left\{ \phi : \begin{array}{ll} \phi \geq \frac{ct}{ct-b} & \text{if } t < \frac{b}{c} \\ \phi \leq \frac{ct}{ct-b} & \text{if } t > \frac{b}{c} \\ \phi \text{ arbitrary} & \text{if } t = \frac{b}{c} \end{array} \right\}.$$

First suppose that $ac-b^2 > 0$. If we are given a scaled update $B(t,\phi)$ with $\phi \in \Phi_t$, then to check efficiency we need to see if the update can be improved with respect to the $n+1$ functions in (3.1). From the proof of Theorem 2.1, we know that (2.11) holds, i.e., $\lambda_2 = \lambda_{n-1} = 1$. Therefore, we need only consider the largest and smallest eigenvalues, $\lambda_\pm$. Moreover, if $r = 1/t$, then the scaled update is linear in (r,ϕ), which implies that the largest and smallest eigenvalues $\lambda_\pm$ are convex and concave functions, respectively, of (r,ϕ). Therefore directions of descent correspond to negative directional derivatives. Thus it is easier to view the functions in the space (r,ϕ), which we do.

First suppose that an efficient point (r,ϕ) is given and that $r < 1$ (or equivalently $t > 1$) and $\lambda_1 > 1 > \lambda_n$. If we hold r fixed, then we obtain $\phi \in \Phi_t$ by Theorem 2.1. Otherwise, at the point (r,ϕ), consider moving in the direction $e = (1\ x)^t \in \Re^2$. Then there is a gain in r and so efficiency implies that we cannot have a gain in both $\lambda_\pm$ as well, i.e., from Lemma 3.1, the directional derivatives (in (r,ϕ) space) cannot satisfy

$$\begin{aligned} \nabla\lambda_+^t e &= \tfrac{a}{2b}(1+d) - x\tfrac{ac-b^2}{2b^2}(1+g) < 0, \\ \nabla\lambda_-^t e &= \tfrac{a}{2b}(1-d) - x\tfrac{ac-b^2}{2b^2}(1-g) > 0. \end{aligned} \tag{3.10}$$

(Note that we have used

$$\frac{\partial\lambda_\pm(r,\phi)}{\partial r} = \frac{\partial\lambda_\pm(t,\phi)}{\partial t}\frac{-1}{r^2},$$

since $t = \frac{1}{r}$.) This reduces to

$$\frac{1+d}{1+g} < \frac{ac-b^2}{2b^2}\frac{2b}{a}x < \frac{1-d}{1-g},$$

which is feasible for x if and only if $d < g$ or equivalently

$$\phi < \frac{ac}{ac-b^2}\frac{t-\frac{b}{c}}{t},$$

i.e., if $t > 1$ and $\lambda_+ > 1 > \lambda_-$, then efficiency implies

$$\phi \geq \frac{ac}{ac-b^2}\frac{t-\frac{b}{c}}{t}.$$

If $\lambda_+ = 1$ (respectively, $\lambda_- = 1$), then the $<$ (respectively, $>$) is replaced by $\leq$ (respectively, $\geq$) in (3.10). Moreover, $ac - b^2 > 0$ implies that $\lambda_+ > \lambda_-$. Thus $(\phi,t) \in \Phi_R$.

The conclusions for $t < 1$ are similar, i.e., finding a direction of improvement implies that $d > g$ or

$$\phi > \frac{ac}{ac-b^2}\frac{t-\frac{b}{c}}{t}.$$

Therefore, if $t < 1$, then efficiency implies

$$\phi \leq \frac{ac}{ac-b^2}\frac{t-\frac{b}{c}}{t}.$$

The case for $t = 1$ follows from Theorem 2.1. The above argument for efficiency is reversible, since the functions $\lambda_\pm$ are convex and concave, respectively.

If $ac = b^2$, then for each fixed t, the scaled-sized Broyden class reduces to the scaled-sized SR1 with $n-1$ unit eigenvalues and the remaining eigenvalue given by

$$1 + \frac{(1 - \frac{c}{b}t)a}{tb}.$$

Therefore we need only solve the multicriteria problem with the two functions

$$\left|\frac{(\frac{1}{t} - \frac{c}{b})a}{b}\right|, \qquad |t-1|.$$

This completes the proof that Φ_R is the efficient region. □

THEOREM 3.2. *The proper efficient updates for problem* (3.1) *are the efficient region updates.*

Proof. To prove proper efficiency, we need to consider several cases arising from the definition of the efficient region. Suppose that $(t, \phi) \in \Phi_R$ is given. As in the proof of Theorem 3.1, we continue to let: (t, ϕ) represent the update $B_+(t, \phi)$; and we let $\lambda_\pm$, or $\lambda_{1,n}$, represent the largest and smallest eigenvalues, respectively, of the corresponding scaled update. As in the proof of Theorem 2.2, we can restrict ourselves to $i = 1, n$ and to efficient updates. But, we need to consider the additional objective function $|t-1|$. Moreover, we let $(\bar{t}, \bar{\phi}) \in \Phi_R$ be a given second efficient point, and consider the points and functions in (r, ϕ) space, where $r = 1/t$. (So that $\lambda_\pm$ are convex and concave functions, respectively.)

By efficiency, we know that $\lambda_1 = \lambda_+ \geq 1 \geq \lambda_- = \lambda_n$. Therefore, a gain (respectively, loss) in the first function corresponds to $\lambda_+ > \bar{\lambda}_+ \geq 1$ (respectively, $\bar{\lambda}_+ > \lambda_+ \geq 1$). Similarly, a gain (respectively, loss) in the n-th function corresponds to $\lambda_- < \bar{\lambda}_- \leq 1$ (respectively, $\bar{\lambda}_- < \lambda_- \leq 1$). However, a gain or loss for the last function is not as simple, e.g., if $r < 1$, then a gain (respectively, loss) can be $r < \bar{r} \leq 1$ or $r \leq 1 \leq \bar{r} < 2 - r$, (respectively, $\bar{r} < r \leq 1$ or $r \leq 1 \leq 2 - r < \bar{r}$). We must treat several different cases corresponding to different choices for the numerator and denominator in the gain over loss ratio. We prove the first few cases directly by finding an upper bound to the gain over loss ratio. We then complete the remaining cases by using a proof by contradiction.

Case 1. Suppose that we measure a gain in the first function for $(\bar{r}, \bar{\phi})$; thus $\bar{\lambda}_1 < \lambda_1$ since efficiency implies $\lambda_1 \geq 1 \geq \lambda_n$. By efficiency of (r, ϕ) we must have a loss in r or in λ_n.

Subcase 1.1. Suppose there is no loss in r and so there is a loss $\bar{\lambda}_n < \lambda_n$. If r stays constant, then a bound for the gain/loss ratio follows by applying Theorem 2.2 to the sized update. Otherwise, the direction between the two points must be $e = (\pm 1 \; x)^t$, for some x.

Subcase 1.1.1. Suppose first that $r < 1$ so that the direction is $e = (1 \; x)^t$. Since $\lambda_\pm$ are convex and concave functions, the ratio of directional derivatives at (r, ϕ), in the direction $-e$, provides an upper bound on gain over loss, i.e.,

$$\frac{\lambda_+ - \bar{\lambda}_+}{\lambda_- - \bar{\lambda}_-} \leq \frac{\nabla \lambda_+^t(-e)}{\nabla \lambda_-^t(-e)}. \tag{3.11}$$

If we differentiate this ratio of directional derivatives, we get

$$\frac{\partial}{\partial x}\left(\frac{\frac{a}{2b}(1+d) - x\frac{ac-b^2}{2b^2}(1+g)}{\frac{a}{2b}(1-d) - x\frac{ac-b^2}{2b^2}(1-g)}\right) = K(d-g)$$

for a positive constant K. We have seen that when $t > 1$, efficiency means that $d - g \geq 0$. Therefore, we need to increase x to infinity to get the largest gain over loss ratio, i.e., this means that t stays constant and we can apply the result in Theorem 2.2 to the sized update and get a bound that depends only on (t, ϕ). In particular, the right-hand side in (3.11) yields $1 + g/1 - g$ as $x \to \infty$. (See, also, (2.25).)

Subcase 1.1.2. If $r = 1$, then we cannot have an improvement in r and the direction must keep r constant if there is no loss in r. We can again apply Theorem 2.2.

Subcase 1.1.3. If $r > 1$ and there is no loss in r, then the direction must be $e = (-1\ x)^t$. Differentiating the ratio of directional derivatives now yields $-(d-g)$ up to a positive constant. Efficiency in the case $t < 1$ now implies that this derivative is nonnegative and so we again increase x to infinity, i.e., t stays constant and we apply Theorem 2.2 to obtain the upper bound for the gain over loss ratio as in Subcase 1.1.1.

Subcase 1.2. The above presents a direct proof of proper efficiency in several cases by providing upper bounds for the gain over loss ratio. The remainder of the proof is by contradiction. First, suppose that (r, ϕ) is efficient but not proper. Then there is a sequence of efficient points (r_k, ϕ_k) such that the corresponding gain over loss ratio goes to infinity. Let $e_k = ((s_k, \phi_k) - (s, \phi))/||(s_k, \phi_k) - (s, \phi)||$ be the normalized direction between the points. By choosing an appropriate subsequence, we can assume that $e_k \to e$. Then the gradient of the loss functions must be orthogonal to e. For example, if the ratio is a gain for λ_+ over a loss for r, with $r_k < r \leq 1$, then

$$\frac{\lambda_+ - \lambda_+^k}{\frac{1}{r_k} - \frac{1}{r}} \leq \frac{\nabla\lambda_+^t(-e_k)}{(\frac{1}{r^2}\ 0)(-e_k)}.$$

Since the left-hand side goes to infinity, and $e_k \to e$, we conclude that e is orthogonal to $(1/r^2\ 0)^t$ and $\nabla\lambda_+^t(-e) \geq 0$, which means $e = (0, 1)^t$. But then e is a direction of decrease, or loss, for λ_-, so that the bound for the gain over loss ratio should use λ_-. Moreover, $\nabla\lambda_-^t e \neq 0$, a contradiction to the ratio being unbounded. We have therefore proven Subcase 1.2, where there is a gain in λ_+, a loss in r, and $r_k < r \leq 1$. To complete this special case we have to consider $r \leq 1 \leq 2 - r < r_k$. This is covered in the case $r > 0$ when we replace r by $2 - r$, since $|r - 1| = |2 - r - 1|$.

Case 2. Where there is a gain in λ_-, this case follows similarly. Note that the gain over loss is $\bar{\lambda}_- - \lambda_-/\bar{\lambda} + -\lambda_+$, which is a concave over a convex function.

Case 3. If we measure a gain in r, then there is a loss in λ_+ (or λ_-). As above, there is a direction $e = (\pm 1\ x)$ such that e is orthogonal to $\nabla\lambda_+$. If $r < 1$, then $e = (1\ x)$ and

$$x = \frac{a}{2b}\frac{2b^2}{ac - b^2}\frac{1+d}{1+g} \text{ so that } \nabla\lambda_-^t e = \frac{a}{2b}(1-d) - \frac{a}{2b}\frac{1+d}{1+g}(1-g) = \frac{a}{2b}2(g-d) < 0,$$

i.e., e is a direction of decrease or loss for λ_- and $\nabla\lambda_-^t e \neq 0$, a contradiction. Therefore, the ratio is bounded again. The other cases follow similarly.

The ratio of gain in t over loss in one of the eigenvalues follows similarly, as does the converse, upon noting that the function in t is linear. □

COROLLARY 3.1. *The efficient updates for problem* (3.1) *without the function* $|t - 1|$ *are the self-scaling efficient curve updates, i.e., the sized Broyden class updates* $B_+(t, \phi)$, *with* $(t, \phi) \in \Phi_C$.

Proof. The proof follows from the proof of the above theorem by combining the two cases $t > 1$ and $t < 1$. □

As was the case for the Broyden efficient class of updates, there is a corresponding inverse efficient region that can be found by scaling the inverse updates and exchanging roles appropriately.

4. Conclusion. We have used multicriteria objectives, based on the eigenvalues of the scaled Hessian, to find an efficient region of updates. We have shown that this region contains the well-known updates used to date. Since it is generally acknowledged that the scaled eigenvalues are the determining factor in selecting good updates, we see that we have found a general framework for deriving good updates. Moreover, we have shown that the efficient region is proper efficient.

This region does *not* certify that each update in it is better than all the updates outside the region. However, it *does* guarantee that, for each update outside, there is an update inside that is better. The DM can now chose between different efficient updates; see, e.g., [19].

Known results from the theory of multicriteria optimization can now be applied to quasi-Newton updates. For example, we can use the characterization of proper efficiency [15] to show that, for a given efficient update, there corresponds weights w_i, $i = 1, \ldots, n+1$ such that the update is the optimal solution of the *single* objective optimization problem

$$\begin{array}{ll} \min_{t,B_+} & \sum_i^n w_i|\lambda_i((tB_c)^{-\frac{1}{2}}B_+(tB_c)^{-\frac{1}{2}}) - 1| + w_{n+1}|t-1| \\ \text{subject to} & B_+s = y,\ B_+ = B_+^t,\ \ t > 0. \end{array} \tag{4.1}$$

This provides new utility functions for deriving updates. Furthermore, by calculating the weights, we can find the relative importance that different updates assign to different eigenvalues.

Up until now, our derivation of the updates has not taken into consideration modern techniques for finding search directions and stepsizes, e.g., using trust regions and inexact line searches. However, the recent success of the selective sizing updates in a trust region framework (Contreras and Tapia [6]) suggests that these techniques should not be ignored. In both the line search and trust region techniques, lower and upper bounds are found for the stepsize to guarantee both sufficient decrease of the objective function, as well as convergence. (We restrict ourselves to the line search algorithm.) Slow convergence can result if guaranteeing sufficient decrease continually forces the stepsize close to its lower bound. For line search algorithms that use only backtracking to guarantee sufficient decrease, it can be advantageous to avoid search directions of small length. This can be done if the eigenvalues of the current Hessian approximation matrix are not too large relative to the eigenvalues of the true Hessian. This is indicated by avoiding $b/c < 1$.

To correct the large eigenvalues, Contreras and Tapia used b/c to size B_c, whenever $b/c < 1$. Byrd, Nocedal, and Yuan [5] showed that the BFGS update can rapidly correct large eigenvalues. This property is diminished as ϕ is decreased in $[0, 1]$. In particular, the DFP update has no such property. Recall that

$$B_+(t, \phi) = tH_c^{\frac{1}{2}}BH_c^{\frac{1}{2}},$$

where B has $n-2$ unit eigenvalues and the two remaining eigenvalues $\lambda_\pm(t, \phi)$ are isotonic with $-\phi$ and $-t$. From this we can see that to decrease the eigenvalues of $B_+(t, \phi)$ we should increase ϕ and t. Moreover, to avoid excess function evaluations caused by an overly large quasi-Newton step, we also must make sure that the eigenvalues are not too small. The efficient region provides a balance among these multiple objectives.

TABLE 1

Methods	1	2	3	4	5	6
Iterations	1.1748	0.9840	0.9577	0.9583	0.9984	0.9267
Fn. eval.	1.1642	1.0655	0.9281	0.9387	0.9662	0.9372
Fails	4.0833	1.3333	1.0000	1.2500	1.0833	1.4167

TABLE 2

Order of best methods for						
Iterations	6	3	4	2	5	1
Funtion evaluations	3	6	4	5	2	1
Failures	3	5	4	2	6	1

To illustrate how the efficient region works in practice, we choose updates inside, close to the boundary, and far away from the efficient region. For comparison, we also include the BFGS update and the selective sizing update of Contreras and Tapia. (Except for BFGS, all the methods size the Hessian approximation by b/c at the initial iteration. Here $\Phi_C(t)$ is the ϕ value corresponding to t on the efficient curve.) We now compare six updating methods.

(inside region) BFGS.
(inside region) if $b/c < 1$, use sized DFP with b/c; otherwise, use DFP.
(inside region) if $b/c < 1$, use $B_+(t, \phi)$, where t is randomly selected from $[b/c, 1]$ with $\phi = 0.9\Phi_C(t)$ when $t < 0.9$ and $\phi = 1$ when $t \geq 0.9$; otherwise use DFP.
(outside region) if $b/c < 1$, use $B_+(t, \phi)$, where t is randomly selected from $[b/c, 1]$ with $\phi = 1.1\Phi_C(t)$ when $t < 0.9$ and $\phi = 1$ when $t \geq 0.9$; otherwise use DFP.
(inside region) if $b/c < 1$, use $B_+(t, \phi)$, where t is randomly selected from $[b/c, 1]$ with $\phi = 0.5\Phi_C(t)$ when $t < 0.9$ and $\phi = 1$ when $t \geq 0.9$; otherwise use DFP.
(outside region) if $b/c < 1$, use $B_+(t, \phi)$, where t is randomly selected from $[b/c, (1 + b/c)/2]$ with $\phi = 0.5(\Phi_C(t) + 1)$; otherwise use DFP.

We use the standard set of 18 test problems from [14] with initial estimates scaled by: $.1, .5, 1, 2, 3, 4, 5, 6, 7, 8, 10, 20$. The tests were done on a SUN SPARCstation 1 using a MATLAB translation of the codes in [11]. We set 450 as a limit on the number of iterations. Failures were a result of too many iterations. In the case of a failure, we added twice the standard deviation of the successes to the maximum of the successes. We have used the priority theory of Lootsma and Saaty to obtain expected values for iteration and function evaluation counts. (See, e.g., Hock and Schittkowski [17].) The given expected values are relative to a value of 1 for the BFGS method with an unscaled initial point. The numerical results and summary are in Tables 1 and 2.

The best method for expected iterations is Method 6; while the best for function evaluations is Method 3. (Since Method 3 was better for failures, this result is dependent on the penalty we assigned for a failure.) This supports our argument that large quasi-Newton steps are obtained by increasing ϕ or decreasing t and that an overly large quasi-Newton step could result in more function evaluations. The best methods for failures are Method 3 and Method 5. Method 4 also did well. However, Method 6 did badly for failures. This shows that the risk of failures is larger for updates far from the efficient region than for those in the efficient region. The bad performance of BFGS emphasizes the importance of sizing.

REFERENCES

[1] J. BORWEIN, *On the existence of Pareto efficient points*, Math. Oper. Res., 8 (1983), pp. 64–73.

[2] R. H. BYRD, *A multiobjective characterization of the Broyden class*, presented at ORSA/TIMS, New York, Sept. 1990.

[3] R. H. BYRD, D. C. LIU, AND R. H. NOCEDAL, *On the behaviour of Broyden's class of quasi-Newton methods*, SIAM J. Optim., 2 (1992), pp. 553–557.

[4] R. H. BYRD AND R. H. NOCEDAL, *A tool for the analysis of quasi-Newton methods with application to unconstrained minimization*, SIAM J. Numer. Anal., 26 (1989), pp. 727–739.

[5] R. H. BYRD, R. H. NOCEDAL, AND Y. YUAN, *Global convergence of a class of quasi-Newton methods on convex problems*, SIAM J. Numer. Anal., 24 (1987), pp. 1171–1191.

[6] M. CONTRERAS AND R. A. TAPIA, *Sizing the* BFGS *and* DFP *updates: A numerical study*, J. Optim. Theory Appl., 78 (1), 1993.

[7] W. C. DAVIDON, *Variable metric methods for minimization*, Tech. Report ANL-5990, Argonne National Labs, Argonne, IL, 1959.

[8] ———, *Optimally conditioned optimization algorithms without line searches*, Math. Programming, 9 (1975), pp. 1–30.

[9] G. DEBREU, *Theory of Value: An Axiomatic Analysis of Economic Equilibrium*, John Wiley and Sons, New York, 1959.

[10] J. E. DENNIS AND H. WOLKOWICZ, *Sizing and least change secant methods*, SIAM J. Numer. Anal., 10 (1993), pp. 1291–1314.

[11] 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.

[12] L. C. W. DIXON, *On the impact of automatic differentiation on the relative performance of parallel truncated Newton and variable metric algorithms*, SIAM J. Optim., 1 (1991), pp. 475–486.

[13] R. FLETCHER, *A new variational result for quasi-Newton formulae*, SIAM J. Optim., 1 (1991), pp. 18–21.

[14] B. S. GARBOW, K. E. HILLSTROM, AND J. J. MORÉ, *User guide for MINPACK*-1, Tech. Report ANL-80-74, Argonne National Labs, Argonne, IL, 1980; Available from National Technical Information Service, Springfield, VA.

[15] M. GEOFFRION, *Proper efficient and the theory of vector maximization*, J. Math. Anal. Appl., 22 (1968), pp. 618–630.

[16] A. O. GRIEWANK, *On automatic differentiation*, in Mathematical Programming 1988, Kluwer Academic Publishers, Japan, 1988.

[17] W. HOCK AND K. SCHITTKOWSKI, *A comparative performance evaluation of* 27 *nonlinear programming codes*, Computing, 30 (1983), pp. 335–358.

[18] Y. F. HU AND C. STOREY, *A family of optimally conditioned quasi-Newton updates for unconstrained optimization*, Tech. Report, Dept. of Mathematical Sciences, Loughborough University of Technology, Leicestershire, 1991.

[19] V. M. OZERNOY, *Choosing the "best" multiple criteria decision-making method*, INFOR, 30 (1992), pp. 159–171.

[20] R. T. ROCKAFELLAR, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[21] R. B. SCHNABEL, *Analysing and improving quasi-Newton methods for unconstrained optimization*, Ph.D. thesis, Dept of Computer Science, Cornell University, Ithaca, NY, 1977; also available as TR-77-320.

[22] R.E. STEUER, *Multiple Criteria Optimization: Theory, Computation, and Application*, Wiley, New York, Toronto, 1986.

[23] J. STOER, *On the convergence rate of imperfect minimization algorithms in Broyden's β class*, Math. Programming, 9 (1975), pp. 313–335.

[24] L. TUNCEL, *A note on the primal-dual affine scaling algorithms*, Tech. Report TR 1004, Cornell University, Ithaca, NY, 1992.

[25] H. WOLKOWICZ, *Measures for symmetric rank-one updates*, Math. Oper. Res., to appear.

[26] Q. ZHAO, *Measures for least change secant methods*, Master's thesis, University of Waterloo, Waterloo, Ontario, 1992.

SIAM J. OPTIMIZATION
Vol. 5, No. 1, pp. 192–218, February 1995

010

AN OPTIMAL POSITIVE DEFINITE UPDATE FOR SPARSE HESSIAN MATRICES*

R. FLETCHER†

Abstract. A Hessian update is described that preserves sparsity and positive definiteness and satisfies a minimal change property. The update reduces to the BFGS update in the dense case and generalises a recent result in [*SIAM J. Numer. Anal.*, 26 (1989), pp. 727–739] relating to the Byrd and Nocedal measure function. A surprising outcome is that a sparsity projection of the inverse Hessian plays a major role. It is shown that the Hessian itself can be recovered from this information under mild assumptions.

The update is computed by solving a concave programming problem derived by using the Wolfe dual. The Hessian of the dual is important and plays a similar role to the matrix Q that arises in the sparse PSB update of Toint [*Math. Comp.*, 31 (1977), pp. 954–961]. This matrix is shown to satisfy the same structural and definiteness conditions as Toint's matrix. The update has been implemented for tridiagonal systems and some numerical experiments are described. These experiments indicate that there is potential for a significant reduction in the number of quasi-Newton iterations, but that more development is needed to obtain an efficient implementation. Solution of the variational problem by primal methods is also discussed and provides an interesting application of generalized elimination. The possibility of instability and nonexistence of a positive definite update raised by Sorensen [*Math. Programming Study*, 18 (1982), pp. 135–159] is still a difficulty and some remedies are discussed.

Key words. sparse matrix update, positive definite matrix, BFGS formula

AMS subject classifications. 65K, 90C

1. Background and introduction.

This paper primarily relates to quasi-Newton line search methods for finding a solution x^* of the unconstrained optimization problem

$$\text{minimize} \quad f(x) \qquad x \in \mathbb{R}^n. \tag{1.1}$$

These methods generate a sequence of iterates $\{x^{(k)}\}$, $k = 1, 2, \ldots$ by

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} s^{(k)}, \tag{1.2}$$

where $s^{(k)}$ is the current search direction and $\alpha^{(k)}$ is a step chosen to approximately minimize $f(x)$. The methods require the gradient vector $g(x) = \nabla f(x)$ to be available, but not the Hessian matrix $G(x) = \nabla^2 f(x)$. The latter is approximated by a symmetric matrix $B^{(k)}$, which is initially arbitrary and is updated after each iteration. $B^{(k)}$ is used to determine the search direction by solving the system

$$B^{(k)} s^{(k)} = -g^{(k)}, \tag{1.3}$$

where $g^{(k)}$ denotes $g(x^{(k)})$. $B^{(k)}$ is required to be positive definite (written $B^{(k)} > 0$), which implies that $s^{(k)}$ is a descent direction and ensures a reduction in $f(x)$ in the line search.

---

*Received by the editors November 6, 1992; accepted for publication (in revised form) December 14, 1993. This paper was presented at the Scottish Computational Mathematics Symposium, Strathclyde University, September 14, 1992.

†Department of Mathematics and Computer Science, University of Dundee, DD1 4HN, Scotland, United Kingdom (fletcher@mcs.dundee.ac.uk).

An important feature of such methods is the *updating formula* that is used to incorporate new information into $B^{(k)}$. After each iteration difference vectors

$$\begin{aligned} \delta^{(k)} &= x^{(k+1)} - x^{(k)}, \\ \gamma^{(k)} &= g^{(k+1)} - g^{(k)} \end{aligned} \tag{1.4}$$

can be calculated. It follows from the Taylor series that

$$\gamma^{(k)} = \bar{G}^{(k)}\delta^{(k)}, \tag{1.5}$$

where

$$\bar{G}^{(k)} = \int_0^1 G(x^{(k)} + \theta\delta^{(k)})\, d\theta \tag{1.6}$$

is the average Hessian matrix along the step. By analogy with (1.5), $B^{(k+1)}$ is chosen to satisfy

$$\gamma^{(k)} = B^{(k+1)}\delta^{(k)}, \tag{1.7}$$

known as the quasi-Newton condition. The most popular updating formula is the Broyden–Fletcher–Goldfarb–Shanno (BFGS) formula $B^{(k+1)} = \text{bfgs}(B^{(k)}, \delta^{(k)}, \gamma^{(k)})$ where

$$\text{bfgs}(B, \delta, \gamma) = B - \frac{B\delta\delta^T B}{\delta^T B\delta} + \frac{\gamma\gamma^T}{\delta^T\gamma} \tag{1.8}$$

and it is clear that (1.7) is satisfied. Moreover, if $B^{(k)} > 0$, then $B^{(k+1)}$ is positive definite if and only if

$$\delta^{(k)T}\gamma^{(k)} > 0. \tag{1.9}$$

The scalar $\delta^{(k)T}\gamma^{(k)}$ represents from (1.5) the component of the average Hessian matrix along $\delta^{(k)}$. It is easily possible to ensure that this condition holds.

A significant result due to Goldfarb [6] is that if $H^{(k)}$ denotes $B^{(k)-1}$, then the correction $E = H^{(k+1)} - H^{(k)}$ in the BFGS formula satisfies a minimum property with respect to a weighted Frobenius norm of the form

$$\|E\|_W^2 = \|W^{\frac{1}{2}}EW^{\frac{1}{2}}\|_F^2 = (\text{trace}(EWEW))^{\frac{1}{2}}, \tag{1.10}$$

where $W > 0$ and $W\delta^{(k)} = \gamma^{(k)}$. This result can be interpreted as showing that $H^{(k)}$ is changed by the minimum amount (in the sense of (1.10)) required to satisfy (1.7) and symmetry. This ensures that previous information accumulated in $B^{(k)}$ is disturbed as little as possible. The well-known Davidon–Fletcher–Powell (DFP) formula can also be interpreted in a similar way. Another formula that satisfies a minimum correction property is the Powell-symmetric-Broyden (PSB) formula

$$B^{(k+1)} = B^{(k)} + \frac{\eta\delta^T + \delta\eta^T}{\delta^T\delta} - \frac{\eta^T\delta}{(\delta^T\delta)^2}\delta\delta^T, \tag{1.11}$$

where $\eta = \gamma - B^{(k)}\delta$ and where δ and γ denote $\delta^{(k)}$ and $\gamma^{(k)}$, respectively. In this case it is the Frobenius norm ($W = I$ in (1.10)) of the correction to $B^{(k)}$ that is minimized

(subject to (1.7) and symmetry). Unfortunately the PSB update does not generally preserve $B^{(k)} > 0$ and practical experience has been disappointing. This is thought to be due to the lack of certain affine invariance properties that hold for the BFGS and DFP formulae. More detail about all the above subject matter is given, for example, in [4].

Quasi-Newton methods become less attractive when n is very large because of the storage and computational requirements associated with large dense matrices. However, it is often the case that the Hessian is a sparse matrix and it is attractive to look for updating formulae that preserve the same sparsity in $B^{(k)}$. Thus we express the sparsity conditions on B as

$$B_{ij} = 0 \quad \forall (i,j) \in \mathcal{S}, \tag{1.12}$$

where $\mathcal{S}$ is a set of pairs of integers in the range $[1 : n]$. Because of symmetry it is assumed that $(i,j) \in \mathcal{S}$ if and only if $(j,i) \in \mathcal{S}$. It is also assumed that $G(x)$ satisfies (1.12) for all $x \in \mathbb{R}^n$. The complementary set of index pairs not in $\mathcal{S}$ is denoted by $\mathcal{S}^\perp$. Because we are concerned with positive definite matrices, it is assumed that

$$(i,i) \in \mathcal{S}^\perp \quad i = 1,2,\ldots,n. \tag{1.13}$$

For such problems it is fruitful to determine a minimum correction update formula that is constrained by (1.12) in addition to the quasi-Newton condition and symmetry. In a seminal paper, Toint [10] shows that it is reasonably straightforward to compute a minimum correction to $B^{(k)}$ in the Frobenius norm subject to these conditions. To present Toint's update, we define the projection operator $\mathcal{G}(M) : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ by

$$\mathcal{G}(M)_{ij} = \begin{cases} 0 & (i,j) \in \mathcal{S} \\ M_{ij} & (i,j) \in \mathcal{S}^\perp. \end{cases}, \tag{1.14}$$

This has been colourfully dubbed the *gangster operator* since it shoots holes in M according to the sparsity pattern defined by (1.12). Toint also introduces the notation

$$\delta_{[i]} = \mathcal{G}(\delta e_i^T)e_i, \tag{1.15}$$

where e_i denotes the unit vector that is column i of I. (The use of subscript $[i]$ is due to Coleman [2] in his very readable monograph on large sparse optimization.) Toint [10] shows that the resulting minimum correction satisfies

$$B^{(k+1)} = B^{(k)} + \mathcal{G}(\delta\lambda^T + \lambda\delta^T), \tag{1.16}$$

where $\lambda \in \mathbb{R}^n$ is obtained by solving the linear system

$$Q\lambda = r \tag{1.17}$$

in which $r = \gamma - B^{(k)}\delta$. It follows from (1.16) and (1.7) that column i of the matrix Q is defined by

$$Qe_i = \delta_i\delta_{[i]} + \delta_{[i]}^T\delta_{[i]}e_i. \tag{1.18}$$

It is easily shown that Q is symmetric positive semidefinite and that $\mathcal{G}(Q) = Q$ (i.e. Q satisfies the sparsity conditions (1.12)). In addition, Q is positive definite if

$$\delta_{[i]} \neq 0 \quad i = 1,2,\ldots,n, \tag{1.19}$$

in which case sparse LDL^T factors of Q are calculated and (1.17) can readily be solved to obtain λ. If Q is singular then $\delta_{[i]} = 0$ for some i. However, it then follows from (1.5) that $\gamma_i = 0$ and hence $r_i = 0$. Thus (1.17) is consistent and can be solved by deleting row and column i from Q (ignoring the effects of round-off error).

As with the dense PSB update, the condition $B^{(k)} > 0$ is not preserved by this update, and likewise practical performance has not been outstanding. In view of this, it is natural to inquire what happens when the sparsity conditions (1.12) are included in the minimum correction property that defines the BFGS or DFP formula. Unfortunately, as Toint [10] points out, the use of a weighted Frobenius norm leads to formulae that are intractable in both cases (see, also, [11] and [2] for more details). However, Toint [11] proves that if

$$\delta_i \neq 0 \quad i = 1, 2, \ldots, n, \tag{1.20}$$

if G is irreducible, and if $\delta^T\gamma > 0$, then there does exist a symmetric update that preserves positive definiteness. The condition $\delta^T\gamma > 0$ is clearly required, else (1.7) would imply $\delta^T B\delta \leq 0$, contradicting $B > 0$. The assumption of irreducibility (that is, G cannot be reduced to block diagonal form by a symmetric permutation) is not a serious restriction because if G is reducible then (1.1) can be decomposed into two or more problems which can be solved separately. It is assumed throughout what follows that G is irreducible. On the other hand, (1.20) is critical and Sorensen [9] shows that a positive definite update may not exist if $\delta_i = 0$ for some i, and that serious growth in B can occur in a neighbourhood of this situation. We return to these points later in the paper.

More recent research into sparse updates has avoided the requirement that $B^{(k)}$ should be positive definite. Most promising has been the approach of Griewank and Toint (e.g., [7]) based on the partially separable optimization problem

$$\text{minimize} \quad f(x) = \sum_{i=1}^{m} f_i(x), \tag{1.21}$$

in which each *element function* $f_i(x)$ depends on only a few of the components of x. Then the Hessian of $f(x)$ can be decomposed into a sum of Hessians of the $f_i(x)$, the nontrivial submatrices of which can be treated as dense matrices. Similar remarks apply for the gradient vector. Griewank and Toint suggest that these element Hessian submatrices are approximated by the use of dense updating techniques. There are, however, some difficulties that must be overcome. The element Hessian submatrices may not be positive definite so it is not possible to rely on the analogue of the condition $\delta^T\gamma > 0$ holding for each submatrix. Thus the BFGS formula cannot be used, and Toint uses the symmetric rank-one formula in the Harwell Subroutine Library code VE08. Consequently, the overall Hessian approximation obtained by summing the submatrix approximations is also not generally positive definite. Also the possibility of zero in the denominator of the updates must be allowed for. Another aspect is that the user must specify the decomposition (1.21) to the code and this is not always convenient. These remarks are not intended to disparage the method, which has been very successful in practice, but they do indicate that if an effective sparse positive definite update were available, then many of these difficulties would be circumvented.

This paper makes a contribution to this objective by providing an update that preserves sparsity and positive definiteness and satisfies a minimal change property. The update reduces to the BFGS update in the dense case. However, the amount of

computation required to compute the update is not trivial (although improvements here may well be possible) and the difficulties noted by Sorensen are still present. The new update arises from a recent observation of Fletcher [5] that the BFGS and DFP formulae can be derived by a variational argument using the measure function

$$\psi(A) = \text{trace}(A) - \ln \det(A), \tag{1.22}$$

where ln denotes the natural logarithm. This function is introduced by Byrd and Nocedal [1] in the convergence analysis of quasi-Newton methods. The function is strictly convex on the set of positive definite matrices and is minimized by $A = I$. The function becomes unbounded as A becomes singular or infinite and so acts as a barrier function that keeps A positive definite. A suitable variational property is to minimize $\psi(H^{(k)}B)$ since, in the absence of any constraints, the solution is just $B = H^{(k)-1}$. Introducing the constraints (1.7) and (1.12) leads to an update in which $H = B^{-1}$ stays close to $H^{(k)}$ in some sense. The objective function can also be expressed as

$$\psi(H^{(k)}B) = \psi(BH^{(k)}) = \psi({H^{(k)}}^{1/2}B{H^{(k)}}^{1/2})$$

using the properties of the trace and determinant. In the sparse case it is also noted that $\psi(H^{(k)}B)$ can be computed from only B and $\mathcal{G}(H^{(k)})$ and the full matrix $H^{(k)}$ is not required.

A theorem extending the BFGS result in [5] to include the sparsity conditions (1.12) is set out in §2. Necessary conditions related to a rank-two correction of the form $\delta\lambda^T + \lambda\delta^T$ are derived. However, there is also a surprising outcome in that the matrix $\mathcal{G}(H)$ is seen to play a major role. The issue of whether $\mathcal{G}(H)$ determines B, and how this calculation can be carried out, is seen to be fundamental to the update. It is shown that the update can be determined by solving a nonlinear system $r(\lambda) = 0$ involving the residual of the quasi-Newton condition. Unfortunately, the DFP formula cannot be generalised in the same way.

Issues concerning the determination of B from $\mathcal{G}(H)$ are considered in §3. A simplifying assumption is made that the sparsity pattern specified by $\mathcal{S}$ is such that elements that fill in during the calculation of LDL^T factors of B are in $\mathcal{S}^{\perp}$. In the notation of Duff, Erisman, and Reid [3] this can be expressed as

$$\mathcal{G}(L\backslash L^T) = L\backslash L^T. \tag{1.23}$$

Another way of expressing this is that B does not fill in with respect to $\mathcal{S}$ when factored. This assumption is not very restrictive in practice since factors of B are required in (1.3) to determine the search direction, which necessitates using a data structure for B that allows for fill-in. With this assumption it is shown that B is readily determined from $\mathcal{G}(H)$. The inverses of certain submatrices of H, referred to as Markowitz submatrices, are shown to play an important role.

In §4 the solution of the system $r(\lambda) = 0$ is considered. It is shown that $r(\lambda)$ is the gradient of a concave programming problem derived by using the Wolfe dual and this enables the solution of $r(\lambda) = 0$ to be undertaken in a reliable way. The Jacobian $Q(\lambda)$ of this system is important and plays a similar role to the matrix Q that arises in the linear system (1.17) in Toint's sparse PSB update. The structure of Q is analysed in detail in §5, and is shown to satisfy the same structural and definiteness conditions (when (1.23) holds) as for Toint's matrix. Thus the nonlinear system can be solved by a few iterations of analogous complexity to (1.17).

The update has been implemented for tridiagonal systems and some numerical experiments are described in §6. These experiments indicate that there is the potential for a significant reduction in the number of quasi-Newton iterations, but that more development is needed to obtain an efficient implementation. Section 7 discusses the possibility of using primal algorithms to determine the update and provides an interesting application of generalized elimination. This leads to Toint's result on the existence of a positive definite update subject to (1.20). The issue of stability raised by Sorensen is discussed in §8, together with other points of interest including a conjecture related to partially separable updates. Directions for further research are suggested.

2. A variational result. The main aim of this section is to extend Theorem 2.1 of [5] to include the sparsity conditions (1.12).

THEOREM 2.1. *Let $B^{(k)}$ be positive definite and consider the solution of the variational problem*

$$\underset{B>0}{\text{minimize}} \quad \psi(H^{(k)}B) \tag{2.1}$$

$$\text{subject to} \quad B^T = B, \tag{2.2}$$

$$B\delta = \gamma, \tag{2.3}$$

$$B_{ij} = 0 \quad \forall\,(i,j) \in \mathcal{S}. \tag{2.4}$$

If a solution exists it is characterised by the existence of $\lambda \in \mathbb{R}^n$ such that

$$\mathcal{G}(H) = \mathcal{G}(H^{(k)} + \lambda\delta^T + \delta\lambda^T), \tag{2.5}$$

where H denotes B^{-1}.

Proof. If a solution to the variational problem exists, it satisfies $B > 0$. Because the remaining constraints in the problem are linear, constraint qualification holds and first order conditions obtained by the method of Lagrange multipliers are necessary for a solution. A suitable Lagrangian function is

$$\begin{aligned}\mathcal{L}(B,\Lambda,\lambda,\Pi) &= \tfrac{1}{2}\psi(H^{(k)}B) + \operatorname{trace}(\Lambda^T(B^T - B)) + \lambda^T(B\delta - \gamma) + \tfrac{1}{2}\operatorname{trace}(\Pi^T B)\\ &= \tfrac{1}{2}(\operatorname{trace}(H^{(k)}B) - \ln\det H^{(k)} - \ln\det B) + \operatorname{trace}(\Lambda^T(B^T - B))\\ &\qquad + \lambda^T(B\delta - \gamma) + \tfrac{1}{2}\operatorname{trace}(\Pi^T B),\end{aligned}$$

where Λ, λ, and Π are Lagrange multipliers for (2.2), (2.3), and (2.4) respectively. Without loss of generality it can be assumed that Λ is strictly lower triangular and Π is symmetric. Because $B_{ij} = 0$ does not apply for $(i,j) \in \mathcal{S}^{\perp}$, it follows that

$$\mathcal{G}(\Pi) = 0. \tag{2.6}$$

To solve the first order conditions it is necessary to find B, Λ, λ, and Π to satisfy (2.2), (2.3), (2.4), and the equations $\partial\mathcal{L}/\partial B_{ij} = 0$. Using the identity $\partial B/\partial B_{ij} = e_i e_j^T$ and Lemma 1.4 of [5], it follows that

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial B_{ij}} = 0 &= \tfrac{1}{2}(\operatorname{trace}(H^{(k)}e_i e_j^T) - (B^{-1})_{ji}) + \operatorname{trace}(\Lambda^T(e_j e_i^T - e_i e_j^T))\\ &\qquad + \lambda^T e_i e_j^T\delta + \tfrac{1}{2}\operatorname{trace}(\Pi^T e_i e_j^T)\\ &= \tfrac{1}{2}((H^{(k)})_{ji} - (B^{-1})_{ji}) + \Lambda_{ji} - \Lambda_{ij} + (\lambda\delta^T)_{ij} + \tfrac{1}{2}\Pi_{ij}.\end{aligned}$$

Transposing and adding, using the symmetry of $H^{(k)}$, B and Π, gives

$$H^{(k)} - B^{-1} + \lambda\delta^T + \delta\lambda^T + \Pi = 0$$

or

$$H = H^{(k)} + \lambda\delta^T + \delta\lambda^T + \Pi. \tag{2.7}$$

Then (2.5) is deduced directly from (2.6) and (2.7). □

Although the derivation of this result is straightforward, the outcome came as a major surprise to me for the following reason. If B is an irreducible sparse matrix then B^{-1} is generally dense. It is therefore most unexpected to find that B is determined by $\mathcal{G}(H)$ (i.e., by zeroing elements of B^{-1} in accordance with the sparsity pattern of B).

The result for the dense case given in [5] corresponds to $\Pi = 0$ and shows that the optimum H matrix involves a rank-two correction of $H^{(k)}$. In that case it is possible to directly solve for H using $B\delta = \gamma$ and the resulting update is the BFGS formula. Because the resulting B matrix is positive definite and ψ is a convex function it is possible to deduce that it solves the variational problem.

When the sparsity conditions (2.4) are included, solution of (2.7) is no longer straightforward because of the additional term Π, and a finite calculation to determine λ does not appear to be possible. However, an iterative approach along the following lines can be envisaged. The following sequence of operations defines r as a function of λ ($\mathbb{R}^n \to \mathbb{R}^n$).

Given λ,
calculate $\mathcal{G}(H)$ from (2.5),
find $B > 0$ such that $\mathcal{G}(B^{-1}) = \mathcal{G}(H)$,
calculate $r := B\delta - \gamma$.

The update is determined by finding λ such that

$$r(\lambda) = 0, \tag{2.8}$$

which is a nonlinear system of n equations in n variables.

This discussion raises a number of interesting and complex issues that are addressed in the rest of the paper. First is the question as to whether $\mathcal{G}(H)$ does indeed determine $B > 0$ and whether the outcome is unique. It is hopeful that $\mathcal{G}(H)$ contains the same number of nonzero elements as B. A solution is given to this question in §3 in the case that assumption (1.23) holds. An illustration of the calculation is given in §6 for the case of tridiagonal matrices. However, when assumption (1.23) does not hold, then the question is as yet unresolved, although it is shown that the condition $(i, i) \in \mathcal{S}^{\perp}$ is necessary.

Given that $B > 0$ is well determined, the next question is that of whether a solution to the variational problem, and hence to (2.8), does exist. The contributions of Toint and Sorensen provide a complete answer to this question as discussed in §1. Toint's result can be seen as a consequence of the presentation on primal algorithms set out in §7 and the issues are discussed in more detail in §8.

Another question relates to the practicality of solving (2.8) reliably and rapidly. At first sight this is not promising since a nonlinear system might be as hard to solve as the original problem (1.1). However, it is shown in §§4 and 5 that there are features

that enable (2.8) to be solved effectively when assumption (1.23) holds and a solution exists. Preliminary numerical experiments described in §6 indicate that the extra expense of solving (2.8) is compensated for by a significant reduction in the number of line searches required to solve (1.1).

Before following up these questions, it is worth remarking that another result in [5] regarding the DFP update does not carry over conveniently to the sparse case. Extending the corollary to Theorem 2.1 in [5] gives the variational problem

$$\text{minimize}_{H>0} \quad \psi(B^{(k)}H), \tag{2.9}$$

$$\text{subject to} \quad H^T = H \tag{2.10}$$

$$H\gamma = \delta, \tag{2.11}$$

$$B_{ij} = 0 \quad \forall (i,j) \in \mathcal{S}. \tag{2.12}$$

After proceeding analogously to Theorem 2.1 here, and setting $\partial \mathcal{L}/\partial H_{ij} = 0$, it follows that

$$B = B^{(k)} + \lambda\gamma^T + \gamma\lambda^T + B\Pi B. \tag{2.13}$$

It is possible to pre and post multiply by H and operate with $\mathcal{G}(.)$ to eliminate Π, but the resulting term $\mathcal{G}(HB^{(k)}H)$ appears to make further progress unlikely.

3. Determination of B from $\mathcal{G}(H)$. The results of this section are a consequence of assumption (1.23) discussed in §1 that $\mathcal{S}$ is chosen such that there is no fill-in when the LDL^T factors of B are calculated. The main result shows that B is well determined by $\mathcal{G}(H)$ and provides the basis of an algorithm for computing the LDL^T factors of B. This algorithm is shown to be particularly efficient when the sparsity pattern of B is formed from dense overlapping blocks on the diagonal.

A lemma is required that shows the effect on the inverse when bordering a partitioned matrix with a rank-one term having some sparsity.

LEMMA 3.1. *Consider symmetric matrices partitioned conformally as*

$$A = \begin{bmatrix} 0 & 0^T & 0^T \\ 0 & A_{11} & A_{12} \\ 0 & A_{21} & A_{22} \end{bmatrix} + \begin{pmatrix} 1 \\ a/\alpha \\ 0 \end{pmatrix} (\alpha \quad a^T \quad 0^T), \quad X = \begin{bmatrix} \xi & x_1^T & x_2^T \\ x_1 & X_{11} & X_{12} \\ x_2 & X_{21} & X_{22} \end{bmatrix} \tag{3.1}$$

($A_{21} = A_{12}^T$, *etc.*) *in which*

$$\begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}^{-1} > 0 \tag{3.2}$$

is positive definite and

$$\begin{bmatrix} \xi & x_1^T \\ x_1 & X_{11} \end{bmatrix} > 0 \tag{3.3}$$

is positive definite. Then a necessary and sufficient condition for $A = X^{-1}$ is that both

$$\begin{bmatrix} \xi & x_1^T \\ x_1 & X_{11} \end{bmatrix} \begin{pmatrix} \alpha \\ a \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{3.4}$$

and

$$x_2 = -A_{22}^{-1}A_{21}x_1 \tag{3.5}$$

hold. Moreover α, a, *and* x_2 *are determined uniquely by* ξ, x_1, *and* A_{11}, A_{12}, *and* A_{22}.

Proof. First of all it is readily established from the properties of the inverse in (3.2) that

$$A_{12}A_{22}^{-1} = -X_{11}^{-1}X_{12} \tag{3.6}$$

and

$$X_{11}^{-1} = A_{11} - A_{12}A_{22}^{-1}A_{21}. \tag{3.7}$$

Also the solution of (3.4) can be expressed as

$$\begin{pmatrix} \alpha \\ a \end{pmatrix} = \begin{pmatrix} 1 \\ -X_{11}^{-1}x_1 \end{pmatrix} / (\xi - x_1^T X_{11}^{-1} x_1). \tag{3.8}$$

To prove the main result, we form the product

$$XA = \begin{bmatrix} \xi & x_1^T \\ x_1 & X_{11} \\ x_2 & X_{21} \end{bmatrix} \begin{pmatrix} \alpha \\ a \end{pmatrix} (1 \quad a^T/\alpha \quad 0^T) + \begin{bmatrix} 0 & x_1^T A_{11} + x_2^T A_{21} & x_1^T A_{12} + x_2^T A_{22} \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}.$$

Now $A = X^{-1}$ if and only if $XA = I$, which is seen to require that both (3.4) and (3.5) hold. Conversely, if (3.4) and (3.5) hold, then it follows that

$$x_2\alpha + X_{21}a = (x_2 - X_{21}X_{11}^{-1}x_1)\alpha = 0$$

from (3.5), (3.6), and (3.8) and

$$A_{11}x_1 + A_{12}x_2 + a/\alpha = (A_{11} - A_{12}A_{22}^{-1}A_{21} - X_{11}^{-1})x_1 = 0$$

from (3.5), (3.7), and (3.8), showing that $XA = I$. Thus the main result is established. It is clear from (3.4) and (3.5) and the existence of the relevant inverses that α, a, and x_2 are determined uniquely by ξ, x_1, and A_{11}, A_{12}, and A_{22}. □

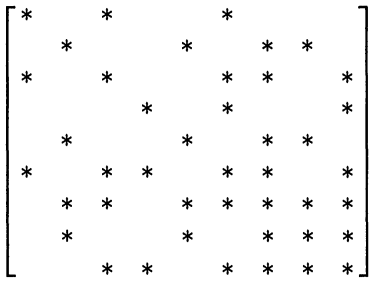
Some other items of terminology are introduced to simplify the description of the main result. When factors $B = LDL^T$ (L unit lower triangular, D diagonal, $B \in \mathbb{R}^{n\times n}$) are calculated by Gaussian elimination, then it is convenient to refer to

$$B^{(i)} = B - \sum_{j=1}^{i-1} l_j d_j l_j^T = \begin{bmatrix} 0 & 0 \\ 0 & B_{22}^{(i)} \end{bmatrix} \tag{3.9}$$

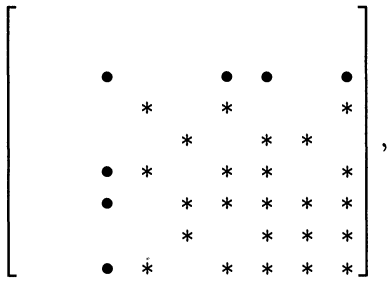
as the ith reduced matrix in the calculation. Here l_i denotes the ith column of L and d_i the ith diagonal element of D.

DEFINITION. *Let sparse factors* $B = LDL^T$ *exist. The* i*th* Markowitz submatrix *of any matrix* M *of the same dimension as* B *is defined to be the submatrix obtained by selecting elements of* M *corresponding to the structural nonzero elements of* $l_i l_i^T$.

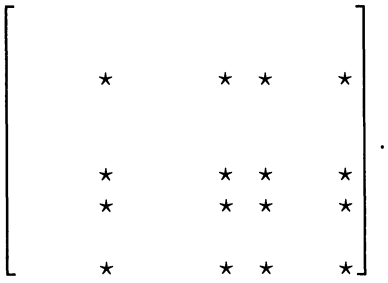
The definition is illustrated by the following matrices:



Matrix B satisfying (1.23)



Reduced matrix $B^{(3)}$



Expanded Markowitz submatrix $M_{[3]}$

The • elements in $B^{(3)}$ are those that determine the vector l_3 and the diagonal element d_3. If M is any $n \times n$ matrix, then $M_{[3]}$ is the matrix obtained by selecting the $\star$ elements of M and zeroing the others. The pattern of these elements is related to the possible fill-in caused by the • elements when processing the reduced matrix $B^{(3)}$ (hence the term Markowitz submatrix). It is convenient to extend Toint's [10] notation by using $M_{[i]}$ to refer to the expanded $(n \times n)$ ith Markowitz submatrix of M as in the above figure, and M_i to refer to the (dense) submatrix itself. The requirement that no fill-in is caused when factorizing B is equivalent to requiring that the pattern of nonzeros of B is that obtained from the overlay of all the matrices $B_{[i]}$.

The main theorem of this section can now be proved. The matrices B and H are assumed to be symmetric and the sparsity pattern of the Markowitz submatrices is that induced by the structure of the LDL^T factorization of B. The theorem shows that B is uniquely determined by $\mathcal{G}(H)$ and provides a construction in which the factors of B are expressed directly in terms of the Markowitz submatrices $H_{[i]}$. To do this the generalized inverse notation $H_{[i]}^{+}$ is used. Because H_i is assumed to be positive definite, $H_{[i]}^{+}$ consists simply of the elements of the matrix H_i^{-1} scattered into the positions corresponding to nonzero elements of $H_{[i]}$.

THEOREM 3.1. *Let assumption* (1.23) *hold. If for* $i = 1, 2, \ldots, n$ *the Markowitz submatrices* H_i *are positive definite, then the* i*th diagonal element of* D *and the* i*th column of* L *are defined by*

$$d_i = e_i^T H_{[i]}^{+} e_i, \qquad l_i = H_{[i]}^{+} e_i / d_i. \tag{3.10}$$

Moreover, B *is uniquely determined by* $\mathcal{G}(H)$ *and* B *is positive definite.*

Proof. Consider the reduced matrix $B^{(i)}$ in the calculation of $B = LDL^T$ and partition

$$B^{(i)} = \begin{bmatrix} 0 & 0 \\ 0 & B_{22}^{(i)} \end{bmatrix}, \qquad H = \begin{bmatrix} H_{11}^{(i)} & H_{12}^{(i)} \\ H_{21}^{(i)} & H_{22}^{(i)} \end{bmatrix}$$

as indicated by (3.9). The notation $H_{11}^{(i)}$, etc. just indicates a different partitioning of the same H matrix as i changes. The theorem only assumes that $\mathcal{G}(H)$ is known, and shows that the unknown elements of H can be deduced. Also from (3.9) it follows that

$$B^{(i)} = B^{(i+1)} + l_i d_i l_i^T, \tag{3.11}$$

where, because of assumption (1.23), column l_i has the same sparsity structure as column i of B and hence as column i of $H_{[i]}$.

The main step is to prove by induction that

$$B_{22}^{(i)} = H_{22}^{(i)-1} > 0 \text{ and is determined uniquely by } H_i, \ldots, H_n. \tag{3.12}$$

When $i = n$, $H_n > 0$ is just a scalar and $B_{22}^{(n)} = H_n^{-1}$ determines $B_{22}^{(n)}$ uniquely. Now we assume that (3.12) is true with i replaced by $i+1$ and deduce that (3.12) itself is true. $H_{22}^{(i)}$ is obtained by bordering $H_{22}^{(i+1)}$ with some elements from column i of H. To apply Lemma 3.1 to $H_{22}^{(i)}$, it can be assumed without loss of generality that a symmetric row and column permutation is made to $H_{22}^{(i)}$ so that the elements of the Markowitz submatrix H_i occur in adjacent rows and columns. Thus we can identify H_i with the matrix

$$\begin{bmatrix} \xi & x_1^T \\ x_1 & X_{11} \end{bmatrix} \tag{3.13}$$

in Lemma 3.1. Note that the vector x_2 in (3.1) represents the unknown elements from H that occur in the border. The positive definiteness of (3.13) follows from the same assumption about H_i. Moreover because the structure of l_i is prescribed, and matches that of column i of $H_{[i]}$, we can identify the subdiagonal part of l_i with $\binom{a}{0}$ and also $d_i = \alpha$. In addition we have

$$B_{22}^{(i+1)} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \qquad H_{22}^{(i+1)} = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix}.$$

It follows from (3.12) with i replaced by $i+1$ that these matrices satisfy condition (3.2) of Lemma 3.1. Thus the lemma can be invoked and it follows that a necessary and sufficient condition to obtain $B_{22}^{(i)} = H_{22}^{-1}$ is that (3.4) and (3.5) hold. It follows from (3.4) and the identification of H_i with (3.13) that l_i and d_i are defined by (3.10). Equation (3.5) determines the unknown elements of H represented by x_2, although these are not required to compute the factors of B. Because of (3.10) and H_i being positive definite, it follows that $d_i > 0$ and hence by induction that $B_{22}^{(i)}$ is positive definite. It follows from the lemma that d_i and l_i are uniquely defined by $B_{22}^{(i+1)}$ and the first column of H_i. Hence by induction $B_{22}^{(i)}$ depends on $H_i, \ldots, H_n$ and (3.12) is established.

Finally for $i = 1$, $B = B^{(1)} = H^{-1}$ and the LDL^T factors of B have been determined uniquely from $H_1, \ldots, H_n$. Because of assumption (1.23), the elements of the Markowitz submatrix are available in $\mathcal{G}(H)$ to make the calculation. □

Note that this proof does not require a prior assumption that H is positive definite. Rather this can be deduced as a consequence of the positive definiteness of the Markowitz submatrices H_i and the inductive argument.

It can be observed that the construction calculates the LDL^T factors of B rather than B itself. This is convenient as B is subsequently used to solve systems. It is also possible to express

$$B = LDL^T = \sum_{i=1}^{n} l_i d_i l_i^T = \sum_{i=1}^{n} \frac{H_{[i]}^{+} e_i e_i^T H_{[i]}^{+}}{e_i^T H_{[i]}^{+} e_i}. \tag{3.14}$$

This form is particularly useful for the purposes of §5.

In the case that the pattern of nonzeros in B consists of overlapping dense diagonal blocks then a particularly efficient algorithm is determined. Special cases of such patterns are the symmetric band matrices of arbitrary bandwidth. The columns of D and L are determined in the order $1, 2, \ldots, n$, and for each i, USU^T factors of H_i are available, where U is upper triangular and $S > 0$ is diagonal. This corresponds to having factorized H_i taking pivots in the reverse order. Then $H_i^{-1} = U^{-T} S^{-1} U^{-1}$ and it is readily observed that $d_i = S_{11}^{-1}$ and the nonzero part of l_i is obtained by solving the system $U^T x = e_1$. Moreover, advantage can be taken of the overlap in the Markowitz submatrices of H in the following way. When i is incremented, the first row and column of U and S are deleted. If i changes to move into the next overlapping block, then the remaining part of the USU^T factors is bordered by elements of a new submatrix. When factorising this submatrix, a low rank matrix is added into the old USU^T factors that can be updated efficiently by the use of square root free Givens' rotations.

It may be that a similar approach can be used when the pattern of nonzeros is less regular, but the organization is likely to be more complex. It is hoped to investigate this aspect in the near future. An observation of some interest is that Duff, Erisman, and Reid [3, §12.7] reference a method of calculation whereby $\mathcal{G}(A^{-1})$ can be calculated from the LU factors of A under the same assumptions about fill-in. The construction given in Theorem 3.1 can be regarded as reversing this calculation in the symmetric case.

Finally, the case when assumption (1.23) does *not* hold is of some interest. The above scheme no longer applies because the elements of the Markowitz submatrix H_i are not always available to calculate column i of $B_{22}^{(i)}$. However, B and $\mathcal{G}(H)$ still have the same number of nonzero elements and it is conjectured that B will remain well determined if H is a positive definite matrix. For a general unsymmetric matrix A, A is not always well determined by $\mathcal{G}(A^{-1})$ as the following example shows.

$$A = \begin{bmatrix} a & b \\ c & 0 \end{bmatrix}, \quad A^{-1} = \begin{bmatrix} 0 & 1/c \\ 1/b & -a/bc \end{bmatrix}, \quad \mathcal{G}(A^{-1}) = \begin{bmatrix} 0 & 1/c \\ 1/b & 0 \end{bmatrix}.$$

Here we have $\mathcal{S} = \{(2,2)\}$ (relaxing the condition $(i,i) \in \mathcal{S}^{\perp}$). When the $(2,2)$ element of A^{-1} is deleted, then all reference to a is lost and A is not uniquely defined by $\mathcal{G}(A^{-1})$.

4. Solving the system $r(\lambda) = 0$. This section considers the practicality of finding a reliable and efficient method for solving the nonlinear system $r(\lambda) = 0$ generated by the prototype algorithm that precedes (2.8). The method we consider is based on Newton's method with iterates $\lambda^{(t)}$, $t = 1, 2, \ldots$, and an iteration formula

$$Q^{(t)} \Delta\lambda^{(t)} = -r(\lambda^{(t)}), \tag{4.1}$$

where $Q^{(t)}$ denotes the Jacobian of $r(\lambda)$ evaluated at $\lambda^{(t)}$. At first sight this is not promising since a nonlinear system might be at least as hard to solve as the main problem (1.1), but it turns out that there are some simplifying features that can be exploited.

It is observed in [5] that the problem posed in (2.1), (2.2), and (2.3) is a convex programming problem and significant progress is made by examining the Wolfe dual (see, for example, [4]). The Wolfe dual is

$$\underset{B,\Lambda,\lambda,\Pi}{\text{maximize}} \quad \mathcal{L}(B,\Lambda,\lambda,\Pi) \tag{4.2}$$

$$\text{subject to} \quad \nabla_B \mathcal{L} = 0, \tag{4.3}$$

and it has been shown in Theorem 2.1 that (4.3) implies that B $(= H^{-1})$ is defined by (2.5). Section 3 gives a scheme for calculating a positive definite matrix B from $\mathcal{G}(H)$ and this enables the variable B in the dual to be expressed as a function $B(\lambda)$. The terms involving Λ and Π are eliminated by virtue of the symmetry and sparsity of $B(\lambda)$, giving rise to the more simple dual problem

$$\begin{aligned} &\underset{\lambda}{\text{maximize}} \quad \phi(B(\lambda),\lambda) = \tfrac{1}{2}\psi(H^{(k)}B) + \lambda^T(B\delta - \gamma) \\ &\text{subject to} \quad B(\lambda) > 0. \end{aligned} \tag{4.4}$$

The chain rule gives

$$\frac{d\phi}{d\lambda_k} = \sum_{(i,j)\in\mathcal{S}^\perp} \frac{\partial\phi}{\partial B_{ij}}\frac{\partial B_{ij}}{\partial\lambda_k} + \frac{\partial\phi}{\partial\lambda_k}$$

and $\partial\phi/\partial B_{ij} = 0$ because of the stationary property of the Lagrangian in Theorem 2.1. It follows from (4.4) that

$$\nabla_\lambda \phi = B\delta - \gamma = r \tag{4.5}$$

so that the nonlinear system $r(\lambda) = 0$ that occurs in (2.8) is seen to arise from the stationary point condition for (4.4). It follows that the Jacobian $Q(\lambda)$ of $r(\lambda)$ is the Hessian of $\phi(\lambda)$ and hence is a symmetric matrix. In fact, it is shown in §5 that Q is a negative semidefinite matrix (and usually negative definite) when $B > 0$, so the simplified dual is a concave programming problem. Thus ϕ provides an objective function with which to measure progress when solving (2.8), and enables a line search in λ to be used. It is only necessary to ensure that λ is chosen so that all the Markowitz submatrices of $H^{(k)} + \lambda\delta^T + \delta\lambda^T$ are positive definite. Initially $\lambda^{(1)} = 0$ is suitable, and subsequently the condition imposes an upper limit on the step in the line search.

A result of particular importance is that the matrix Q has the same sparsity structure as B if assumption (1.23) holds. Together with the negative semidefinite property, this enables the Newton step (4.1) to be calculated efficiently, using the same data structure as for B, without the need to allow for fill-in or pivoting. The situation is analogous to the solution of $Q\lambda = r$ in Toint's sparse PSB update. This gives some reason to hope that the cost of solving (2.8) may not be prohibitive. It might be expected that as the outer iterates $x^{(k)}$ approach the solution, fewer iterations of the inner iteration would hopefully be required. Some preliminary numerical experience in this respect is outlined in §6. Detailed expressions for calculating Q and results about the structure of Q are given in §5.

5. Properties of the Jacobian $Q(\lambda)$. This section sets out the properties of the Jacobian matrix $Q(\lambda)$ of the nonlinear system $r(\lambda) = 0$ generated by the

algorithm preceding (2.8). General results with respect to symmetry and negative semidefiniteness are established without the need for assumption (1.23). A necessary and sufficient condition for Q to be negative definite is given, analogous to a result of Toint [10]. It is also shown that if assumption (1.23) holds, then the sparsity structure of Q is identical to that of B, and an expression is given from which Q can be calculated. Finally, an example is provided that shows that if (1.23) does not hold, then Q may be less sparse than B. The significance of these results has already been discussed in §4.

In formulating some general results about Q without the need for assumption (1.23), it is necessary to assume that B is positive definite and is well determined by $\mathcal{G}(H)$. Precisely what minimal set of conditions are required to assure this situation is as yet an open question. To establish the first results in this section, it is assumed that B has been determined from $\mathcal{G}(H^{(k)} + \delta\lambda^T + \lambda\delta^T)$ in such a way as to be a continuously differentiable function of λ, and the effect of a perturbation

$$\lambda \to \lambda + \varepsilon z \tag{5.1}$$

($\varepsilon \in \mathbb{R}$, $z \in \mathbb{R}^n$) is considered. This induces a perturbation ΔB in B that maintains the sparse structure of B. A perturbation ΔH in H is also induced that generally affects all the elements of H. It follows that the derivative with respect to ε of the perturbation in H is

$$\dot{H} = \lim_{\varepsilon \to 0} \Delta H/\varepsilon = \mathcal{G}(z\delta^T + \delta z^T) + \Omega, \tag{5.2}$$

where Ω allows for the variation of the elements of H that are zeroed by $\mathcal{G}(H)$. Consequently,

$$\mathcal{G}(\Omega) = 0, \qquad \Omega^T = \Omega, \tag{5.3}$$

and Ω depends upon z. An expression for the derivative of the perturbation in B can be established by differentiating through the equation $BH = I$, giving

$$\dot{B} = -B\dot{H}B. \tag{5.4}$$

In the case that z is a unit vector e_k, the derivative

$$\begin{aligned} \frac{\partial H}{\partial \lambda_k} &= \mathcal{G}(e_k\delta^T + \delta e_k^T) + \Omega_k \\ &= e_k\delta_{[k]}^T + \delta_{[k]}e_k^T + \Omega_k \end{aligned} \tag{5.5}$$

is obtained, making use of Toint's notation described in §1. Similarly, Ω_k depends upon k and satisfies (5.3), although its value is unlikely to be readily available. An expression for $\partial B/\partial\lambda_k$ that follows from (5.4) is

$$\frac{\partial B}{\partial \lambda_k} = -B(e_k\delta_{[k]}^T + \delta_{[k]}e_k^T + \Omega_k)B. \tag{5.6}$$

A consequence of the sparsity of ΔB is that $\partial B/\partial\lambda_k$ has the same sparsity pattern as B. In effect, the Ω_k matrix in (5.6) is determined by the need to achieve this outcome (there are just enough free parameters).

Turning now to r, it is convenient to use the sparsity and symmetry of B to write

$$r_j = e_j^T B\delta - \gamma_j = e_j^T B\delta_{[j]} - \gamma_j = \tfrac{1}{2}\,\mathrm{trace}((e_j\delta_{[j]}^T + \delta_{[j]}e_j^T)B) - \gamma_j, \tag{5.7}$$

and hence

$$\frac{\partial r_j}{\partial \lambda_k} = \tfrac{1}{2}\,\mathrm{trace}\left((e_j\delta_{[j]}^T + \delta_{[j]}e_j^T)\frac{\partial B}{\partial \lambda_k}\right). \tag{5.8}$$

Moreover, because of the sparsity of $\partial B/\partial\lambda_k$ and (5.3) it follows that

$$\mathrm{trace}\left(\Omega_j \frac{\partial B}{\partial \lambda_k}\right) = 0. \tag{5.9}$$

Equations (5.6)–(5.9) can be incorporated to give

$$\frac{\partial r_j}{\partial \lambda_k} = -\tfrac{1}{2}\,\mathrm{trace}\{(e_j\delta_{[j]}^T + \delta_{[j]}e_j^T + \Omega_j)B(e_k\delta_{[k]}^T + \delta_{[k]}e_k^T + \Omega_k)B\}, \tag{5.10}$$

which shows that the matrix $Q = [\partial r_j/\partial\lambda_k]$ is symmetric.

Next the issue of definiteness of Q is considered and we need to examine z^TQz for some $z \neq 0$. It follows by the chain rule that

$$\sum_{k=1}^{n} Q_{jk}z_k = \sum_{k=1}^{n} \frac{\partial r_j}{\partial \lambda_k}\frac{d\lambda_k}{d\varepsilon} = \frac{dr_j}{d\varepsilon} = \dot{r}_j$$

and hence $z^TQz = z^T\dot{r}$. If λ is perturbed as in (5.1) then it follows from (5.2) and (5.4) that

$$\dot{B} = -B(\mathcal{G}(z\delta^T + \delta z^T) + \Omega)B \tag{5.11}$$

and, as above, $\dot{B}$ has the same sparsity pattern as B. Because of this we can write

$$z^T\dot{r} = z^T\dot{B}\delta = \tfrac{1}{2}\,\mathrm{trace}\{(\mathcal{G}(z\delta^T + \delta z^T))\dot{B}\} \tag{5.12}$$

and hence

$$z^TQz = -\tfrac{1}{2}\,\mathrm{trace}\{(\mathcal{G}(z\delta^T + \delta z^T) + \Omega)B(\mathcal{G}(z\delta^T + \delta z^T) + \Omega)B\}, \tag{5.13}$$

using the fact that $\mathrm{trace}(\Omega\dot{B}) = 0$. Using a weighted Frobenius norm

$$\|A\|_W = (\mathrm{trace}(AWAW))^{1/2}, \qquad W > 0,$$

(5.13) can be expressed as

$$z^TQz = -\tfrac{1}{2}\|\mathcal{G}(z\delta^T + \delta z^T) + \Omega\|_B^2 \leq 0, \tag{5.14}$$

which proves that the matrix Q is negative semidefinite.

The next general result is to show that Toint's condition

$$\delta_{[j]} \neq 0, \qquad j = 1, 2, \ldots, n \tag{5.15}$$

is necessary and sufficient for Q to be negative definite. If $\delta_{[j]} = 0$ for some j, then it follows from (5.7) that $r_j = -\gamma_j$ is independent of B and hence of λ. Thus row (and column) j of Q is zero and Q is singular. For the converse result, let Q be singular so that there exists some $z \neq 0$ such that $z^T Q z = 0$. It is then a consequence of (5.14) that

$$\mathcal{G}(z\delta^T + \delta z^T) + \Omega = 0. \tag{5.16}$$

It follows directly from the diagonal elements that $z_j \delta_j = 0, \quad j = 1, 2, \ldots, n$. Let $z_j \neq 0$ for some j and hence $\delta_j = 0$. Row j of (5.16) implies that

$$z_j \delta_k + \delta_j z_k = 0 \qquad \forall (j,k) \in \mathcal{S}^{\perp}$$

and it follows that $\delta_{[j]} = 0$, which contradicts (5.15).

The singularity of Q when $\delta_{[j]} = 0$ is unlikely to cause difficulties in practice. It is assumed that the vector γ is faithful to the sparsity pattern of the true Hessian G, that is it can be expressed as

$$\gamma = \bar{G}\delta, \tag{5.17}$$

where $\bar{G}$ is the averaged Hessian matrix (1.6). It then follows that $\delta_{[j]} = 0$ implies $\gamma_j = 0$ and hence $r_j = 0$, so that the Newton system (4.1) is consistent. Thus in exact arithmetic a solution with $\Delta\lambda_j = 0$ can be computed and there is no difficulty. Inexact arithmetic poses possible difficulties due to round-off error, but it is hoped that these can be handled adequately by the use of tolerances.

The results so far derived in this section are not very useful for computation because they involve the unknown matrices Ω_k. In the case that assumption (1.23) holds, it is possible to use (3.14) to derive an expression from which Q can be calculated. This expression also enables the sparsity structure of Q to be determined. It follows from (3.14) using $H^+_{[i]}\delta = H^+_{[i]}\delta_{[i]}$ that

$$r_j = \left(\sum_{i=1}^{n} \frac{e_j^T H^+_{[i]} e_i e_i^T H^+_{[i]} \delta_{[i]}}{e_i^T H^+_{[i]} e_i} \right) - \gamma_j. \tag{5.18}$$

In fact, the sum in (5.18) need only be taken over those i for which j is in the scope of $[i]$ (that is j is one of the nonzero rows in the expanded Markowitz submatrix with index $[i]$), because otherwise $e_j^T H^+_{[i]} = 0$. Now let k be in the scope of $[i]$. It follows from (2.5) and the definition of $H_{[i]}$ that

$$\frac{\partial H_{[i]}}{\partial \lambda_k} = e_k \delta^T_{[i]} + \delta_{[i]} e_k^T \tag{5.19}$$

and hence

$$\frac{\partial H^+_{[i]}}{\partial \lambda_k} = -H^+_{[i]}(e_k \delta^T_{[i]} + \delta_{[i]} e_k^T) H^+_{[i]} \tag{5.20}$$

by the same argument as for (5.4). If k is not in the scope of $[i]$ then

$$\frac{\partial H_{[i]}}{\partial \lambda_k} = \frac{\partial H^+_{[i]}}{\partial \lambda_k} = 0. \tag{5.21}$$

It now follows from (5.18) and (5.21) that

(5.22)
$$\frac{\partial r_j}{\partial \lambda_k} = \sum_{i=1}^{n} \left\{ \frac{e_j^T \frac{\partial H_{[i]}^+}{\partial \lambda_k} e_i e_i^T H_{[i]}^+ \delta_{[i]} + e_j^T H_{[i]}^+ e_i e_i^T \frac{\partial H_{[i]}^+}{\partial \lambda_k} \delta_{[i]}}{e_i^T H_{[i]}^+ e_i} - \frac{e_j^T H_{[i]}^+ e_i e_i^T H_{[i]}^+ \delta_{[i]} e_i^T \frac{\partial H_{[i]}^+}{\partial \lambda_k} e_i}{(e_i^T H_{[i]}^+ e_i)^2} \right\},$$

where the sum is taken over those i for which both j and k are in the scope of $[i]$. After substituting (5.20) and denoting $H_{[i]}^+ = M$, $H_{[i]}^+ \delta_{[i]} = v$ and $\delta_{[i]}^T H_{[i]}^+ \delta_{[i]} = \mu$, the ith term in (5.22) can be rearranged as

$$\left(\frac{2M_{ij}M_{ik}}{M_{ii}^2} - \frac{M_{jk}}{M_{ii}} \right) v_i^2 - \frac{(M_{ij}v_k + M_{ik}v_j)}{M_{ii}} v_i - \frac{\mu M_{ij}M_{ik}}{M_{ii}}. \tag{5.23}$$

The symmetry with respect to $j \leftrightarrow k$ can readily be observed. Because the ith term only contributes to the sum if both j and k are in the scope of $[i]$, it follows that the ith term has the same sparsity pattern as $B_{[i]}$, and hence Q has the same sparsity pattern as B. Equations (5.22) and (5.23) provide a (rather complicated) formula from which Q can be evaluated. The expression can be simplified a little by using the fact that $d_i = M_{ii}$ and $l_{ji} = M_{ij}/d_i$ derived from the LDL^T factors of B.

It is also possible to give an example which shows that Q may be less sparse than B if assumption (1.23) does not hold. Consider the matrix

$$B = \begin{bmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{bmatrix}, \tag{5.24}$$

which is sparse on the reverse diagonal and fills in in the (2,3) position when factorized. Consider the computation of $Q_{41} = \partial r_4/\partial \lambda_1$. We first need (5.5), which can be written

$$\frac{\partial H}{\partial \lambda_1} = \begin{bmatrix} 2\delta_1 & \delta_2 & \delta_3 & \omega_1 \\ \delta_2 & & \omega_2 & \\ \delta_3 & \omega_2 & & \\ \omega_1 & & & \end{bmatrix}. \tag{5.25}$$

Then $\partial B/\partial \lambda_1$ can be calculated using (5.6) and (5.24), and ω_1 and ω_2 are chosen to make the reverse diagonal zero. This gives rise to the system

$$\begin{bmatrix} 8 & 1 \\ 1 & 8 \end{bmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \end{pmatrix} = \begin{pmatrix} 2\delta_2 + 2\delta_3 \\ -\delta_1 + 2\delta_2 + 2\delta_3 \end{pmatrix}. \tag{5.26}$$

Then (5.8) gives

$$Q_{41} = \partial r_4/\partial \lambda_1 = (\delta_2 + \delta_3)(\delta_2 + \delta_3 - 4\omega_1 - 4\omega_2) + 2\delta_4\omega_2. \tag{5.27}$$

Clearly from (5.26), ω_2 is not zero, and the presence of the $2\delta_4\omega_2$ term in (5.27) ensures that there are cases in which Q_{41} is not zero. In general, Q is a dense matrix. However, if the sparsity of the (2,3) element is relaxed and B is treated as a band matrix, then the ω_2 parameter is removed and ω_1 is chosen to zero the (4,1) element of (5.6). This

provides the equation $\delta_2 + \delta_3 = 4\omega_1$ and it follows from (5.27) that $Q_{41} = 0$, so that Q is also a band matrix.

6. Numerical experiments. In this section some numerical experiments are described that are designed to test the effectiveness of both the sparse updates in a line search quasi-Newton method and the inner Newton iteration based on (4.1). The experiments are limited to the relatively simple case of a tridiagonal Hessian matrix. The computations have been carried out on a SUN SPARCstation SLC in single precision.

In the case of a tridiagonal Hessian matrix, we may write

$$\mathcal{G}(H) = \begin{bmatrix} a_1 & b_1 & & & \\ b_1 & a_2 & b_2 & & \\ & b_2 & a_3 & \ddots & \\ & & \ddots & \ddots & b_{n-1} \\ & & & b_{n-1} & a_n \end{bmatrix}. \tag{6.1}$$

Clearly assumption (1.23) holds when factorizing B, and the Markowitz submatices are the 2×2 blocks on the diagonal, and the final 1×1 block. It is readily verified from Theorem 3.1 that the LDL^T factors of B are given by

$$L = \begin{bmatrix} 1 & & & & \\ -b_1/a_2 & 1 & & & \\ & -b_2/a_3 & 1 & & \\ & & \ddots & \ddots & \\ & & & -b_{n-1}/a_n & 1 \end{bmatrix}, \tag{6.2}$$

$$D = \begin{bmatrix} a_2/\Delta_1 & & & & \\ & a_3/\Delta_2 & & & \\ & & \ddots & & \\ & & & a_n/\Delta_{n-1} & \\ & & & & a_n^{-1} \end{bmatrix}, \tag{6.3}$$

where $\Delta_i = a_i a_{i+1} - b_i^2$. The elements of Q are given by (5.23), which simplifies considerably in this case.

Two test problems of variable dimensions have been used in the experiments. One is the *boundary value problem*

$$\text{minimize} \quad f(x) = \tfrac{1}{2} x^T T x - e_n^T x - h^2 \sum (\kappa \cos x_i + 2x_i), \tag{6.4}$$

where $h = 1/(n+1)$ and

$$T = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{bmatrix}.$$

An initial point $x_i^{(1)} = ih,\ i = 1, 2, \ldots, n$ has been used. Choosing $\kappa = 0$ gives rise to a quadratic function that is useful for testing purposes. The problem is otherwise nonquadratic and the value $\kappa = 1$ has been used. The second test problem is the *chained Rosenbrock problem*

$$\text{(6.5)} \qquad \text{minimize} \qquad f(x) = \sum_{i=1}^{n-1} 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2,$$

where n is even. The solution of this problem is $x^* = (1, 1, \ldots, 1)^T$ and the usual initial point is $x^{(1)} = (-1.2, 1, -1.2, 1, \ldots, -1.2, 1)^T$. However, this initial point sometimes leads to the location of a local minimum that exists in the vicinity of $x_1 = -1$ with $f(x) \simeq 4$. Thus the initial point $x^{(1)} = 0$ has been used that avoids these difficulties and does not appear to make the problem easier.

The new sparse update (spqn) is implemented in a very crude way. A standard quasi-Newton code is used with two-sided Wolfe–Powell conditions in the line search (e.g., [4]) using the parameters $\rho = 0.01$ and $\sigma = 0.1$. The inner (dual) iteration is implemented with an Armijo-type line search with cutback factor 0.25. The initial step in the dual line search is either 1, if feasible ($B > 0$), or otherwise 0.9 of the distance to the step, which would make $H_{ii} = 0$. If B does not become positive definite then the Armijo cutback is used. The inner iteration is terminated when the predicted increase in ϕ is less than $10^{-7}n$. A comparison is made with other methods that do not take advantage of the tridiagonal structure, including a standard BFGS code, an implementation of Nocedal's low storage method [8] based on five stored difference pairs, and an implementation of the Polak–Ribiere conjugate gradient method. The same line search is used for all these methods. Results are also given for Newton's method (exact Hessian) and the LANCELOT method which do take advantage of the tridiagonal structure. The line search in Newton's method uses the parameter $\sigma = 0.9$.

The outcome of these tests is set out in Tables 1–4. All the methods are able to solve the problems, although less accuracy is obtained by the conjugate gradient method, particularly for the larger problems. In all cases the methods that make use of the tridiagonal structure of the Hessian obtain significantly better results. The most marked difference is for the $n = 100$ boundary value problem, both for $\kappa = 0$ and $\kappa = 1$. These problems are quadratic or nearly so, and once a good approximation to the Hessian is obtained then the problem is effectively solved. It is clear that the new update enables the Hessian to be approximated very rapidly because of the relatively few elements in the sparse matrix that must be determined. The same is true for the LANCELOT code. In passing, it is also interesting to notice that Nocedal's limited memory method is comparable with BFGS, even for near quadratic problems. The chained Rosenbrock problem is highly nonlinear and the improvement obtained by the new update is less spectacular. A possible interpretation of this is that the local Hessian matrix for the problem changes markedly as the iterations proceed towards the solution. The new update is likely to get a good estimate of the local Hessian more quickly than say BFGS, but both methods are efficient at revising this estimate as the local Hessian changes.

The results for the LANCELOT code on the chained Rosenbrock problem (I am indebted to Nick Gould for providing these results.) lie between those for the new update and for Newton's method. The LANCELOT code assumes that the objective function has the form

$$\text{(6.6)} \qquad f(x) = \sum_i g_i \left(\textstyle\sum_j f_j(x) + a_i^T x - b_i\right).$$

TABLE 1
Results for boundary value problem $n = 10$.

Method	κ	f^*	Number of iterations	Function evaluations	Gradient evaluations
spqn	0	−0.552217	5	10	10
	1	−0.615442	7	12	12
bfgs	0	−0.552216	6	12	11
	1	−0.615442	11	21	20
nocedal	0	−0.552216	7	12	12
	1	−0.615441	11	23	22
PR-cg	0	−0.552217	9	24	12
	1	−0.615442	13	34	19
lancelot	0	−0.552216	1	2	2
	1	−0.615441	5	6	6
newton	0	−0.552216	2	3	3
	1	−0.615441	2	3	3

TABLE 2
Results for boundary value problem $n = 100$.

Method	κ	f^*	Number of iterations	Function evaluations	Gradient evaluations
spqn	0	−0.506503	3	10	9
	1	−0.514007	5	15	14
bfgs	0	−0.506503	50	68	64
	1	−0.514002	48	71	62
nocedal	0	−0.506501	48	133	111
	1	−0.514000	48	149	123
PR-cg	0	−0.506498	48	78	61
	1	−0.513997	49	101	75
lancelot	0	−0.506502	1	2	2
	1	−0.514005	2	3	3
newton	0	−0.506503	2	3	3
	1	−0.514007	2	3	3

This is referred to as *group partial separability* and it extends the ideas of Griewank and Toint referred to in (1.21). Using group functions that are squares enables the quadratic parts of (6.4) and (6.5) to be specified exactly. Thus the only nonlinear terms that need to be approximated are the $\cos x_i$ term in (6.4) and the x_i^2 term in the first bracket of (6.5). This enables the LANCELOT code to approach more closely the performance of Newton's method, whilst only requiring first derivatives of the nonlinear functions $g_i(\alpha)$ and $f_j(x)$. The new update given in this paper requires less information from the user than LANCELOT, only requiring first derivatives of $f(x)$, along with the sparsity pattern of G. Thus the comparative performance of the

TABLE 3
Results for chained Rosenbrock problem $n = 10$.

Method	f^*	Number of iterations	Function evaluations	Gradient evaluations
spqn	$1.4_{10}-9$	37	91	78
bfgs	$3.0_{10}-8$	52	123	104
nocedal	$3.2_{10}-8$	53	109	102
PR-cg	$2.0_{10}-7$	107	218	186
lancelot	$4.5_{10}-21$	34	35	30
newton	$3.9_{10}-9$	24	29	27

TABLE 4
Results for chained Rosenbrock problem $n = 100$.

Method	f^*	Number of iterations	Function evaluations	Gradient evaluations
spqn	$1.0_{10}-10$	290	727	648
bfgs	$2.7_{10}-8$	426	920	802
nocedal	$5.3_{10}-8$	479	881	871
PR-cg	$3.0_{10}-6$	652	1107	1083
lancelot	$1.7_{10}-13$	227	228	191
newton	$1.8_{10}-10$	161	190	182

new update is very much what might reasonably have been hoped for.

On the other hand, the cost of calculating the new update is significant and dominates the computation time for test problems such as these that are readily evaluated. Even when $x^{(k)}$ is close to x^*, it is observed that about four iterations are required to solve the dual problem (4.4) and up to a dozen or more on the early iterates of the outer problem. The inner iteration is solved to high accuracy and the second order convergence associated with Newton's method is observed, which gives confidence in the correctness of the calculations. However, a lower accuracy solution to the dual might be more effective overall. The Armijo line search also shows up rather poorly, particularly when the singularity on the boundary of $B > 0$ comes into play. A special purpose line search for the dual would undoubtedly have improved the overall performance. This would involve determining precisely the step to the boundary in the dual line search. Because a rank-two update of each Markowitz submatrix H_i is involved, the solution of a quadratic equation for each distinct Markowitz submatrix is required (even for a general sparsity pattern). Another possibility for improving the performance of the dual iteration is to seek some quick method for estimating a good initial value $\lambda^{(1)}$ rather than the value $\lambda^{(1)} = 0$ used here.

It is disappointing not to have observed that only one dual step is required when $x^{(k)}$ is asymptotically close to x^*. I had expected that this would be the case as the Hessian approximation converges. Possibly the outer iteration converges before the phenomenon becomes apparent.

One referee correctly points out that the number of function and gradient calls for bfgs, nocedal and sqpn in the tables could be appreciably improved by using a weaker

tolerance (e.g., $\sigma = 0.9$) in the line search, albeit at the expense of some increase in the number of iterations. The best choice of σ for any particular problem depends on the cost of evaluating the function and gradient. However, I would agree that $\sigma = 0.9$ is probably a better choice for the default option for these methods.

7. Primal algorithms. In this section the possibility of using primal algorithms to solve the problem in (2.1)–(2.4) is considered. If the positive definite constraint is inactive, then this problem is one with linear constraints on the elements of B and a nonquadratic objective function. Generalised elimination techniques (e.g., [4]) can therefore be used to provide a reduced unconstrained minimization problem for which there are various possible methods of solution. The basis vectors calculated in the elimination process are shown to have a particularly nice interpretation.

It is convenient to express $B = B^{(k)} + E$, where E is the change in $B^{(k)}$. Then the symmetry and sparsity constraints (2.2) and (2.3) can be immediately satisfied by choosing only independent nonzero elements in E as the unknowns. For example, a 4×4 tridiagonal matrix can be expressed as

$$E = \begin{bmatrix} x_1 & x_5 & & \\ x_5 & x_2 & x_6 & \\ & x_6 & x_3 & x_7 \\ & & x_7 & x_4 \end{bmatrix}. \tag{7.1}$$

The system $(B^{(k)} + E)\delta = \gamma$ is then rearranged as an underdetermined system

$$A^T x = b, \tag{7.2}$$

where $b = \gamma - B^{(k)}\delta$, and where $b \in \mathbb{R}^n$, $x \in \mathbb{R}^\tau$, and τ is the number of independent unknowns in E. In generalised elimination, A is bordered by an arbitrary matrix V so that $[A\ V]$ is nonsingular, and the inverse

$$[A\ V]^{-T} = [Y\ Z] \tag{7.3}$$

is used to define the matrices Y (having the same dimensions as A) and Z. Then the feasible region of (7.2) can be parametrized by

$$x = Yb + Zy, \tag{7.4}$$

where Yb is a feasible point of (7.2) and Zy represents an arbitrary correction in null space of A. The reduced optimization problem can therefore be expressed in terms of the vector y, which has $\tau - n$ components.

In this particular application the columns of Y and Z can be regarded as elementary $n \times n$ matrices Y_i and Z_i by scattering their elements according to the sparsity pattern of E. A particularly convenient form is obtained if the columns of V are unit vectors with unit element in positions corresponding to off-diagonal elements of E. This approach is only numerically stable[1]

if the elements $\delta_i \quad i = 1, 2, \ldots, n$ are not close to zero. To illustrate this construction, (7.1) can be rearranged in the form (7.2) as

$$A^T x = \begin{bmatrix} \delta_1 & & & & \delta_2 & & \\ & \delta_2 & & & \delta_1 & \delta_3 & \\ & & \delta_3 & & & \delta_2 & \delta_4 \\ & & & \delta_4 & & & \delta_3 \end{bmatrix} x = b. \tag{7.5}$$

[1]There are other more stable constructions for V based on Gaussian elimination with pivoting or the use of QR factors; see [4].

Using the above choice of V,

$$[A\ V] = \begin{bmatrix} \delta_1 & & & & & & \\ & \delta_2 & & & & & \\ & & \delta_3 & & & & \\ & & & \delta_4 & & & \\ \delta_2 & \delta_1 & & & 1 & & \\ & \delta_3 & \delta_2 & & & 1 & \\ & & \delta_4 & \delta_3 & & & 1 \end{bmatrix}. \tag{7.6}$$

It is readily verified from (7.3) that $Y_i = e_i \delta_i^{-1} e_i^T \quad i = 1, 2, 3, 4$, and

$$Z_1 = \begin{bmatrix} -\delta_2\delta_1^{-1} & 1 & & \\ 1 & -\delta_1\delta_2^{-1} & & \\ & & 0 & \\ & & & 0 \end{bmatrix}, \qquad Z_2 = \begin{bmatrix} 0 & & & \\ & -\delta_3\delta_2^{-1} & 1 & \\ & 1 & -\delta_2\delta_3^{-1} & \\ & & & 0 \end{bmatrix},$$

and

$$Z_3 = \begin{bmatrix} 0 & & & \\ & 0 & & \\ & & -\delta_4\delta_3^{-1} & 1 \\ & & 1 & -\delta_3\delta_4^{-1} \end{bmatrix}. \tag{7.7}$$

In general, matrices Y_i are used to construct a feasible correction, $E^{(k)}$ say, by

$$E^{(k)} = \sum Y_i b = \operatorname{diag}(b_i/\delta_i). \tag{7.8}$$

Moreover, it is readily verified that $Z_i\delta = 0$ illustrating the null space property of the Z_i matrices. Thus a parametrization of feasible matrices in (2.2)–(2.4) is given by

$$B(y) = B^{(k)} + E^{(k)} + \sum Z_i y_i. \tag{7.9}$$

For more general sparsity patterns a similar outcome is obtained and there is one Z_i matrix, having a similar structure to (7.7), for each off-diagonal element of E.

This construction is essentially given by Toint [11]. Assuming that $\delta_i \neq 0 \quad i = 1, 2, \ldots, n$, that B is irreducible, and that $\delta^T\gamma > 0$, Toint proves that a positive semi-definite matrix T with $\operatorname{rank}(T) = n - 1$ can be constructed from a linear combination of the matrices Z_i. It then follows that $\hat{B} = B^{(k)} + E^{(k)} + \alpha T$ is positive definite for sufficiently large α. This proves the existence of a positive definite matrix B that satisfies the constraints (2.2)–(2.4). A consequence of Toint's result is that the primal always has a solution under these conditions. The implications of this result are discussed in more detail in §8.

If a primal algorithm is to be used, then it is advantageous if the initial iterate $B^{(1)}$ is a positive definite matrix. This enables ψ to be used as a merit function to force convergence. Also the barrier function property of ψ ensures that all subsequent iterates stay positive definite. A possible choice of $B^{(1)}$ is the matrix $\hat{B}$ referred to in the previous paragraph.

In constructing an efficient algorithm, it is important to get at least first derivatives of the reduced objective function $\psi(B(y))$ derived from (2.1) and (7.9). Differentiating ψ with respect to B_{ij} as in §2 and using the chain rule, it follows that

$$\frac{\partial \psi}{\partial y_i} = \operatorname{trace}(Z_i(H^{(k)} - H)). \tag{7.10}$$

Because $\mathcal{G}(Z_i) = Z_i$, it follows that only $\mathcal{G}(H^{(k)} - H)$ is required in (7.10). As mentioned in §3, Duff, Erisman, and Reid [3] indicate that the elements of $\mathcal{G}(H)$ can be determined efficiently from factors of B if assumption (1.23) holds. Thus calculation of the reduced gradient of ψ is not unduly expensive. Note also that the form of (7.10) confirms the characterisation result given in (2.5) since $\operatorname{trace}(\mathcal{G}(\delta\lambda^T + \lambda\delta^T)Z_i) = 0$ showing that if B is derived from (2.5) then it is a stationary point of the primal (and a solution if B is positive definite).

The next stage is to look at the Hessian matrix of $\psi(y)$. It follows using (5.4) that

$$\frac{\partial^2\psi}{\partial y_j \partial y_i} = -\operatorname{trace}\left(Z_i \frac{\partial H}{\partial y_j}\right) = \operatorname{trace}(Z_i H Z_j H). \tag{7.11}$$

However, it seems unlikely that the determination of this matrix and its use in a primal Newton method will be profitable. First, the reduced primal has $\tau - n$ variables and this could be significantly larger than n. Also the reduced Hessian may not be all that sparse. Perhaps the most likely option is to calculate an approximate solution of the update problem, using a few steps of preconditioned conjugate gradients with a diagonal or perhaps tridiagonal matrix derived from (7.11) as preconditioner.

8. Stability issues and discussion. This section takes up the issue of the numerical stability of the sparse positive definite update. An example of Sorensen [9] highlights a potential difficulty of sparse positive definite updates. A possible solution to these difficulties is suggested. A conjecture relating the new update to partially separable optimization is discussed and possibilities for further work are suggested.

The assumption $\delta_i \neq 0$ $i = 1, 2, \ldots, n$ used by Toint (see §7) is not there to simplify the proof, but is symptomatic of a serious difficulty that can arise when B is sparse and $\delta_i = 0$. This is made clear by Sorensen [9] who essentially cites the following example in which B is required to solve

$$\begin{bmatrix} a & b & \\ b & c & * \\ & * & * \end{bmatrix} \begin{pmatrix} -1 \\ \varepsilon \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix}. \tag{8.1}$$

When $\varepsilon \neq 0$ the first equation implies $b = (a+1)/\varepsilon$ and $a > 0$ is required for positive definiteness. Thus b grows without limit as ε goes to zero. Moreover, the inequality $ac > b^2$ implies that ac increases like ε^{-2} so the rate of growth is quadratic. If $\varepsilon = 0$, then the only solution has $a = -1$ and there does not exist a positive definite update. Yet $\delta^T\gamma = 1 > 0$.

Both types of algorithm described in this paper fail on this example when $\varepsilon = 0$. The primal problem has no feasible point so cannot be started. Because the primal is infeasible, so the dual is unbounded, and the dual iteration is seen to cause both λ and $\phi(\lambda)$ to increase without bound. At the same time both $B(\lambda)$ and $H(\lambda)$ increase without bound. The most growth is seen to occur in the ith diagonal element of B. In addition, both algorithms exhibit ill conditioning as $\varepsilon \to 0$. In the primal, the matrices Y_i and Z_i become arbitrarily large giving similarly large B matrices in the calculation. The dual iteration again exhibits large growth in λ, B, H, and $\psi(\lambda)$.

These phenomena can be seen in the following example derived from (8.1). Let δ and γ be as in (8.1), let $B^{(1)} = I$, and let $B^{(2)}$ be tridiagonal. A short calculation

using (7.10) indicates that the solution has the form

$$
\begin{aligned}
B &= \begin{bmatrix} 3 & 4\varepsilon^{-1} & \\ 4\varepsilon^{-1} & 8\varepsilon^{-2} & -4\varepsilon^{-1} \\ & -4\varepsilon^{-1} & 6 \end{bmatrix} (1+O(\varepsilon^2)), \\
H &= \begin{bmatrix} 16\varepsilon^{-2} & -12\varepsilon^{-1} & -8\varepsilon^{-2} \\ -12\varepsilon^{-1} & 9 & 6\varepsilon^{-1} \\ -8\varepsilon^{-2} & 6\varepsilon^{-1} & 4\varepsilon^{-2} \end{bmatrix} (1+O(\varepsilon^2))
\end{aligned} \tag{8.2}
$$

with $\lambda = (-8\varepsilon^{-2},\ 4\varepsilon^{-1},\ 2\varepsilon^{-2})^T(1+O(\varepsilon^2))$. The increasing growth as $\varepsilon \to 0$ is readily observed.

Some possible ways of avoiding such difficulties are now discussed. It is important to realise that such a situation is only likely to occur when the average Hessian matrix $\bar{G}$ in (1.6) is indefinite or ill conditioned. Because of (1.5), $\bar{G}$ is always feasible in the primal and, if $\bar{G}$ is positive definite and well-behaved, then $\delta^T\gamma/\delta^T\delta$ is not close to zero. It follows that if $B^{(k)}$ is well-behaved, then there is no possibility of serious growth in $\psi(B^{(k)}H)$ and hence $B^{(k+1)}$ is well-behaved. Even if $\bar{G}$ is indefinite then a satisfactory update may yet be obtained (this happens for example in the dense case). Thus it may be that these difficulties arise relatively infrequently and an ad hoc solution may be adequate. However, when difficulties do arise then their effects are severe due to the ε^{-2} growth. In addition it is likely that no such growth is evident in $\bar{G}$ when, for example, it is indefinite. Thus any heuristic should avoid letting B and H grow large. One possibility is to skip the update if some δ_i is close to zero, but this precludes a useful update if $\bar{G}$ is well-behaved. Another possibility is to make no change to row/column i of B. This may decouple two parts of the matrix (as for (8.1)) and require $\delta^T\gamma > 0$ for each part. It may also cause organisational problems by changing the effective sparsity pattern. The solution that currently appeals the most is simply to impose an upper limit on the size of B and H, and to abort the dual iteration if either of these upper limits are exceeded. It is worth pointing out that if an upper limit on the trace of both B and H is imposed, then it is readily deduced that $cond(B)$ is bounded and it follows (see [4], p.31) that the resulting quasi-Newton method is globally convergent. Of course, the user does not find it easy to set such upper limits and too small a value might preclude superlinear convergence. Further experience of these situations would be valuable and might lead to improved heuristics.

As it stands, the dual iteration (4.1) does not have much in common with the calculation involved in the BFGS update (1.8). For example in the dense case, a unit step of (4.1) does not provide the BFGS correction. Experience with some simple cases indicates that it might be possible to express the update as

$$
B_{(i)}^{(k+1)} = \text{bfgs}(B_{(i)}^{(k)}, \delta_{[i]}, \gamma_{(i)}). \tag{8.3}
$$

The matrices $B_{(i)}$ derive from an additive decomposition of B corresponding to distinct Markowitz submatrices (that is $B = \sum B_{(i)}$). Terms in which a Markowitz submatrix is a submatrix of another Markowitz submatrix would be excluded. A corresponding decomposition $\gamma_{(i)}$ for which $\sum \gamma_{(i)} = \gamma$ is also required. The updates in (8.3) require that the scalar products

$$
\delta_{[i]}^T \gamma_{(i)} > 0 \tag{8.4}
$$

are all positive. The existence of vectors $\gamma_{(i)}$ for which this holds is related to Toint's condition that $\delta_i \neq 0 \quad i = 1, 2, \ldots, n$. For example it is clear that this property

cannot be attained in (8.1) when $\varepsilon = 0$. The outcome in (8.3) would have the flavour of a partially separable update but with the blocks being determined by the result of fill-in in the LDL^T factors rather than being prescribed by the user. At present, however, it is not clear how the decomposition of γ would be determined.

A related problem to the one considered in this paper arises if the structural constraints on B are expressed as

$$B_{ij} = \beta_{ij} \quad \forall (i,j) \in \mathcal{S}, \tag{8.5}$$

where the β_{ij} are known values of the true Hessian that are independent of x but might be nonzero. For example, in the boundary value problem (6.4), one would require $B_{i,i+1} = -1$. To some extent this problem can be transformed by taking the product $\bar{B}\delta$ over to the other side of the quasi-Newton equation (2.3), where $\bar{B}$ denotes the matrix with entries β_{ij} for $(i,j) \in \mathcal{S}$ and zero otherwise. Condition (2.5) is then valid for this modified problem, but there is a difference in that B must be positive definite and not $B - \bar{B}$. Also the possibility of instability or nonexistence of the update may be compounded as there are fewer elements to adjust. For example, in the case of (6.4) the modified problem decomposes into distinct 1×1 diagonal blocks and an update is immediately determined, which may or may not correspond to a positive definite B. On the other hand there is the possibility of determining the unknown elements more rapidly, and the argument used earlier in the section regarding $\bar{G}$ might indicate that a successful update will often be obtained. Again some practical experience is called for.

In summary, it is felt that an update of some potential interest has been suggested in this paper, although further development is required before the idea can be incorporated into production software. Possible areas of future work include the following.

- Better implementation of the dual line search;
- Implementation and numerical experience for band matrices and more general sparsity patterns (overlapping blocks, skyline, random (subject to (1.23)));
- Low accuracy solution of dual;
- Heuristics for alleviating ill-conditioning when some δ_i is close to zero;
- Primal methods;
- Alternative ways of computing the update such as (8.3);
- Relationship to partially separable approach;
- Theory of obtaining B from $\mathcal{G}(H)$ when (1.23) does not hold;
- Superlinear convergence of $x^{(k)} \to x^*$;
- Experience with $B_{ij} = \beta_{ij}$ problems;
- Use in an NLP context.

Most of these have already been discussed at some point in this paper.

REFERENCES

[1] R. H. Byrd and J. Nocedal, *A tool for the analysis of quasi-Newton methods with application to unconstrained minimization*, SIAM J. Numer. Anal., 26 (1989), pp. 727–739.

[2] T. F. Coleman, *Large Sparse Numerical Optimization*, Lecture Notes in Computer Science 165, Springer-Verlag, Berlin, 1984.

[3] I. S. Duff, A. M. Erisman, and J. K. Reid, *Direct Methods for Sparse Matrices*, Clarendon Press, Oxford, 1986.

[4] R. Fletcher, *Practical Methods of Optimization*, 2nd. ed., John Wiley, Chichester, 1987.

[5] _____ , *A new result for quasi-Newton formulae*, SIAM J. Optim., 1 (1991), pp. 18–21.

[6] D. GOLDFARB, *A family of variable metric methods derived by variational means*, Maths. Comput., 24 (1970), pp. 23–26.

[7] A. GRIEWANK AND PH. L. TOINT, *Partitioned variable metric updates for large structured optimization problems*, Numer. Math., 39 (1982), pp. 429–448.

[8] J. NOCEDAL, *Updating quasi-Newton matrices with limited storage*, Math. Comput., 35 (1980), pp. 773–782.

[9] D. C. SORENSEN, *Collinear scaling and sequential estimation in sparse optimization algorithms*, Math. Programming Study, 18 (1982), pp. 135–159.

[10] PH. L. TOINT, *On sparse and symmetric updating subject to a linear equation*, Maths. Comput., 31 (1977), pp. 954–961.

[11] _____ , *A note on sparsity exploiting quasi-Newton methods*, Math. Programming, 21 (1981), pp. 172–181.

SIAM J. OPTIMIZATION
Vol. 5, No. 1, pp. 219–230, February 1995

011

TRUST REGION ALGORITHMS FOR SOLVING NONSMOOTH EQUATIONS*

LIQUN QI†

This paper is dedicated to Professor R. Tyrrell Rockafellar on the occasion of his 60th birthday.

Abstract. Two globally convergent trust region algorithms are presented for solving nonsmooth equations, where the functions are only locally Lipschitzian. The first algorithm is an extension of the classic Levenberg–Marquardt method by approximating the locally Lipschitzian function with a smooth function and using the derivative of the smooth function in the algorithm wherever a derivative is needed. Global convergence for this algorithm is established under a regular condition. In the second algorithm, successive smooth approximation functions and their derivatives are used. Global convergence for the second algorithm is established under mild assumptions. Both objective functions of subproblems of these two algorithms are quadratic functions.

Key words. nonsmoothness, algorithm, approximation, convergence

AMS subject classifications. 90C30, 90C33

1. Introduction.

In recent years, there is growing interest in solving nonsmooth equations [2]–[4], [10], [12]–[19], [24]–[28], [32]–[36]. Nonsmooth equations arise from nonlinear complementarity, variational inequality, nonlinear programming, the maximal monotone operator problem, nonsmooth partial differential equations, the nonsmooth compact fixed point problem, and the Newton method for the complex eigenvalue problem. See [28], [3], and other references listed above. In general, the nonsmooth equation problem is to solve

$$F(x) = 0, \tag{1.1}$$

where $F : \Re^n \to \Re^n$ is locally Lipschitzian. Various nonsmooth variants of Newton's methods [3], [12], [13], [15]–[18], [24], [25], [27], [32], [34]–[36], quasi-Newton methods [2]–[4], [14], [15], [19], and Gauss–Newton methods [12], [27], [28], have been proposed and studied. Local superlinear convergence has been established for some of these methods [2]–[4], [14]–[17], [24], [25], [32], [34]–[36]. A characterization for superlinear convergence was given in [28]. Global convergence was established via line search [12], [24], [25], [27], [28], [33], or path search [35].

In this paper, we present a globally convergent trust region algorithm for solving the nonsmooth equation (1.1). In a certain sense, this algorithm is a nonsmooth version of the Levenberg–Marquardt method.

The classic Levenberg–Marquardt method is for solving smooth nonlinear least squares and smooth nonlinear equations. The nonlinear least squares problem is

$$\min f(x) = \frac{1}{2} F(x)^T F(x),$$

where $F : \Re^n \to \Re^m$. If $m = n$, it is equivalent to solving the nonlinear equation problem (1.1). In the smooth case, the Levenberg–Marquardt method can be viewed

*Received by the editors August 13, 1992; accepted for publication (in revised form) November 9, 1993. This research was supported by the Australian Research Council.

†School of Mathematics, The University of New South Wales, Sydney, New South Wales 2052, Australia (qi@solution.maths.unsw.edu.au).

as a method for generating a sequence $\{x_k\}$ of iterates where the step d_k between iterates is a solution to the problem

$$\min\{\|F(x_k) + F'(x_k)d\|_2 : \|d\| \leq \Delta_k\} \tag{1.2}$$

for some bound $\Delta_k > 0$. The norm $\|\cdot\|$ is arbitrary but it is usually chosen as the 2-norm because, for this choice, Marquardt proved if the step d_k is determined by solving the linear system

$$(F'(x_k)^T F'(x_k) + \lambda_k I)d_k = -F'(x_k)^T F(x_k)$$

with some parameter $\lambda_k \geq 0$, then d_k solves (1.2) with $\Delta_k = \|d_k\|$. Trust region strategies are applied to adjust Δ_k. See [7], [20], and [21]. Global convergence results were established in [20], [23], and [29].

In [8], $\|\cdot\|_1$ was used in the objective function while $\|\cdot\|_\infty$ was used in the constraint of (1.2). In [11], two arbitrary norms are used in these two places. In the other sections of our paper, all norms are 2-norms, and we only consider the case of (1.1), i.e., $m = n$.

In the nonsmooth case, $F'(x_k)$ may not exist. However, in many cases, one may decompose F into $F = p + q$, where $p : \Re^n \to \Re^n$ is smooth, $q : \Re^n \to \Re^n$ is locally Lipschitzian and relatively small. We call such a decomposition a smooth plus nonsmooth (SPN) decomposition. In a certain sense, q can be regarded as the perturbation to the system. Two examples of SPN decomposition are given in §2. For more examples, see [2]–[4]. We now use

$$\min\{\|F(x_k) + p'(x_k)d\|_2 : \|d\| \leq \Delta_k\} \tag{1.3}$$

to replace (1.2). In §2, we present such an algorithm and introduce a regularity condition for the SPN decomposition of F. Global convergence for that algorithm is established under such a regularity condition in §3. This regularity condition is somehow restrictive. It only holds when the perturbation is relatively "mild." We seek some improvements to that algorithm in §4. In some cases, one may decompose F into $F = p_k + q_k$, where $p_k : \Re^n \to \Re^n$ is smooth, $q_k : \Re^n \to \Re^n$ uniformly tends to zero as k tends to infinity. In §4, we present an algorithm by replacing (1.2) with

$$\min\{\|p_k(x_k) + p_k'(x_k)d\|_2 : \|d\| \leq \Delta_k\}. \tag{1.4}$$

Global convergence for this algorithm is established under mild conditions. Finally, in §5, we give a numerical example for the second algorithm.

One merit of our algorithms is that the objective functions of (1.3) and (1.4) are quadratic. If we use $\|\cdot\|_\infty$ in the constraints of (1.3) and (1.4) (for this change, only minor modifications are needed for proofs), then (1.3) and (1.4) are merely convex quadratic programs, which can be solved by some general subroutines.

2. Algorithm 1 and regular SPN decomposition. In this section, we use $F_i, p_i, q_i, \ldots$, to denote component functions of $F, p, q, \ldots$. This is different from other sections. For example, in this section, $p_i : \Re^n \to \Re$ and is the ith component function of p, while in other sections, $p_k : \Re^n \to \Re^n$.

Consider (1.1) where $F : \Re^n \to \Re^n$ is locally Lipschitzian. SPN decompositions of F naturally exist for nonsmooth equations arising in nonsmooth partial differential equations [2]–[4]. For example, consider the nonsmooth partial differential equation

$$-\Delta u + \psi(u) = 0$$

in a domain of the two-dimensional space. Assume that the value of u is given at the boundary of that domain. Usually, ψ, which is continuous but not smooth, reflects the perturbation to the system. Discretizing the partial differential equation by a finite difference method or a finite element method, we obtain a system of nonsmooth equations

$$F(x) = Ax + q(x) = 0$$

for $x \in \Re^n$, where A is an $n \times n$ matrix, q is continuous but not differentiable in $\Re^n$ [22]. Let $p(x) = Ax$. We have an SPN decomposition of $F = p + q$. Here, p represents the Laplace operator and q represents the nonsmooth perturbation ψ. We see that q is locally Lipschitzian but nonsmooth. For $x \in \Re^n$ and $r > 0$, let $B(x,r) = \{y \in \Re^n : \|y - x\| \leq r\}$. In general, for a locally Lipschitzian function $q : \Re^n \to \Re^n$, define the limiting Lipschitzian constant of q at x by

$$l_q(x) = \lim_{r \downarrow 0} \sup_{\substack{y,z \in B(x,r) \\ y \neq z}} \frac{\|q(y) - q(z)\|}{\|y - z\|}.$$

Then $0 \leq l_q(x) < \infty$. In this example, if the perturbation ψ is relatively mild, i.e., it does not change very rapidly, then $l_q(x)$ is relatively small for all x. We will see that our Algorithm 1 converges in such a case.

For most nonsmooth equations arising in optimization, SPN decompositions do not exist naturally but, in some cases, can be constructed. For example, consider the following nonlinear complementarity problem: find $x \in \Re^n$ such that

$$g(x) \geq 0, \ h(x) \geq 0, \ g(x)^T h(x) = 0,$$

where $g, h : \Re^n \to \Re^n$ are smooth functions. This nonlinear complementarity problem is equivalent to the nonsmooth equation (1.1) with each component of F to be defined by

$$F_i(x) = \min\{g_i(x), h_i(x)\}.$$

Let $w > 0$. Define the components of $p, q : \Re^n \to \Re^n$ by

$$p_i(x) = \begin{cases} F_i(x) & \text{if } |g_i(x) - h_i(x)| \geq w, \\ \hat{p}_i(x) & \text{if } |g_i(x) - h_i(x)| < w \end{cases}$$

and

$$q_i(x) = \begin{cases} 0 & \text{if } |g_i(x) - h_i(x)| \geq w, \\ \hat{q}_i(x) & \text{if } |g_i(x) - h_i(x)| < w, \end{cases}$$

where

$$\hat{p}_i(x) = \frac{g_i(x) + h_i(x)}{2} - \frac{[g_i(x) - h_i(x)]^2 + w^2}{4w},$$

$$\hat{q}_i(x) = \frac{[|g_i(x) - h_i(x)| - w]^2}{4w}.$$

Then $F = p + q$, p is smooth and $\|q\|_\infty \leq \frac{w}{4}$. Since w is arbitrary, we can make q as small as possible. This property assures that our Algorithm 2 converges. See also [3] for this decomposition and its applications.

Let

$$f(x) = \frac{1}{2} F(x)^T F(x).$$

Let x_0 be an initial point for our algorithm. Assume that the level set $L_0 = \{x \in \Re^n : f(x) \leq f(x_0)\}$ is bounded and let $D \subset \Re^n$ be a bounded open convex set containing L_0. Let Δ_0 be the diameter of D.

We say that $F = p + q$ is a *regular* SPN *decomposition* of F on D if $p : \Re^n \to \Re^n$ is smooth and for any $x \in D$,

$$p'(x)^T F(x) \neq 0 \tag{2.1}$$

as long as $F(x) \neq 0$.

Suppose now $F = p + q$ is a regular SPN decomposition. For $x \in D$, define $R_F(x) = 0$ if $F(x) = 0$. Define

$$R_F(x) = \frac{l_q(x)\|F(x)\|}{\|p'(x)^T F(x)\|},$$

otherwise. We call $R_F(x)$ the *regularity condition number* of F at x.

We now describe our first algorithm. Let c_0, c_1, c_2, c_3, and c_4 be positive constants satisfying $c_0 \leq 1$, $c_2 < c_1 < 1$, $c_3 < 1 < c_4$.

ALGORITHM 1. At the kth iteration, if $F(x_k) = 0$, stop. Otherwise, given x_k and Δ_k, solve the subproblem

$$\min \left\{ Q_k(d) \equiv \frac{1}{2}\|F(x_k) + p'(x_k)d\|^2 : \|d\| \leq \Delta_k \right\}. \tag{2.2}$$

Assume the solution of (2.2) is d_k^*. Suppose that d_k is an *inexact* solution of (2.2) in the sense that d_k satisfies

$$f(x_k) - Q_k(d_k) \geq c_0[f(x_k) - Q_k(d_k^*)] \tag{2.3}$$

and

$$\|d_k\| \leq \Delta_k.$$

Let

$$r_k = \frac{f(x_k) - f(x_k + d_k)}{f(x_k) - Q_k(d_k)}, \tag{2.4}$$

$$x_{k+1} = \begin{cases} x_k + d_k & \text{if } r_k > c_2, \\ x_k & \text{otherwise,} \end{cases} \tag{2.5}$$

$$\Delta_{k+1} = \begin{cases} c_3\Delta_k & \text{if } r_k \leq c_2, \\ \Delta_k & \text{if } c_2 < r_k \leq c_1, \\ \min\{c_4\Delta_k, \Delta_0\} & \text{otherwise.} \end{cases} \tag{2.6}$$

Go to the $(k+1)$th step.

In the next section, we show that if $F = p + q$ is a regular SPN decomposition, then this algorithm is well defined, and that if for one accumulation point $\bar{x}$ of the sequence $\{x_k\}$,

$$R_F(\bar{x}) < \frac{c_0}{2}, \qquad c_2 < 1 - \frac{2R_F(\bar{x})}{c_0}, \tag{2.7}$$

then every accumulation point of $\{x_k\}$ is a solution of (1.1). The condition (2.7) depends upon the size of $l_q(\bar{x})$, hence is somehow strong. It holds when the perturbation is relatively mild, but does not hold in general. We improve Algorithm 1 in §4.

3. Global convergence of Algorithm 1. We now prove the global convergence of Algorithm 1. Assume that $F = p+q$ is a regular SPN decomposition. We use some known techniques in the literature about trust region algorithms [6], [9], [21], [30], [31], [38]. The proof was shortened and the conclusion was strengthened due to a referee's suggestion.

LEMMA 1. *At the kth step of Algorithm* 1, *if* $F(x_k) \neq 0$, *then*

$$f(x_k) - Q_k(d_k) \geq \frac{c_0}{2}\|p'(x_k)^T F(x_k)\| \min\left\{\Delta_k, \frac{\|p'(x_k)^T F(x_k)\|}{\|p'(x_k)\|^2}\right\}. \tag{3.1}$$

Proof. Let

$$\bar{d}_k = -\frac{\Delta_k p'(x_k)^T F(x_k)}{\|p'(x_k)^T F(x_k)\|}. \tag{3.2}$$

Then we have

$$\begin{aligned} f(x_k) - Q_k(d_k) &\geq c_0[f(x_k) - Q_k(d_k^*)] && \text{(by (2.3))} \\ &\geq c_0 \min_{0\leq\alpha\leq 1}[f(x_k) - Q_k(\alpha\bar{d}_k)] && \text{(by optimality of } d_k^*) \\ &\geq c_0 \min_{0\leq\alpha\leq 1}\left[\alpha\Delta_k\|p'(x_k)^T F(x_k)\| - \frac{1}{2}\alpha^2\|p'(x_k)\bar{d}_k\|^2\right] \\ & && \text{(by (2.2) and (3.2))} \\ &\geq c_0 \min_{0\leq\alpha\leq 1}\left[\alpha\Delta_k\|p'(x_k)^T F(x_k)\| - \frac{1}{2}\alpha^2\|p'(x_k)\|^2\Delta_k^2\right]. \end{aligned}$$

By this and (2.1), we get (3.1). □

Remark 1. By (2.1), the right-hand side of (3.1) is always positive if $F(x_k) \neq 0$. Hence $d_k \neq 0$ if $F(x_k) \neq 0$. This shows that Algorithm 1 is well defined.

If $F(x_k) = 0$, then Algorithm 1 stops and x_k is a solution of (1.1). Hence, we may assume $F(x_k) \neq 0$ for all k.

THEOREM 1. *Suppose that* $\{x_k\}$ *is the sequence generated by Algorithm* 1. *Then either* (i) *every accumulation point of* $\{x_k\}$ *is a solution of* (1.1); *or* (ii) $\{x_k\}$ *converges to a point* $\bar{x}$ *such that* $F(\bar{x}) \neq 0$.

Proof. Since L_0 is bounded, $\{x_k\}$ is bounded. Hence $\{x_k\}$ has at least one accumulation point $\bar{x}$. If $F(\bar{x}) = 0$, then $f(\bar{x}) = 0$. Since $0 \leq f(x_{k+1}) \leq f(x_k)$ for all k, we have

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

This implies case (i).

By (2.1), $p'(x_k) \neq 0$ for all k. Assume that $F(\bar{x}) \neq 0$. By above discussion, no accumulation point of $\{x_k\}$ is a solution of (1.1). This implies that

$$\liminf_{k\to\infty} \|p'(x_k)^T F(x_k)\| > 0 \tag{3.3}$$

and

$$\liminf_{k\to\infty} \frac{\|p'(x_k)^T F(x_k)\|}{\|p'(x_k)\|^2} > 0. \tag{3.4}$$

By (3.1), (2.3), (2.4), and (2.5), we have that

$$\sum_{k\in K} \|p'(x_k)^T F(x_k)\| \min\left\{\Delta_k, \frac{\|p'(x_k)^T F(x_k)\|}{\|p'(x_k)\|^2}\right\} < \infty, \tag{3.5}$$

where $K = \{k : r_k > c_2\}$ is the set of successful iterations. By (3.3), (3.4), and (3.5), we have

$$\sum_{k\in K} \Delta_k < \infty. \tag{3.6}$$

By (3.6) and the fact that a subsequence of $\{x_k\}$ converges to $\bar{x}$, we see that

$$\lim_{k\to\infty} x_k = \bar{x}. \tag{3.7}$$

This is case (ii). The theorem is proved. □

To avoid case (ii), we need a stronger condition.

THEOREM 2. *Suppose that $\bar{x}$ is an accumulation point of $\{x_k\}$ and (2.7) holds. Then $F(\bar{x}) = 0$ and every accumulation point of $\{x_k\}$ is a solution of (1.1).*

Proof. Suppose that $F(\bar{x}) \neq 0$. By Theorem 1, (3.6) and (3.7) hold. We have

(3.8)

$$\begin{aligned}
&f(x_k + d_k) - Q_k(d_k) \\
&= \frac{1}{2}\|F(x_k + d_k)\|^2 - \frac{1}{2}\|F(x_k)\|^2 - (p'(x_k)^T F(x_k))^T d_k - \frac{1}{2}\|p'(x_k)d_k\|^2 \\
&\leq \frac{1}{2}\|F(x_k + d_k) - F(x_k)\|^2 + F(x_k)^T[F(x_k + d_k) - F(x_k) - p'(x_k)d_k] \\
&= O(\|d_k\|^2) + F(x_k)^T[p(x_k + d_k) - p(x_k) - p'(x_k)d_k] + F(x_k)^T[q(x_k + d_k) - q(x_k)] \\
&= F(x_k)^T[q(x_k + d_k) - q(x_k)] + O(\Delta_k^2) \\
&\leq \|F(x_k)\| l_q(\bar{x})\Delta_k + o(\Delta_k).
\end{aligned}$$

Since $p'(\bar{x})^T F(\bar{x}) \neq 0$, by (3.6) and (3.7), for all large k,

$$\Delta_k \|p'(x_k)\|^2 \leq \|p'(x_k)^T F(x_k)\|.$$

By this and (3.1), we have

$$f(x_k) - Q_k(d_k) \geq \frac{c_0\Delta_k}{2}\|p'(x_k)^T F(x_k)\|. \tag{3.9}$$

By (3.8) and (3.9), we have

$$\frac{f(x_k + d_k) - Q_k(d_k)}{f(x_k) - Q_k(d_k)} \leq \frac{2\|F(x_k)\| l_q(\bar{x}) + o(1)}{c_0\|p'(x_k)^T F(x_k)\|}. \tag{3.10}$$

By (2.7) and (3.10), for all large k,

$$r_k \equiv \frac{f(x_k) - f(x_k + d_k)}{f(x_k) - Q_k(d_k)} > c_2.$$

Consequently, by (2.6), $\Delta_{k+1} \geq \Delta_k$ for all large k. This contradicts (3.6). Hence $F(\bar{x}) = 0$ and the conclusion follows Theorem 1. □

Remark 2. If F is smooth, we may let $p = F$ and $q = 0$, then $l_q(x) \equiv 0$. In this case, $\|F(x_k)\| \to 0$ as long as (2.1) holds for all x. We will use this fact in §4.

4. Algorithm 2 and successive approximation. The success of Algorithm 1 depends upon (2.7). If $R_F(x) < \frac{1}{2}$ for all $x \in D$, we may choose c_0 and c_2 to satisfy this requirement. In the general case, we need to explore more powerful methods.

In §2, for the example of the nonsmooth equation arising in nonlinear complementarity problem, we see that we can decompose $F = p + q$ such that p is smooth and q is as small as we wish. Actually, theoretically, this is always possible according to approximation theory. But in this example, it is computationally possible, too. Hence in some cases, we may decompose

$$F = p_k + q_k,$$

such that p_k is smooth and uniformly converges to F. Notice that $p_k : \Re^n \to \Re^n$ here, while in §2 we use p_i to denote the ith component function of p.

In the following, we denote

$$\|q_k\| = \sup\{\|q_k(x)\| : x \in \Re^n\}.$$

Let c_0, c_1, c_2, c_3, and c_4 be positive constants satisfying $c_0 \leq 1$, $c_2 < c_1 < 1$, $c_3 < 1 < c_4$ as before. Let $1 < c_5 \leq 2$.

Let x_0 be an initial point for our algorithm. Assume that $F(x_0) \neq 0$. Let $F = p_0 + q_0$ such that p_0 is smooth and $\|q_0\| \leq c_5^{-1}\|F(x_0)\|$. Let

$$f_k(x) = \frac{1}{2} p_k(x)^T p_k(x),$$

where $F = p_k + q_k$ for $k \geq 1$ will be constructed later. Assume that the level set $L_0 = \{x \in \Re^n : \|F(x)\| \leq 3\|F(x_0)\|\}$ is bounded and let $D \subset \Re^n$ be a bounded open convex set containing L_0. Let Δ_0 be the diameter of D.

We call $F = p_k + q_k$ a *relatively regular* SPN *decomposition* if $p_k : \Re^n \to \Re^n$ is smooth and for all $x \in D$,

$$p_k'(x)^T p_k(x) \neq 0 \tag{4.1}$$

as long as $F(x) \neq 0$.

Our second algorithm is as follows.

ALGORITHM 2. At the kth iteration, given x_k, Δ_k and $F = p_k + q_k$, solve the subproblem

$$\min\left\{Q_k(d) \equiv \frac{1}{2}\|p_k(x_k) + p_k'(x_k)d\|^2 : \|d\| \leq \Delta_k\right\}. \tag{4.2}$$

Assume the solution of (4.2) is d_k^*. Suppose that d_k is an inexact solution of (4.2) in the sense that d_k satisfies

$$f_k(x_k) - Q_k(d_k) \geq c_0[f_k(x_k) - Q_k(d_k^*)] \tag{4.3}$$

and

$$\|d_k\| \le \Delta_k.$$

Let

$$r_k = \frac{f_k(x_k) - f_k(x_k + d_k)}{f_k(x_k) - Q_k(d_k)}, \tag{4.4}$$

$$x_{k+1} = \begin{cases} x_k + d_k & \text{if } r_k > c_2, \\ x_k & \text{otherwise,} \end{cases} \tag{4.5}$$

$$\Delta_{k+1} = \begin{cases} c_3\Delta_k & \text{if } r_k \le c_2, \\ \Delta_k & \text{if } c_2 < r_k \le c_1, \\ \min\{c_4\Delta_k, \Delta_0\} & \text{otherwise.} \end{cases} \tag{4.6}$$

If $F(x_k) = 0$, stop. Otherwise, if $\|F(x_{k+1})\| > c_5\|q_k\|$, let $p_{k+1} \equiv p_k$ and $q_{k+1} \equiv q_k$; if $\|F(x_{k+1})\| \le c_5\|q_k\|$, construct $F = p_{k+1} + q_{k+1}$ such that it is a relatively regular SPN decomposition and

$$\|q_{k+1}\| \le \min\left\{c_5^{-1}\|F(x_{k+1})\|, \frac{1}{2}\|q_k\|\right\}. \tag{4.7}$$

Go to the $(k+1)$th step.

THEOREM 3. *In Algorithm* 2, *if at each step* $F = p_k + q_k$ *can be constructed such that it is a relatively regular* SPN *decomposition and* (4.7) *holds, then this algorithm is well defined and any accumulation point of* $\{x_k\}$ *generated by this algorithm is a solution of* (1.1).

Proof. Without loss of generality, assume that $F(x_k) \ne 0$ for all k. Let $K = \{0\} \cup \{k : \|F(x_k)\| \le c_5\|q_{k-1}\|\}$. Assume that K consists of $k_0 = 0 < k_1 < k_2 < \dots$. Let k be an arbitrary nonnegative integer. Let k_j be the largest number in K such that $k_j \le k$. Then

$$p_k = p_{k_j}, \qquad q_k = q_{k_j}.$$

We may regard iterations k_j to k of Algorithm 2 as some iterations of Algorithm 1 with $F = p = p_{k_j}, q = 0$. Let $U_j = \{x \in \Re^n : \|p_{k_j}(x)\| \le \|p_{k_j}(x_{k_j})\|\}$. For any $x \in U_j$,

$$\begin{aligned} \|F(x)\| &\le \|p_{k_j}(x)\| + \|q_{k_j}\| \\ &\le \|p_{k_j}(x_{k_j})\| + \|q_{k_j}\| \\ &\le \|F(x_{k_j})\| + 2\|q_{k_j}\|. \end{aligned} \tag{4.8}$$

If $j = 0$, then by (4.8), for any $x \in U_0$,

$$\|F(x)\| \le \|F(x_0)\| + 2\|q_0\| \le 3\|F(x_0)\|,$$

i.e., $U_0 \subseteq L_0$. If $j > 0$, then by (4.8) and (4.7), for any $x \in U_j$,

$$\begin{aligned} \|F(x)\| &\le \|F(x_{k_j})\| + 2\|q_{k_j}\| \\ &\le c_5\|q_{k_j-1}\| + \frac{1}{2^{j-1}}\|q_0\| \\ &= c_5\|q_{k_{j-1}}\| + \frac{1}{2^{j-1}}\|q_0\| \\ &\le \frac{1+c_5}{2^{j-1}}\|q_0\| \\ &\le 3\|F(x_0)\|. \end{aligned}$$

We also have $U_j \subseteq L_0$. Hence, in any case, U_j is bounded. Then this algorithm is well defined. Notice that (4.1) holds as long as $\|p_k(x)\| > (c_5 - 1)\|q_k\|$. By a consideration similar to Remark 2, we see that eventually there is $k' \geq k$ such that

$$\|p_{k'+1}(x_{k'+1})\| \leq (c_5 - 1)\|q_{k'}\|.$$

This implies that

$$\|F(x_{k'+1})\| \leq c_5\|q_{k'}\|,$$

i.e., $k' + 1 \in K$. Therefore, K is infinite. By our construction,

$$\|q_{k_j}\| \leq \frac{1}{2^j}\|q_0\|.$$

We have

$$\lim_{j\to\infty} \|q_{k_j}\| = 0 \tag{4.9}$$

and

$$\lim_{j\to\infty} \|F(x_{k_j})\| = 0. \tag{4.10}$$

Let k be an arbitrary nonnegative integer. Let k_j be the largest number in K such that $k_j \leq k$. Since we may regard iterations k_j to k of Algorithm 2 as some iterations of Algorithm 1 with $F = p = p_{k_j}, q = 0$,

$$\|p_k(x_k)\| \leq \|p_{k_j}(x_{k_j})\|.$$

Then,

$$\begin{aligned} \|F(x_k)\| &\leq \|p_k(x_k)\| + \|q_k\| \\ &\leq \|p_{k_j}(x_{k_j})\| + \|q_{k_j}\| \\ &\leq \|F(x_{k_j})\| + 2\|q_{k_j}\|. \end{aligned} \tag{4.11}$$

By (4.9), (4.10), and (4.11), we have

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

This completes the proof. □

Remark 3. Notice that the function value of f may not be monotonically decreasing for all k, though it is monotonically decreasing for $k \in K$.

Remark 4. Actually, one can combine successive approximation strategy with other globally convergent strategies for solving smooth nonlinear equations, such as line search. The proof of Theorem 3 is still true in those extensions. This results in other successive approximation methods for solving nonsmooth equations. Approximation strategy has also been used in other optimization branches, such as in stochastic programming [1], [37].

5. A numerical example. In this section, we give a numerical example to test Algorithm 2. Consider the nonlinear complementarity problem described in §2. We

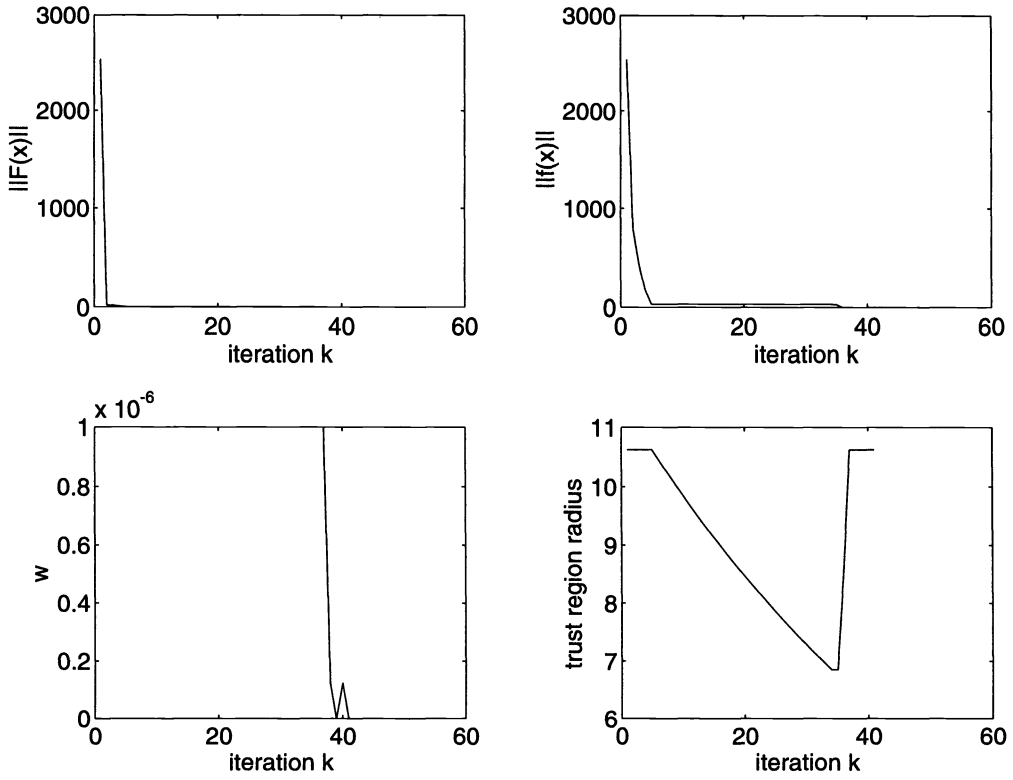


FIG. 1

use the example given in [15]. In this example, $g(x) \equiv x$ and $h : R^4 \to R^4$ is defined by

$$h(x) = \begin{pmatrix} 3x_1^2 + 2x_1x_2 + 2x_2^2 + x_3 + 3x_4 - 6 \\ 2x_1^2 + x_1 + x_2^2 + 10x_3 + 2x_4 - 2 \\ 3x_1^2 + x_1x_2 + 2x_2^2 + 2x_3 + 9x_4 - 9 \\ x_1^2 + 3x_2^2 + 2x_3 + 3x_4 - 3 \end{pmatrix}.$$

Let F be defined by g and h as in §2.

This is a degenerate nonlinear complementarity problem. At many points, the generalized Jacobian $\partial F(x)$ includes singular elements. It was shown in [3] that the generalized Newton method

$$x_{k+1} = x_k - V_k^{-1}F(x_k), \qquad V_k \in \partial F(x)$$

failed when V_k was singular at some points x_k. In Algorithm 2, we do not need an inverse of V_k. The algorithm is well defined. For showing the global convergence of Algorithm 2, we choose initial point $x_0 = (50, 50, 0, 0) + \epsilon(1, 1, 1, 1)$, where ϵ is a random number. Also we choose $\Delta_0 = 10.625$, $c_1 = 0.265, c_2 = 0.025, c_3 = 0.985, c_4 = 1.25, c_5 = 1.5, w_0 = 10^{-6}$. At the kth step, let w_k be the constant w for the SPN decomposition $F = p_k + q_k$ as described in §2. Let w_{k+1} be choosen such that (4.7) holds. In Fig. 1, we see that $||F(x_k)||$ and $f(x_k)$ are monotonically decreasing and w_k tends to zero when $||F(x_k)||$ tends to zero. Notice w_k decreases suddenly when $k = 38$.

The computation was done by Dr. Xiaojun Chen.

Acknowledgments. I am grateful to Professor Andrew R. Conn, Professor Jong-Shi Pang, Dr. Xiaojun Chen, Mr. Houyuan Jiang, and two referees for their comments.

REFERENCES

[1] J. R. BIRGE AND L. QI, *Subdifferential convergence in stochastic programming*, SIAM J. Optim., to appear.

[2] X. CHEN, *On the convergence of Broyden-like methods for nonlinear equations with nondifferentiable terms*, Ann. Institut. Statist. Math., 42 (1990), pp. 387–401.

[3] X. CHEN AND L. QI, *Parameterized Newton method and Broyden-like method for solving nonsmooth equations*, Comput. Optim. Appl., 3 (1994), pp. 157–179.

[4] X. CHEN AND T. YAMAMOTO, *On the convergence of some quasi-Newton methods for nonlinear equations with nondifferentiable operators*, Computing, 49 (1992), pp. 87–94.

[5] F. H. CLARKE, *Optimization and Nonsmooth Analysis*, Wiley, New York, 1983.

[6] J. E. DENNIS, S. B. LI, AND R. A. TAPIA, *A unified approach to global convergence of trust-region methods for nonsmooth optimization*, Mathematical Sciences Technical Report, TR 89-5, Rice University, Houston, TX, 1990, revised.

[7] J. E. DENNIS AND R. B. SCHNABEL, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[8] I. S. DUFF, J. NOCEDAL AND J. K. REID, *The use of linear programming for the solution of sparse sets of nonlinear equations*, SIAM J. Statist. Comput., 8 (1987), pp. 99–108.

[9] R. FLETCHER, *A model algorithm for composite* NDO *problem*, Math. Programming Study, 17 (1982), pp. 67–76.

[10] S. A. GABRIEL AND J.-S. PANG, *An inexact* NE/SQP *method for solving the nonlinear complementarity problem*, Comput. Optim. Appl., 1 (1992), pp. 67–92.

[11] M. EL HALLABI AND R. A. TAPIA, *A global convergence theory for arbitrary norm trust region methods for nonlinear equations*, Mathematical Sciences Technical Report, TR 87-25, Rice University, Houston, TX, 1987.

[12] S. P. HAN, J. S. PANG, AND N. RANGARAJ, *Globally convergent Newton methods for nonsmooth equations*, Math. Oper. Res., 17 (1992), pp. 586–607.

[13] P. T. HARKER AND B. XIAO, *Newton's method for the nonlinear complementarity problem: A* B*-differentiable equation approach*, Math. Programming, 48 (1990), pp. 339–357.

[14] C. M. IP AND J. KYPARISIS, *Local convergence of quasi-Newton methods for* B*-differentiable equations*, Math. Programming, 56 (1992), pp. 71–89.

[15] 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.

[16] 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. Zimmermann, eds., Akademi-Verlag, Berlin, 1988, pp. 114–125.

[17] ———, *Newton's method based on generalized derivatives for nonsmooth functions: Convergence Analysis*, in Advances in Optimization, W. Oettli and D. Pallaschke, eds., Springer-Verlag, Berlin, 1992, pp. 171–194.

[18] J. M. MARTÍNEZ AND L. QI, *Inexact Newton methods for solving nonsmooth equations*, J. Comput. Appl. Math., to appear.

[19] J. M. MARTÍNEZ AND M. C. ZAMBALDI, *Least change update methods for nonlinear systems with nondifferential terms*, Numer. Funct. Anal. Optim., 14 (1993), pp. 405–415.

[20] J. J. MORÉ, *The Levenberg–Marquardt algorithm: implementation and theory*, in Numerical Analysis, G.A. Watson, ed., Springer-Verlag, Berlin, 1978, pp. 105–116.

[21] ———, *Recent developments in algorithms and software for trust region methods*, in Mathematical Programming: The State of the Art - Bonn 1982, A. Bachem, M. Grötschel, and B. Korte, eds., Springer-Verlag, Berlin, 1983, pp. 258–287.

[22] J. M. ORTEGA AND W. C. RHEINBOLDT, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[23] M. R. OSBORNE, *Nonlinear least squares—The Levenberg algorithm revisited*, J. Australian Math. Soc. B, 19 (1976), pp. 343–357.

[24] J. S. PANG, *Newton's method for* B*-differentiable equations*, Math. Oper. Res., 15 (1990), pp. 311–341.

[25] ———, *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.

[26] ———, *A degree-theoretic approach to parametric nonsmooth equations with multivalued solution sets*, Math. Programming, 62 (1993), pp. 359–383.

[27] J. S. PANG AND S. A. GABRIEL, NE/SQP: *A robust algorithm for the nonlinear complementarity problem*, Math. Programming, 60 (1993), pp. 295–337.

[28] J. S. PANG AND L. QI, *Nonsmooth equations: motivation and algorithms*, SIAM J. Optim., 3 (1993), pp. 443–465.

[29] M. J. D. POWELL, *Convergence properties of a class of minimization algorithms*, in Nonlinear Programming 2, O. L. Mangasarian, R. L. Meyer, and S. M. Robinson, eds., Academic Press, New York, 1975, pp. 1–27.

[30] ———, *On the global convergence of trust region algorithms for unconstrained minimization*, Math. Programming, 29 (1984), pp. 297–303.

[31] M. J. D. POWELL AND Y. YUAN, *A trust region algorithm for equality constrained minimization*, Math. Programming, 49 (1991), pp. 189–211.

[32] L. QI, *Convergence analysis of some algorithms for solving nonsmooth equations*, Math. Oper. Res., 18 (1993), pp. 227–244.

[33] L. QI AND X. CHEN, *A globally convergent successive approximation method for solving severely nonsmooth equations*, SIAM J. Control Optim., to appear.

[34] L. QI AND J. SUN, *A nonsmooth version of Newton's method*, Math. Programming, 58 (1993), pp. 353–368.

[35] D. RALPH, *Global convergence of damped Newton's method for nonsmooth equations via the path search*, Math. Oper. Res., 19 (1994), pp. 352–389.

[36] S. M. ROBINSON, *Newton's method for a class of nonsmooth functions*, Industrial Engineering Working Paper, University of Wisconsin, Madison, 1988.

[37] R. J.-B. WETS, *Stochastic programming: Solution techniques and approximation schemes*, in Mathematical Programming: The State of the Art - Bonn 1982, A. Bachem, M. Grötschel and B. Korte, eds., Springer-Verlag, Berlin, 1983, pp. 566–603.

[38] Y. YUAN, *Conditions for convergence of trust region algorithms for nonsmooth optimization*, Math. Programming, 31 (1985), pp. 220–228.

SIAM J. OPTIMIZATION
Vol. 5, No. 2, pp. 231–235, May 1995

WHY BROYDEN'S NONSYMMETRIC METHOD TERMINATES ON LINEAR EQUATIONS*

DIANNE P. O'LEARY†

Abstract. The family of algorithms introduced by Broyden in 1965 for solving systems of nonlinear equations has been used quite effectively on a variety of problems. In 1979, Gay proved the then surprising result that the algorithms terminate in at most $2n$ steps on linear problems with n variables [*SIAM J. Numer Anal.*, 16 (1979), pp. 623–630]. His very clever proof gives no insight into properties of the intermediate iterates, however. In this work we show that Broyden's methods are projection methods, forcing the residuals to lie in a nested set of subspaces of decreasing dimension. The results apply to linear systems as well as linear least squares problems.

Key words. Broyden's method, nonlinear equations, quasi-Newton methods

AMS subject classifications. 65H10, 65F10

1. Introduction.

In 1965, Broyden introduced a method for solving systems of nonlinear equations $g(x^*) = 0$, where $g : \mathcal{R}^n \to \mathcal{R}^n$ is differentiable [2]. He named it a *quasi-Newton method.* Methods in this class mimic the Newton iteration

$$x_{k+1} = x_k - \{\nabla g(x_k)\}^{-1} g(x_k), \qquad k = 0, 1, \ldots,$$

by substituting an approximation H_k for $\{\nabla g(x_k)\}^{-1}$, the inverse of the Jacobian matrix. This matrix approximation is built up step by step, and the correction to H_k is fashioned so that the *secant condition*

$$H_{k+1} y_k = s_k,$$

is satisfied, where

$$y_k = g(x_{k+1}) - g(x_k) \equiv g_{k+1} - g_k$$

and

$$s_k = x_{k+1} - x_k.$$

This condition is satisfied if $H_{k+1} = \{\nabla g(x_{k+1})\}^{-1}$ and g is linear, and this is the motivation for choosing the correction to H_k so that H_{k+1} satisfies this condition but so that H_{k+1}^{-1} behaves as H_k^{-1} in all directions orthogonal to s_k. This reasoning led Broyden to the formula

$$H_{k+1} = H_k + (s_k - H_k y_k) v_k^T,$$

where v_k is chosen so that $v_k^T y_k = 1$. Specific choices of v_k lead to different algorithms in Broyden's family.

Although certain members of the family proved to be very effective algorithms (specifically, the so called Broyden "good" method that takes v_k in the same direction as s_k), the algorithms were little understood, to the extent that Broyden himself in

* Received by the editors April 2, 1993; accepted for publication (in revised form) November 2, 1993.

† Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, Maryland 20742 (oleary@cs.umd.edu). This work was supported by National Science Foundation grant CCR 9115568.

1972 said, "The algorithm does not enjoy the property of quadratic termination even when used for function minimization with exact line searches." [3, p. 97].

This misconception, common to the entire optimization community, was disproved in 1979 by David Gay in a surprising and clever construction [4]. He showed that if

$$g(x) = Ax - b,$$

where $A \in \mathcal{R}^{n \times n}$ is nonsingular, then the following result holds.

LEMMA 1.1. (Gay) *If g_k and y_{k-1} are linearly independent, then for $1 \leq j \leq \lfloor (k+1)/2 \rfloor$, the vectors $\{(AH_{k-2j+1})^i g_{k-2j+1}\}$ are linearly independent for $0 \leq i \leq j$.*

This result implies that for some $k \leq 2n$, g_{k-1} and y_{k-2} must be linearly dependent, and Gay showed that in this case $g_k = 0$ and termination occurs.

Gay's construction leads to a proof of a $2n$-step Q-quadratic rate of convergence for Broyden's good method. It yields little insight, though, into how the intermediate iterates in Broyden's method are behaving in the case of linear systems, and the purpose of this work is to develop that understanding.

2. The character of the Broyden iterates. First we establish some useful relations. The change in the x vector is given by

$$s_k = -H_k g_k,$$

and the change in the residual g is

$$y_k = As_k = -AH_k g_k,$$

so we can express the new residual as

$$g_{k+1} = g_k + y_k = (I - AH_k)g_k.$$

For convenience, we denote the matrix in this expression as

$$F_k = I - AH_k,$$

and denote the product of such factors as

$$P_k = F_k F_{k-1} \ldots F_0 = F_k P_{k-1}.$$

Then a simple induction-style argument gives us a useful formula for g_{k+1} and thus for y_k and s_k.

LEMMA 2.1. *It holds that*

$$\begin{aligned} g_{k+1} &= P_k g_0, \\ y_k &= -AH_k P_{k-1} g_0, \\ s_k &= -H_k P_{k-1} g_0. \end{aligned}$$

Thus, the character of the product matrices P_k determines the behavior of the residuals g_k in the course of the iteration. The key to this behavior is the nature of the left null vectors of P_k, the vectors z for which $z^T P_k = 0$. We prove in the next section that these vectors have a very special form. In particular, the factor matrices F_k

are quite *defective*, having a zero eigenvalue with a Jordan block of size $\lceil k/2 \rceil$. The linearly independent left *principal vectors* of F_k, vectors z_i satisfying

$$\begin{aligned} z_1^T F_k &= 0, \\ z_3^T F_k &= z_1^T, \\ z_i^T F_k &= z_{i-2}^T, \end{aligned}$$

do not depend on k, and in fact are left *eigenvectors* of P_k corresponding to a zero eigenvalue. Thus the vector g_{k+1} is orthogonal to the expanding subspace spanned by the vectors $\{z_i\}$ (i odd and $i \leq k$) and, after at most $2n$ steps, is forced to be zero.

3. The behavior of the Broyden iterates. David Gay proved an important fact about the rank of the factor matrices F_k, and the following result is essentially his.

LEMMA 3.1. (Gay) *For $k \geq 1$, if $y_k \neq 0$, $v_k^T y_{k-1} \neq 0$, and* $\operatorname{rank}(F_k) = n - 1$, *then* $\operatorname{rank}(F_{k+1}) = n - 1$, *and y_k spans the null space of F_{k+1}.*

Proof. For $k \geq 0$,

$$\begin{aligned} F_{k+1} &= I - AH_{k+1} \\ &= I - AH_k - (As_k - AH_k y_k)v_k^T \\ &= I - AH_k - (y_k - AH_k y_k)v_k^T \\ &= (I - AH_k)(I - y_k v_k^T) \\ &= F_k(I - y_k v_k^T). \end{aligned}$$

Since $v_k^T y_k = 1$, we see that $F_{k+1} y_k = 0$. Similarly, y_{k-1} spans the null space of F_k ($k \geq 1$). Any other right null vector y of F_{k+1} must (after scaling) satisfy $(I - y_k v_k^T)y = y_{k-1}$. But v_k^T spans the null space of $(I - y_k v_k^T)^T$ and because, by assumption, $v_k^T y_{k-1} \neq 0$, y_{k-1} is not in the range of $(I - y_k v_k^T)$, and thus y_k spans the null space of F_{k+1}. □

This lemma leads to the important observation that the sequence of matrices H_k does not terminate with the inverse of the matrix A, at least in the *usual* case in which all $v_k^T y_{k-1} \neq 0$. In fact, each matrix H_k and A^{-1} agree only on a subspace of dimension 1.

We make several assumptions on the iteration.

1. $v_j^T y_{j-1} \neq 0$, $j = 1, 2, \ldots, k$, and $y_k \neq 0$.
2. v_0 is in the range of F_0^T.
3. $k \leq 2n - 1$ is odd. (This is for notational convenience.)

We can now show that the matrix F_k is defective and exhibit the left null vectors of P_k (which are the same as the left null vectors of P_{k+1}).

LEMMA 3.2. *Define the sequence of vectors*

$$z_1^T F_1 = 0, \quad z_i^T F_i = z_{i-2}^T, \quad i = 3, 5, \ldots, k.$$

These vectors exist and are linearly independent, and z_i^T is a left null vector of P_j for $j = i, i+1, \ldots, k$. Furthermore, $z_i^T g_j = 0$, $j = i+1, i+2, \ldots, k$.

Proof. Define z_1 by $z_1^T F_0 = v_0^T$. This nonzero vector exists by assumption 2, and since

$$F_j = F_0(1 - y_0 v_0^T) \ldots (1 - y_{j-1} v_{j-1}^T),$$

it is easy to see that, for $j = 1, 2, \ldots, k$, $z_1^T F_j = 0$, and therefore $z_1^T P_j = 0$ as well. In order to continue the construction, we need an orthogonality result: for $j = 2, 3, \ldots, k$,

$$z_1^T y_j = z_1^T g_{j+1} - z_1^T g_j = z_1^T P_j g_0 - z_1^T P_{j-1} g_0 = 0.$$

For the "induction step," assume that linearly independent vectors $z_1, \ldots, z_{i-2}$ have been constructed for $i \leq k$, that each of these vectors satisfies $z_m^T F_j = z_{m-2}^T$ ($z_1^T F_j = 0$) and $z_m^T P_j = 0$ for $j = m, \ldots, k$, and that $z_{i-2}^T y_j = 0$ for $j = i-1, \ldots, k$. (Linear independence follows from the fact that they are principal vectors for F_{i-2}.) Let $z_i^T F_i = z_{i-2}^T$. Note that z_i exists since $z_{i-2}^T y_{i-1} = 0$, and y_{i-1} spans the null space of F_i. We have that, for $j = i, \ldots, k$, $z_i^T F_j = z_{i-2}^T$, and therefore $z_i^T P_j = z_{i-2}^T P_{j-1} = 0$ as well. Then for $j = i+1, \ldots, k$, we have

$$z_i^T y_j = z_i^T g_{j+1} - z_i^T g_j = z_i^T P_j g_0 - z_i^T P_{j-1} g_0 = 0. \qquad \square$$

In the course of this proof, we have established the following result.

THEOREM 3.3. *Under the above three assumptions, and if A is nonsingular, then after $k+1$ steps of Broyden's method, the residual $g(x_{k+1})$ is orthogonal to the linearly independent vectors $z_1, z_3, \ldots, z_k$, and thus the algorithm must terminate with the exact solution vector after at most $2n$ iterations.*

4. Overdetermined or rank deficient linear systems. Gerber and Luk [5] gave a nice generalization of Gay's results to overdetermined or rank deficient linear systems, and the results in this paper can be generalized this way as well. The matrices F_k have dimensions $m \times m$, where $m \geq n$ is the number of equations. The two lemmas and their proofs are unchanged, but the theorem has a slightly different statement. Let $\mathbf{R}$ denote the range of a matrix, and $\mathbf{N}$ denote the null space.

THEOREM 4.1. *Under the above three assumptions, and if $A \in \mathcal{R}^{m \times n}$ has* rank *p, and if $\mathbf{N}(H_k) = \mathbf{N}(A^T)$, then after $k+1$ steps of Broyden's method, the residual $g(x_{k+1})$ is orthogonal to the linearly independent vectors $z_1, z_3, \ldots, z_k$, which are contained in the range of A, and thus the algorithm must terminate with the least squares solution vector after at most $2p$ iterations.*

Proof. The fact that z_i is in the range of A is established by induction. $\square$

Gerber and Luk give sufficient conditions guaranteeing that $\mathbf{N}(H_k) = \mathbf{N}(A^T)$.

1. $\mathbf{R}(H_0) = \mathbf{R}(A^T)$,
2. $\mathbf{N}(H_0) = \mathbf{N}(A^T)$,

3a. $v_k = H_k^T u_k$ for some vector u_k, with $u_k^T s_k \neq 0$.

Condition 3a is satisfied by Broyden's "good" method (v_k parallel to s_k) but not the "bad" method (v_k parallel to y_k). It is easy to show by induction that the condition $\mathbf{N}(H_k) = \mathbf{N}(A^T)$ also holds under the following assumption, valid for the "bad" method:

3b. $v_k = A u_k$ for some vector u_k, with $u_k^T s_k \neq 0$.

First, assume that $z \in \mathbf{N}(H_k)$ and that $\mathbf{N}(H_k) = \mathbf{N}(A^T)$. Then

$$H_{k+1} z = H_k z + (s_k - H_k y_k) v_k^T z,$$

and this is zero if and only if $v_k^T z = 0$, which is assured if either $v_k^T = u_k^T H_k$ or $v_k^T = u_k^T A^T$. Thus, $\mathbf{N}(H_k) \subseteq \mathbf{N}(H_{k+1})$. Now, by writing v_k as $H_k^T u_k$, valid under either 3a or 3b, we can show [5] that $H_{k+1} = (I - H_k g_{k+1} u_k^T) H_k$, and that the determinant of the first factor is $u_k^T s_k$, nonzero by condition 3. Thus $\mathbf{N}(H_{k+1}) = \mathbf{N}(H_k)$.

5. Concluding notes. Since the vectors v_k are arbitrary except for a normalization, it is easy to ensure that assumptions 1 and 2 of §3 are are satisfied. But if not, termination actually occurs earlier: if F_k has rank $n-1$ and $v_k^T y_{k-1} = 0$, then F_{k+1} has rank $n-2$ with right null vectors y_k and y_{k-1}. Similarly, P_{k+1} has one extra zero eigenvalue.

The conclusions in this work apply to the Broyden family of methods, whether they are implemented by updating an approximation to the Hessian, its inverse, or its factors. The inverse approximation was only used for notational convenience; other implementations are preferred in numerical computation.

The conclusions depend critically on the assumption that the step length parameter is 1 (i.e., if $s_k = -\alpha H_k g_k$, then $\alpha = 1$).

As noted by a referee, the conclusions depend on only two properties of Broyden's method, consequences of Lemma 2.1:

$$F_{k+1} = F_k(I - (F_k - I)g_k v_k^T),$$
$$g_{k+1} = F_k g_k,$$

where g_0 and F_0 are given, and the v_k are (almost) arbitrary. Thus the results can be applied to a broader class of methods, in the same spirit as the work by Boggs and Tolle [1].

Note added in proof. Other results on the convergence of these methods are given by Hwang and Kelley [6], who cite a termination proof by W. Burmeister in 1975.

Acknowledgment. Thanks to Gene H. Golub for providing the Gerber and Luk reference, and to David Gay and C. G. Broyden for careful reading of a draft of the manuscript.

REFERENCES

[1] P. T. Boggs and J. W. Tolle, *Convergence properties of a class of rank-two updates*, SIAM J. Optim., 4 (1994), pp. 262–287.

[2] C. G. Broyden, *A class of methods for solving nonlinear simultaneous equations*, Math. Comp., 19 (1965), pp. 577–593.

[3] ———, *Quasi-Newton methods*, in Numerical Methods for Unconstrained Optimization, W. Murray, ed., Academic Press, New York, 1972, pp. 87–106.

[4] D. M. Gay, *Some convergence properties of Broyden's method*, SIAM J. Numer. Anal., 16 (1979), pp. 623–630.

[5] R. R. Gerber and F. T. Luk, *A generalized Broyden's method for solving simultaneous linear equations*, SIAM J. Numer. Anal., 18 (1981), pp. 882–890.

[6] D. M. Hwang and C. T. Kelley, *Convergence of Broyden's method in Banach spaces*, SIAM J. Optim., 2 (1992), pp. 505–532.

SIAM J. OPTIMIZATION
Vol. 5, No. 2, pp. 236–246, May 1995

002

A NEW INFINITY-NORM PATH FOLLOWING ALGORITHM FOR LINEAR PROGRAMMING*

KURT M. ANSTREICHER† AND ROBERT A. BOSCH‡

Abstract. We devise a new primal-dual path following algorithm for linear programming that is based entirely on an infinity-norm centering measure. The algorithm makes reductions in a path parameter μ, each of which is followed by a sequence of centering steps. The algorithm has similarities with both long step path following and predictor-corrector methods. We also consider a "modified" version of the algorithm that uses partial updating of the projection equations. The analysis of the modified algorithm has some interesting differences compared with previously devised partial updating methods. In particular, partial updating obtains a factor-of-$\sqrt{n}$ complexity reduction even though the permissible relative error in the approximate scaling factors is extremely small — only $O(1/\sqrt{n})$.

Key words. interior point method, primal-dual algorithm, partial updating, rank-one updates, modified method

AMS subject classification. 90C05

1. Introduction.

The motivation for a primal-dual path following algorithm for linear programming (LP) was provided by Megiddo (1989), and the idea was first operationalized by Kojima, Mizuno, and Yoshise (1989a), and Monteiro and Adler (1989a). Primal-dual path following algorithms maintain primal and dual iterates which are near the central path, a distinguished set of solutions. A key element in the design of such an algorithm is the precise manner in which "near" in the previous sentence is characterized. Kojima, Mizuno, and Yoshise (1989a), using a "one-sided infinity-norm" measure, obtained a complexity of $O(nL)$ steps, while Monteiro and Adler (1989a) obtained $O(\sqrt{n}L)$ step complexity using a stricter 2-norm measure. Here n is the number of variables in a standard form linear program, with integer data having total bit size L.

Although looser proximity measures, such as those employed by Kojima, Mizuno, and Yoshise (1989a), invariably result in inferior worst-case complexity bounds, in practice the resulting algorithms work much better than methods that use stricter 2-norm measures. Some theoretical justification for this phenomenon is provided by Mizuno, Todd, and Ye (1993), where a number of different primal-dual path following algorithms are compared using both worst-case and anticipated behavior criteria. Although the analysis of anticipated behavior does not provide a rigorous probabilistic result, it does indicate that infinity-norm based algorithms have a good chance of converging faster than methods based on 2-norm measures.

One particular method described in Mizuno, Todd, and Ye (1993), the predictor–corrector (PC) algorithm, provides considerable motivation for the algorithm we develop in this paper. The PC algorithm works with a 2-norm centering measure, which is used to define neighborhoods of the central trajectory. Beginning with a primal-dual pair in a "small" neighborhood, the PC algorithm first computes the primal and

*Received by the editors February 23, 1993; accepted for publication (in revised form) February 1, 1994.

†Department of Management Science, University of Iowa, Iowa City, Iowa 52242 (kanstrei@scout-po.biz.uiowa.edu).

‡Department of Mathematics, Oberlin College, Oberlin, Ohio 44074 (bobb@occs.cs.oberlin.edu).

dual directions designed to reduce the primal-dual gap as quickly as possible. A step is then taken, using the longest steplength, that keeps the new points in a larger neighborhood of the central path. This is the predictor step. Each predictor step is followed by a corrector step, which holds the gap constant, but returns the iterates to the small neighborhood so that the process can be repeated. The resulting algorithm attains $O(\sqrt{n}L)$ step complexity, as is typical for 2-norm based methods.

In this paper we are particularly interested in how corrector steps, as used by the PC algorithm, work when the 2-norm centering measure is replaced by an infinity-norm measure. Indeed, Mizuno, Todd, and Ye (1993) note, at the end of their §3, that "... corrector steps may not behave well with the l_∞ norm." Basically the algorithm we develop replaces a single corrector step with a sequence of steps which ultimately return the iterates to the "small" neighborhood. However, we do not begin with a predictor step, but instead consider the path parameter μ to be exogenous, and simply reduce μ as much as possible, keeping the current primal-dual pair in a "large" neighborhood. As a result, the overall structure of our algorithm is similar to the "long step path following" methods of Gonzaga (1991) and Roos and Vial (1990), except that (i) we use primal-dual steps; (ii) the reduction in the path parameter can be "large," i.e., $\Omega(1)$, but not arbitrarily large; and (iii) the bound on the number of steps required to return to a small neighborhood of the central path is not based on reduction in a potential or barrier function, but rather reduction in the centering measure itself. See also Jansen et al. (1994) for a primal-dual long step path following algorithm, and Ling (1993) for a polynomial-time affine scaling method based on a novel infinity-norm centering measure.

In addition to the basic algorithm, we develop a "modified" version that uses partial updating of the projection equations that are solved on each corrector step. Primal-dual path following methods using partial updating are analyzed by Kojima, Mizuno, and Yoshise (1989b), and Monteiro and Adler (1989b), and by now the use of partial updating to reduce the complexity of interior point algorithms for LP is quite standard. However, to our knowledge, no one has previously examined the behavior of partial updating when infinity-norm neighborhoods are used, and in doing so we find some surprising differences. In particular, we show that for our modified algorithm, partial updating provides a factor-of-$\sqrt{n}$ complexity reduction even though the allowable relative error in the scaling factors is extremely small, only $O(1/\sqrt{n})$, compared to $\Omega(1)$ in all previous applications.

2. The basic algorithm. Consider the primal-dual pair of linear programs

$$\text{(P)} \qquad \min\{c^T x \mid Ax = b,\ x \geq 0\},$$

$$\text{(D)} \qquad \max\{b^T y \mid A^T y + s = c,\ s \geq 0\},$$

where A is an $m \times n$ matrix with linearly independent rows. Throughout the paper we make extensive use of a well-known "centering measure" for (P) and (D). In particular, if $0 < \beta < 1$, x and (y, s) are feasible for (P) and (D), respectively, and μ is a positive scalar for which

$$\| \frac{Xs}{\mu} - e \|_\infty \leq \beta, \tag{2.1}$$

then we say that x and s are *β-centered with respect to μ*. In (2.1), X denotes the diagonal matrix $X = \text{diag}(x)$; similar notation is used for other diagonal matrices throughout the paper. Throughout the paper e denotes the vector with each component equal to one. In the context of primal-dual algorithms, for example, the methods

devised in Mizuno, Todd, and Ye (1993), the centering measure (2.1) is often employed with $\mu = x^T s/n$. Throughout this paper, however, μ is *always* considered to be an independent parameter.

The algorithm which we develop in this section is comprised of inner and outer steps. Outer steps correspond to simply reducing the parameter μ, and each outer step is followed by a sequence of inner steps. Suppose that x and s are β-centered with respect to $\mu > 0$. An inner step finds feasible solutions more finely centered with respect to μ than are x and s. That is, an inner step finds feasible solutions that are β'-centered with respect to μ, where $\beta' < \beta$. This is accomplished by making a step of the form

$$x(\theta) = x + \theta d_x, \quad y(\theta) = y + \theta d_y, \quad s(\theta) = s + \theta d_s,$$

where the directions d_x, d_y, d_s are obtained by solving the system

$$\begin{aligned} S d_x + X d_s &= \mu e - Xs, \\ A d_x &= 0, \\ A^T d_y + d_s &= 0. \end{aligned} \tag{2.2}$$

Lemma 2.1, Theorem 2.2, and Corollary 2.3 below present the relevant results concerning inner steps. The main result is Corollary 2.3. Note that from (2.2) it follows immediately that

$$\begin{aligned} X(\theta)\, s(\theta) - \mu e &= Xs - \mu e + \theta(S d_x + X d_s) + \theta^2 D_x d_s \\ &= (1-\theta)(Xs - \mu e) + \theta^2 D_x d_s, \end{aligned} \tag{2.3}$$

so that when x and s are β-centered with respect to μ, and $0 \le \theta \le 1$, we easily obtain

$$\| \frac{X(\theta)\, s(\theta)}{\mu} - e \|_\infty \le (1-\theta)\beta + \frac{\theta^2}{\mu} \| D_x d_s \|_\infty. \tag{2.4}$$

LEMMA 2.1. *Suppose that x and s are β-centered with respect to μ, and that d_x, d_s are obtained by solving* (2.2). *Then*

$$\| D_x d_s \|_\infty \le \frac{n\beta^2\mu}{4(1-\beta)}.$$

Proof. As in Mizuno, Todd, and Ye (1993), we express $D_x d_s$ differently. It is easy to show that

$$\begin{aligned} d_x &= X^{.5} S^{-.5} p, \\ d_s &= X^{-.5} S^{.5} q, \end{aligned} \qquad \text{where} \qquad \begin{aligned} p &= P_{\mathcal{N}(\bar{A})} r, \\ q &= P_{\mathcal{R}(\bar{A}^T)} r, \\ r &= X^{-.5} S^{-.5} (\mu e - Xs), \end{aligned}$$

$\bar{A} = A X^{.5} S^{-.5}$, and $P_{\mathcal{N}(\cdot)}$ and $P_{\mathcal{R}(\cdot)}$ denote projection onto the nullspace and range, respectively. Note that $D_x d_s = Pq$. Since $p + q = r$ and $p^T q = 0$, Lemma 1(c) of Mizuno, Todd, and Ye (1993) applies here, yielding

$$\| Pq \|_\infty \le \frac{\| r \|_2^2}{4} \le \frac{n}{4} \| X^{-.5} S^{-.5} e \|_\infty^2 \| \mu e - Xs \|_\infty^2,$$

which gives

$$\| Pq \|_\infty \leq \frac{n}{4} \max_i \left\{ \frac{1}{x_i s_i} \right\} \beta^2 \mu^2.$$

But $\| Xs - \mu e \|_\infty \leq \beta\mu$ also implies that $x_i s_i \geq (1-\beta)\mu$ for each i. The lemma follows immediately. □

THEOREM 2.2. *Suppose that x and s are β-centered with respect to μ, and that d_x, d_s are obtained by solving (2.2). Let $\hat{\theta} = 2(1-\beta)/(n\beta)$, and assume that $\hat{\theta} \leq 1$. Then $(x(\hat{\theta}), s(\hat{\theta})) > 0$, and*

$$\| \frac{X(\hat{\theta})\, s(\hat{\theta})}{\mu} - e \|_\infty \leq \beta - \frac{1-\beta}{n}.$$

Proof. From (2.4) and Lemma 2.1, we have

$$\| \frac{X(\theta)\, s(\theta)}{\mu} - e \|_\infty \leq (1-\theta)\beta + \theta^2 \frac{n\beta^2}{4(1-\beta)}, \tag{2.5}$$

for $0 \leq \theta \leq 1$. The right-hand side of (2.5) is a convex quadratic function of θ, which is minimized at $\theta = \hat{\theta}$. Then $(x(\hat{\theta}), s(\hat{\theta})) > 0$ follows from the fact that the right-hand side of (2.5) is decreasing for $0 \leq \theta \leq \hat{\theta}$, and substituting $\hat{\theta}$ into (2.5) completes the proof. □

COROLLARY 2.3. *Let $0 < \underline{\beta} < \overline{\beta} < 1$, where $\underline{\beta}$ and $(1-\overline{\beta})$ are $\Omega(1)$, and $\underline{\beta} \geq 2/(n+2)$. Let $\mu > 0$. If x and s are $\overline{\beta}$-centered with respect to μ, then at most $O(n)$ inner steps are required to obtain solutions that are $\underline{\beta}$-centered with respect to μ.*

Proof. Note that $\hat{\theta} \leq 1$, as required in Theorem 2.2, holds for $\underline{\beta} \leq \beta \leq \overline{\beta}$ so long as $\underline{\beta} \geq 2/(n+2)$. The corollary then follows immediately from repeated application of Theorem 2.2. □

We now describe the entire basic algorithm. We are given $0 < \underline{\beta} < \overline{\beta} < 1$, and $\mu > 0$, as well as solutions to (P) and (D) that are $\overline{\beta}$-centered with respect to μ. Inner steps are then performed until we obtain solutions that are $\underline{\beta}$-centered with respect to μ. At this point, we perform an *outer step*; that is, we reduce μ to a new value μ' chosen so that the current solutions to (P) and (D) are $\overline{\beta}$-centered with respect to μ'. We then set $\mu := \mu'$ and repeat the entire process, terminating when μ is sufficiently small. In the case where all the data in (P) is integral, with total bit size L, it suffices to have $\mu \leq 2^{-O(L)}$ to invoke a "rounding" procedure, which obtains optimal basic solutions to (P) and (D).

The following lemma demonstrates that "large," i.e., $\Omega(1)$, reductions of μ are permissible on outer steps. In particular, we may set $\mu' := \eta\mu$, where $0 < \eta < 1$ is independent of n. As a result, at most $O(L)$ outer steps are required, so long as initial solutions which are $\overline{\beta}$-centered with respect to a value of $\mu = O(L)$ are available. This in turn implies that the algorithm requires a total of at most $O(nL)$ inner steps. Each inner step requires $O(n^3)$ arithmetic operations; $O(n^3)$ for factorizing $AXS^{-1}A^T$, and $O(n^2)$ for everything else. As a result, the basic algorithm's overall worst-case complexity is $O(n^4 L)$ operations.

LEMMA 2.4. *Suppose that x and s are $\underline{\beta}$-centered with respect to μ, and $\mu' = \eta\mu$, where $0 < \eta < 1$. Let $\overline{\beta} = (1+\underline{\beta})/\eta - 1$. Then x and s are $\overline{\beta}$-centered with respect to μ'.*

Proof. Since $\| e \|_\infty = 1$, we have

$$\begin{aligned} \| Xs - \mu' e \|_\infty &\leq \| Xs - \mu e \|_\infty + \| \mu e - \mu' e \|_\infty \\ &\leq \underline{\beta}\mu + (\mu - \mu') \\ &= (1 + \underline{\beta})\mu - \mu'. \end{aligned}$$

Dividing by μ' completes the proof. □

Note that if, for example, $\eta = 3/4$ and $\underline{\beta} = 1/4$, then $\overline{\beta} = 2/3$. Consequently, solutions that are $(1/4)$-centered with respect to μ will be $(2/3)$-centered with respect to $(3/4)\mu$. Also note that $\underline{\beta} = 1/4$ satisfies $\underline{\beta} \geq 2/(n+2)$, as required in Corollary 2.3, for all $n \geq 6$.

As noted above, Corollary 2.3 and Lemma 2.4 together demonstrate that the basic algorithm of this section, suitably initialized, has a complexity of $O(nL)$ steps. Although the steplength $\hat{\theta}$ is used in Theorem 2.2 to establish a guaranteed descent in the centering measure $\| X(\theta)s(\theta)/\mu - e \|_\infty$, in practice on each inner step one may perform a linesearch of this quantity in $\theta \geq 0$ to "re–center" the iterates as rapidly as possible. Using such a linesearch, one would expect that far fewer than $O(n)$ inner steps would be required on each outer step, greatly improving the practical performance of the algorithm.

As mentioned in the Introduction, the algorithm of this section was partially motivated by the PC algorithm of Mizuno, Todd, and Ye (1993). In the PC algorithm, the parameter μ is endogenously defined by $\mu = x^T s/n$, and the "outer" step is replaced by a predictor step. The direction for the predictor step is obtained by solving (2.2), with the equations $Sd_x + Xd_s = \mu e - Xs$ replaced by $Sd_x + Xd_s = -Xs$. A step $x(\bar{\theta}), s(\bar{\theta})$ is then taken using the resulting directions, where $\bar{\theta}$ is chosen to be the maximum value such that (in our notation) $x(\theta)$ and $s(\theta)$ are $\overline{\beta}$-centered with respect to $\mu(\theta) = x(\theta)^T s(\theta)/n = (1-\theta)\mu$, for all $0 \leq \theta \leq \bar{\theta}$. Mizuno, Todd, and Ye (1993) uses a 2-norm measure analogous to (2.1), but it is easy to devise a similar infinity-norm based algorithm. With $\mu = x^T s/n$, the analysis of corrector steps based on an infinity-norm measure is essentially identical to Theorem 2.2. For predictor steps, the key issue is the steplength $\bar{\theta}$, which determines the gap reduction. Unfortunately, following the analysis in Mizuno, Todd, and Ye (1993), using infinity-norm measures, results in $\bar{\theta} = O(1/\sqrt{n})$, the same value as obtained in the 2-norm case. The result is a method that requires $O(\sqrt{n}L)$ predictor steps, each followed by $O(n)$ corrector steps, for a total complexity of $O(n^{4.5}L)$ operations.

Finally, it is worthwhile to note that Theorem 2.2 and Lemma 2.4 can also be combined to give a simple "short step path following" algorithm for linear programming. In particular, consider the basic algorithm of this section, but where following a reduction in μ, only a *single* inner step is taken. From Theorem 2.2 and Lemma 2.4, this single step will produce a primal-dual pair which is $\underline{\beta}$-centered with respect to $\mu' = \eta\mu$ if

$$\begin{aligned} \overline{\beta} - \frac{1-\overline{\beta}}{n} &\leq \underline{\beta}, \\ \overline{\beta} - \frac{1-\overline{\beta}}{n} &\leq \eta(1+\overline{\beta}) - 1, \\ \eta &\geq 1 - \frac{1-\overline{\beta}}{n(1+\overline{\beta})}. \end{aligned}$$

Then $\mu' = (1 - \Omega(1/n))\mu$ immediately leads to an algorithm with $O(nL)$ step complexity.

3. The modified algorithm. In this section we describe a partial updating version of the basic algorithm. Here we maintain approximations $\tilde{x}$ and $\tilde{s}$ to x and s, controlling the quality of the approximations via a parameter $0 < \gamma < 1$. While running the algorithm, we ensure that

$$1-\gamma \le \frac{x_i}{\tilde{x}_i} \le 1+\gamma \quad \text{and} \quad 1-\gamma \le \frac{s_i}{\tilde{s}_i} \le 1+\gamma \quad \text{for } i = 1,\dots,n, \tag{3.1}$$

updating a component of $\tilde{x}$ or $\tilde{s}$ only when it fails to satisfy (3.1).

The modified algorithm's inner steps are very similar to those of the basic algorithm. The difference is that the directions d_x, d_y, d_s are obtained by solving the following altered version of (2.2):

$$\begin{aligned} \tilde{S}d_x + \tilde{X}d_s &= \mu e - Xs, \\ Ad_x &= 0, \\ A^T d_y + d_s &= 0. \end{aligned} \tag{3.2}$$

It is easy to show that the solutions d_x and d_s of (3.2) are given by

$$\begin{aligned} d_x &= \tilde{X}^{.5}\tilde{S}^{-.5}p, \\ d_s &= \tilde{X}^{-.5}\tilde{S}^{.5}q, \end{aligned} \qquad \text{where} \qquad \begin{aligned} p &= P_{\mathcal{N}(\tilde{A})}r, \\ q &= P_{\mathcal{R}(\tilde{A}^T)}r, \\ r &= \tilde{X}^{-.5}\tilde{S}^{-.5}(\mu e - Xs), \end{aligned} \tag{3.3}$$

and $\tilde{A} = A\tilde{X}^{.5}\tilde{S}^{-.5}$. Note that (2.3) continues to hold using d_x, d_s from (3.3). However, we do *not* have $Sd_x + Xd_s = \mu e - Xs$ as before; instead (3.2) gives $\tilde{S}d_x + \tilde{X}d_s = \mu e - Xs$. Accordingly, we use $Sd_x + Xd_s = (\tilde{S}d_x + \tilde{X}d_s) + (S-\tilde{S})d_x + (X-\tilde{X})d_s$ to write (2.3) as

$$X(\theta)\, s(\theta) - \mu e = (1-\theta)(Xs - \mu e) + \theta(S-\tilde{S})d_x + \theta(X-\tilde{X})d_s + \theta^2 D_x d_s.$$

It then follows immediately that for $0 \le \theta \le 1$,

$$\begin{aligned} \| X(\theta)\, s(\theta) - \mu e \|_\infty \le (1-\theta)\beta\mu + \theta(\| (S-\tilde{S})d_x \|_\infty + \| (X-\tilde{X})d_s \|_\infty) \\ + \theta^2 \| D_x d_s \|_\infty. \end{aligned} \tag{3.4}$$

LEMMA 3.1. *Suppose that x and s are β-centered with respect to μ, and that $\tilde{x}$ and $\tilde{s}$ satisfy* (3.1). *Then*

$$\frac{(1-\beta)\mu}{(1+\gamma)^2} \le \tilde{x}_i\tilde{s}_i \le \frac{(1+\beta)\mu}{(1-\gamma)^2} \qquad \text{for } i = 1,\dots,n.$$

Proof. Since x and s are β-centered with respect to μ, we have

$$(1-\beta)\mu \le x_i s_i \le (1+\beta)\mu \qquad \text{for } i = 1,\dots,n. \tag{3.5}$$

Since $\tilde{x}$ and $\tilde{s}$ satisfy (3.1), we have

$$\frac{x_i s_i}{(1+\gamma)^2} \le \tilde{x}_i\tilde{s}_i \le \frac{x_i s_i}{(1-\gamma)^2} \qquad \text{for } i = 1,\dots,n. \tag{3.6}$$

The lemma follows by combining (3.5) and (3.6). □

LEMMA 3.2. *Suppose that x and s are β-centered with respect to μ, that $\tilde{x}$ and $\tilde{s}$ satisfy* (3.1), *and that d_x, d_s are obtained by solving* (3.2). *Then* $\| (S - \tilde{S})d_x \|_\infty \leq \sqrt{n}\,\lambda\beta\mu$, *and* $\| (X - \tilde{X})d_s \|_\infty \leq \sqrt{n}\,\lambda\beta\mu$, *where*

$$\lambda = \gamma \frac{1+\gamma}{1-\gamma}\sqrt{\frac{1+\beta}{1-\beta}}.$$

Proof. By (3.3), (3.1), and Lemma 3.1 we obtain

$$\begin{aligned} \| (S - \tilde{S})d_x \|_\infty &= \| (S - \tilde{S})\tilde{S}^{-1}(\tilde{X}^{.5}\tilde{S}^{.5}p) \|_\infty \\ &\leq \| \tilde{S}^{-1}(s - \tilde{s}) \|_\infty \| \tilde{X}^{.5}\tilde{S}^{.5}e \|_\infty \| p \|_\infty \\ &\leq \gamma \frac{\sqrt{(1+\beta)\mu}}{1-\gamma} \| p \|_\infty. \end{aligned} \tag{3.7}$$

Since $\| p \|_\infty \leq \| p \|_2 \leq \| r \|_2 \leq \sqrt{n} \| r \|_\infty$, we have

$$\begin{aligned} \| p \|_\infty &\leq \sqrt{n} \| \tilde{X}^{-.5}\tilde{S}^{-.5}(\mu e - Xs) \|_\infty \\ &\leq \sqrt{n} \| \tilde{X}^{-.5}\tilde{S}^{-.5}e \|_\infty \| \mu e - Xs \|_\infty \\ &\leq \sqrt{n} \frac{1+\gamma}{\sqrt{(1-\beta)\mu}} \beta\mu, \end{aligned} \tag{3.8}$$

where the last inequality follows from Lemma 3.1 and the fact that x and s are β-centered with respect to μ. By combining (3.7) and (3.8), we obtain the first inequality of the lemma. The proof of the second inequality is very similar. □

LEMMA 3.3. *Suppose that x and s are β-centered with respect to μ, that $\tilde{x}$ and $\tilde{s}$ satisfy* (3.1), *and that d_x, d_s are obtained by solving* (3.2). *Then*

$$\| D_x d_s \|_\infty \leq \frac{n\beta^2\mu(1+\gamma)^2}{4(1-\beta)}.$$

Proof. First note that $D_x d_s = Pq$. Since $p + q = r$ and $p^T q = 0$, Lemma 1(c) of Mizuno, Todd, and Ye (1993) again applies, yielding

$$\| Pq \|_\infty \leq \frac{\| r \|_2^2}{4} \leq \frac{n}{4} \| \tilde{X}^{-.5}\tilde{S}^{-.5}e \|_\infty^2 \| \mu e - Xs \|_\infty^2.$$

From this, Lemma 3.1, and the fact that x and s are β-centered with respect to μ, we obtain

$$\| Pq \|_\infty \leq \frac{n}{4} \frac{(1+\gamma)^2}{(1-\beta)\mu} \beta^2\mu^2. \qquad \square$$

THEOREM 3.4. *Suppose that x and s are β-centered with respect to μ, that $\tilde{x}$ and $\tilde{s}$ satisfy* (3.1), *and that d_x, d_s are obtained by solving* (3.2). *Let*

$$\hat{\theta} = \frac{2(1-\beta)\tau}{n\beta(1+\gamma)}, \qquad \text{where} \quad \tau = \frac{1 - 2\sqrt{n}\,\lambda}{1+\gamma},$$

and λ is as in Lemma 3.2. *Assume that γ and β are such that $\tau > 0$ and $\hat{\theta} \leq 1$. Then $(x(\hat{\theta}), s(\hat{\theta})) > 0$, and*

$$\| \frac{X(\hat{\theta})\, s(\hat{\theta})}{\mu} - e \|_\infty \leq \beta - \frac{(1-\beta)\tau^2}{n}.$$

Proof. Combining (3.4) with Lemmas 3.2 and 3.3, and dividing by μ, yields

$$\| \frac{X(\theta)\, s(\theta)}{\mu} - e \|_\infty \leq (1-\theta)\beta + 2\theta\sqrt{n}\,\lambda\beta + \theta^2 \frac{n\beta^2(1+\gamma)^2}{4(1-\beta)}$$

$$= \beta - \theta\beta(1+\gamma)\tau + \theta^2 \frac{n\beta^2(1+\gamma)^2}{4(1-\beta)}, \tag{3.9}$$

for $0 \leq \theta \leq 1$. The right-hand side of (3.9) is a convex quadratic function of θ, is decreasing in $\theta \geq 0$ so long as $\tau > 0$, and is then minimized at $\theta = \hat{\theta}$. Then $(x(\hat{\theta}), s(\hat{\theta})) > 0$ follows from the fact that the right-hand side of (3.9) is decreasing in θ for $0 \leq \theta \leq \hat{\theta}$, and substituting $\hat{\theta}$ into (3.9) completes the proof. □

Note that the condition $\tau > 0$ required in Theorem 3.4, combined with the definition of λ from Lemma 3.2, immediately implies that $\gamma = O(1/\sqrt{n})$. As a result, our modified algorithm differs fundamentally from all previous methods based on partial updating (for example Kojima, Mizuno, and Yoshise (1989b), Monteiro and Adler (1989b), Anstreicher and Bosch (1992), Bosch and Anstreicher (1993), and Den Hertog, Roos, and Vial (1992)), where relative errors $\gamma = \Omega(1)$ are permitted.

COROLLARY 3.5. *Let $0 < \underline{\beta} < \overline{\beta} < 1$, where $\underline{\beta}$ and $(1-\overline{\beta})$ are $\Omega(1)$. Let $\gamma = \kappa/\sqrt{n}$ for some $\kappa > 0$, $\kappa = \Omega(1)$, where $\underline{\beta}$, $\overline{\beta}$, and κ are chosen so that $\tau > 0$, $\tau = \Omega(1)$, and $\hat{\theta} \leq 1$ in Theorem 3.4 for all $\underline{\beta} \leq \beta \leq \overline{\beta}$. Let $\mu > 0$, and suppose that x and s are $\overline{\beta}$-centered with respect to μ. Then at most $O(n)$ inner steps are required to obtain solutions that are $\underline{\beta}$-centered with respect to μ.*

Proof. The corollary follows immediately from repeated application of Theorem 3.4. □

The conditions on $\underline{\beta}$, $\overline{\beta}$, and κ required by Corollary 3.5 are not at all restrictive. For example, using $\underline{\beta} = 1/4$ and $\overline{\beta} = 2/3$, as suggested in the previous section, it is easy to verify that $\kappa = 1/8$ gives $\tau \geq .35$, and $\hat{\theta} \leq 1$, for all $\underline{\beta} \leq \beta \leq \overline{\beta}$, and $n \geq 5$.

Next we give a detailed description of what happens after the modified algorithm takes an outer step. At this point, we have x and s that are $\overline{\beta}$-centered with respect to the most recent value of μ. First, we set $k := 0$, $x^0 := x$, $s^0 := s$, $\tilde{x}^0 := x$, $\tilde{s}^0 := s$. Then we factorize $A\tilde{X}\tilde{S}^{-1}A^T$. Finally, we take as many inner steps as are needed to obtain solutions that are $\underline{\beta}$-centered with respect to μ, performing updates as needed so as to always satisfy (3.1). In other words, we perform the following procedure:

while $\| X^k s^k - \mu e \|_\infty > \underline{\beta}\mu$ **do**
  Set $x^{k+1} := x^k + \hat{\theta}^k d_x^k$ and $s^{k+1} := s^k + \hat{\theta}^k d_s^k$
  Set $U_x^k := \emptyset$ and $U_s^k := \emptyset$
  for $i = 1, \ldots, n$ **do**
    if $|(x_i^{k+1} - \tilde{x}_i^k)/\tilde{x}_i^k| > \gamma$ **then** set $\tilde{x}_i^{k+1} := x_i^{k+1}$ and $U_x^k := U_x^k \cup \{i\}$
      else set $\tilde{x}_i^{k+1} := \tilde{x}_i^k$
    if $|(s_i^{k+1} - \tilde{s}_i^k)/\tilde{s}_i^k| > \gamma$ **then** set $\tilde{s}_i^{k+1} := s_i^{k+1}$ and $U_s^k := U_s^k \cup \{i\}$
      else set $\tilde{s}_i^{k+1} := \tilde{s}_i^k$
    if either $\tilde{x}_i^{k+1} \neq \tilde{x}_i^k$ or $\tilde{s}_i^{k+1} \neq \tilde{s}_i^k$ **then** perform the corresponding rank-one update of the factorization of $A\tilde{X}\tilde{S}^{-1}A^T$
  Set $k := k + 1$

The set U_x^k contains the indices of the components of $\tilde{x}$ that need to be updated on the kth inner step. The set U_s^k is similarly defined, but pertains to $\tilde{s}$. See for example Shanno (1988) for details of updating a Cholesky factorization of $A\tilde{X}\tilde{S}^{-1}A^T$.

Since the outer step is no different from that of the basic algorithm, the modified algorithm continues to require at most $O(L)$ outer steps. From Corollary 3.5, we know that at most $O(n)$ inner steps are needed per outer step. Hence, the modified algorithm requires at most $O(nL)$ inner steps in total. In the sequel we show that at most $O(n^{1.5})$ rank-one updates are performed per outer step, where each update requires $O(n^2)$ operations. This implies that the modified algorithm has a total complexity of $O(n^{3.5}L)$ operations. Our analysis is similar to that presented in Bosch and Anstreicher (1995). The main result is Theorem 3.8. We begin by defining $\phi_x^k = \| \tilde{X}_k^{-1}(x^k - \tilde{x}^k) \|_1$ and $\phi_s^k = \| \tilde{S}_k^{-1}(s^k - \tilde{s}^k) \|_1$.

LEMMA 3.6. *Let x^k and s^k be β^k-centered with respect to μ, and consider a step of the modified algorithm $x^{k+1} = x^k + \hat{\theta}^k d_x^k$, $s^{k+1} = s^k + \hat{\theta}^k d_s^k$. Then*

$$\| \tilde{X}_k^{-1}(x^{k+1} - \tilde{x}^k) \|_1 \leq \frac{\beta^k(1+\gamma)^2}{1-\beta^k} n \hat{\theta}^k + \phi_x^k,$$

$$\| \tilde{S}_k^{-1}(s^{k+1} - \tilde{s}^k) \|_1 \leq \frac{\beta^k(1+\gamma)^2}{1-\beta^k} n \hat{\theta}^k + \phi_s^k.$$

Proof. First, note that

$$\begin{aligned} \| \tilde{X}_k^{-1}(x^{k+1} - \tilde{x}^k) \|_1 &= \| \tilde{X}_k^{-1}(x^{k+1} - x^k + x^k - \tilde{x}^k) \|_1 \\ &\leq \hat{\theta} \| \tilde{X}_k^{-1} d_x^k \|_1 + \phi_x^k. \end{aligned} \tag{3.10}$$

From (3.3), we have $\| \tilde{X}_k^{-1} d_x^k \|_1 = \| \tilde{X}_k^{-.5} \tilde{S}_k^{-.5} p^k \|_1 \leq \sqrt{n} \| \tilde{X}_k^{-.5} \tilde{S}_k^{-.5} e \|_\infty \| p^k \|_2$, and in addition $\| p^k \|_2 \leq \| r^k \|_2 \leq \| \tilde{X}_k^{-.5} \tilde{S}_k^{-.5} e \|_\infty \| \mu e - X_k s^k \|_2$. Combining the two yields

$$\begin{aligned} \| \tilde{X}_k^{-1} d_x^k \|_1 &\leq \sqrt{n} \| \tilde{X}_k^{-.5} \tilde{S}_k^{-.5} e \|_\infty^2 \| \mu e - X_k s^k \|_2 \\ &\leq n \| \tilde{X}_k^{-.5} \tilde{S}_k^{-.5} e \|_\infty^2 \| \mu e - X_k s^k \|_\infty. \end{aligned}$$

Applying Lemma 3.1 and the fact that x^k and s^k are β^k-centered with respect to μ, we obtain

$$\| \tilde{X}_k^{-1} d_x^k \|_1 \leq n \frac{(1+\gamma)^2}{(1-\beta^k)\mu} \beta^k \mu.$$

The first part of the lemma follows from the above inequality and (3.10). The proof of the second part of the lemma is nearly identical. □

LEMMA 3.7. *Let x^k and s^k be β^k-centered with respect to μ, and consider a step of the modified algorithm $x^{k+1} = x^k + \hat{\theta}^k d_x^k$, $s^{k+1} = s^k + \hat{\theta}^k d_s^k$. Then*

$$\gamma |U_x^k| \leq \phi_x^k - \phi_x^{k+1} + \frac{\beta^k(1+\gamma)^2}{1-\beta^k} n \hat{\theta}^k,$$

$$\gamma |U_s^k| \leq \phi_s^k - \phi_s^{k+1} + \frac{\beta^k(1+\gamma)^2}{1-\beta^k} n \hat{\theta}^k.$$

Proof. As in the proof of Lemma 2.2 of Bosch and Anstreicher (1995), we have

$$\begin{aligned} \phi_x^{k+1} &= \sum_{i=1}^n \left| \frac{x_i^{k+1} - \tilde{x}_i^{k+1}}{\tilde{x}_i^{k+1}} \right| \\ &= \sum_{i \notin U_x^k} \left| \frac{x_i^{k+1} - \tilde{x}_i^k}{\tilde{x}_i^k} \right| \\ &= \sum_{i=1}^n \left| \frac{x_i^{k+1} - \tilde{x}_i^k}{\tilde{x}_i^k} \right| - \sum_{i \in U_x^k} \left| \frac{x_i^{k+1} - \tilde{x}_i^k}{\tilde{x}_i^k} \right| \\ &\leq \| \tilde{X}_k^{-1}(x^{k+1} - \tilde{x}^k) \|_1 - \gamma |U_x^k|. \end{aligned}$$

The first part of the lemma follows from the above inequality and Lemma 3.6. The proof of the second part of the lemma is nearly identical. □

THEOREM 3.8. *Assume that $\underline{\beta}$ and $(1-\overline{\beta})$ are $\Omega(1)$, and let K be the number of inner steps performed between two outer steps of the modified algorithm. If $\gamma = \kappa/\sqrt{n}$ for some positive constant κ independent of n, then $\sum_{k=0}^{K-1} \left(|U_x^k| + |U_s^k|\right) = O(n^{1.5})$.*

Proof. By Lemma 3.7,

$$\gamma \sum_{k=0}^{K-1} |U_x^k| \leq \sum_{k=0}^{K-1} \left(\phi_x^k - \phi_x^{k+1} + \frac{\beta^k (1+\gamma)^2}{1-\beta^k} n \hat{\theta}^k \right)$$
$$= \phi_x^0 - \phi_x^K + n(1+\gamma)^2 \sum_{k=0}^{K-1} \frac{\beta^k}{1-\beta^k} \hat{\theta}^k.$$

Then $\sum_{k=0}^{K-1} |U_x^k| = O(n^{1.5})$ follows from $\phi_x^0 = 0$, $\phi_x^K \geq 0$, $\gamma = \kappa/\sqrt{n}$, $K = O(n)$, $\underline{\beta} \leq \beta^k \leq \overline{\beta}$, and $\hat{\theta}^k = O(1/n)$ for all k. The argument that $\sum_{k=0}^{K-1} |U_s^k| = O(n^{1.5})$ is similar. □

As noted above, the analysis of this section demonstrates that the modified algorithm, suitably initialized, has a complexity of $O(nL)$ steps, and $O(n^{3.5}L)$ total operations. Although the steplength $\hat{\theta}$ is used in Theorem 3.4 to establish a guaranteed descent in the centering measure $\| X(\theta)s(\theta)/\mu - e \|_\infty$, it is interesting to consider the use of a linesearch of this measure in $\theta \geq 0$, as suggested for the basic algorithm of the previous section, to improve the practical performance of the algorithm. In the context of the modified algorithm, such a linesearch is potentially problematic due to the fact that longer steplengths lead to more updates, as is clear from the role played by $\hat{\theta}^k$ in Theorem 3.8. This issue has been dealt with in a number of papers concerned with partial updating algorithms; see for example Anstreicher and Bosch (1992), Bosch and Anstreicher (1993), and Den Hertog, Roos, and Vial (1992). In the context of the present algorithm, the basic principle is that a linesearch is acceptable so long as the reduction in the centering measure is commensurate with the steplength. In particular, consider an inner step $x(\theta)$, $s(\theta)$, starting at x and s which are β-centered with respect to μ. Let $\beta' = \| X(\theta)s(\theta)/\mu - e \|_\infty$. Then an appropriate "safeguard" condition on the steplength θ is

$$\frac{\beta - \beta'}{\theta} \geq \nu \frac{(1-\beta)\tau^2/n}{\hat{\theta}} = \nu \frac{\tau\beta(1+\gamma)}{2}, \tag{3.11}$$

where $0 < \nu < 1$, $\nu = \Omega(1)$, and τ and $\hat{\theta}$ are as in Theorem 3.4. It is then straightforward to show that (3.11), combined with a condition

$$\beta - \beta' \geq \Omega(1/n), \tag{3.12}$$

suffices to preserve the overall $O(n^{3.5}L)$ complexity of the modified algorithm when linesearch is used on the inner steps. Note that $\theta = \hat{\theta}$ immediately satisfies (3.11) and (3.12).

REFERENCES

K. M. ANSTREICHER AND R. A. BOSCH (1992), *Long steps in an $O(n^3L)$ algorithm for linear programming*, Math. Programming 54, pp. 251–265.

R. A. BOSCH AND K. M. ANSTREICHER (1993), *On partial updating in a potential reduction linear programming algorithm of Kojima, Mizuno, and Yoshise*, Algorithmica 9, pp. 184–197.

——— (1995), *A partial updating algorithm for linear programs with many more variables than constraints*, Optimization Methods and Software, to appear.

D. DEN HERTOG, C. ROOS, AND J.-PH. VIAL (1992), *A complexity reduction for the the long-step path-following algorithm for linear programming*, SIAM J. Optim. 2, pp. 71–87.

C. C. GONZAGA (1991), *Large-step path-following methods for linear programming, Part* I: *Barrier function method*, SIAM J. Optim. 1, pp. 268-279.

B. JANSEN, C. ROOS, T. TERLAKY, AND J.-PH. VIAL (1994), *Primal-dual algorithms for linear programming based on the logarithmic barrier method*, JOTA 83, pp. 1–26.

M. KOJIMA, S. MIZUNO, AND A. YOSHISE (1989a), *A primal-dual interior point algorithm for linear programming*, in Progress in Mathematical Programming, N. Megiddo, ed., Springer–Verlag, Berlin, pp. 29–47.

——— (1989b), *A polynomial-time algorithm for a class of linear complementarity problems*, Math. Programming 44, pp. 1–26.

P. D. LING (1993), *A new proof for the new primal-dual affine scaling interior-point algorithm of Jansen, Roos, and Terlaky*, preprint, University of East Anglia.

N. MEGIDDO (1989), *Pathways to the optimal set in linear programming*, in Progress in Mathematical Programming, N. Megiddo, ed., Springer–Verlag, Berlin, pp. 131–158.

S. MIZUNO, M.J. TODD, AND Y. YE (1993), *On adaptive-step primal-dual interior-point algorithms for linear programming*, Math. Oper. Res. 18, pp. 964–981.

R. C. MONTEIRO AND I. ADLER (1989a), *Interior path following primal-dual algorithms. Part* I: *Linear programming*, Math. Programming 44, pp. 27–41.

——— (1989b), *Interior path following primal-dual algorithms. Part* II: *Convex quadratic programming*, Math. Programming 44, pp. 43–66.

C. ROOS AND J.–PH. VIAL (1990), *Long steps with the logarithmic penalty barrier function in linear programming*, in Economic Decision Making: Games, Economics, and Optimization, J. Gabszewicz, J.-F. Richard, and L. Wolsey, eds., Elsevier Science, Amsterdam, pp. 433–441.

D. F. SHANNO (1988), *Computing Karmarkar projections quickly*, Math. Programming 41, pp. 61–71.

SIAM J. OPTIMIZATION
Vol. 5, No. 2, pp. 247–268, May 1995

003

A POTENTIAL REDUCTION ALGORITHM WITH USER-SPECIFIED PHASE I–PHASE II BALANCE FOR SOLVING A LINEAR PROGRAM FROM AN INFEASIBLE WARM START*

ROBERT M. FREUND†

Abstract. This paper develops a potential reduction algorithm for solving a linear programming problem directly from a "warm start" initial point that is neither feasible nor optimal. The algorithm is an "interior point" variety that seeks to reduce a single potential function which simultaneously coerces feasibility improvement (Phase I) and objective value improvement (Phase II). The key feature of the algorithm is the ability to specify beforehand the desired balance between infeasibility and nonoptimality in the following sense. Given a prespecified balancing parameter $\beta > 0$, the algorithm maintains the following Phase I–Phase II "β-balancing constraint" throughout

$$(c^T x - z^*) < \beta \xi^T x,$$

where $c^T x$ is the objective function, z^* is the (unknown) optimal objective value of the linear program, and $\xi^T x$ measures the infeasibility of the current iterate x. This balancing constraint can be used to either emphasize rapid attainment of feasibility (set β large) at the possible expense of good objective function values or to emphasize rapid attainment of good objective values (set β small) at the possible expense of a lower infeasibility gap. The algorithm seeks to minimize the feasibility gap while maintaining the β-balancing condition, thus solving the original linear program as a consequence. The algorithm exhibits the following advantageous features: (i) the iterate solutions monotonically decrease the infeasibility measure, (ii) the iterate solutions satisfy the β-balancing constraint, (iii) the iterate solutions achieve constant improvement in both Phase I and Phase II in $O(n)$ iterations, (iv) there is always a possibility of finite termination of the Phase I problem, and (v) the algorithm is amenable to acceleration via linesearch of the potential function.

Key words. linear program, potential function, interior point algorithm, polynomial time complexity

AMS subject classifications. 90C05, 49D35

1. Introduction.

This paper is concerned with the problem of solving a linear programming problem directly from an infeasible "warm start" solution that is hopefully close to both feasibility and to optimality. Quite often in the practice of using a linear programming model, a practitioner needs to solve many slightly altered versions of the same base case model. It makes sense in this scenario that the optimal solution (or optimal basis) of a previous version of the linear programming model should serve as an excellent warm start starting point for the current version of the model, if the two versions of the model are similar. (Here, we use the term warm start in a relative sense: the closer a given starting solution is to satisfying feasibility and optimality in some appropriate measure, then the "warmer" the starting point is. Hence, a "cold start" is a starting point that is very far from feasibility and optimality, and a "hot start" is a point that is a near optimum.) Experience with the simplex method over the years has borne this out to be true in practice; the optimal basis for a previous version of the model usually serves as an excellent starting basis for the next version of the model, even when this basis is infeasible. Intuitively, a good warm start infeasible solution (that is not very infeasible and whose objective value is not far from optimality) should give an algorithm valuable information and should be a good starting point for an algorithm that will solve the linear programming model to feasibility and optimality. In spite of the success of warm start solutions in solving linear programming problems efficiently with the simplex method, there is no underlying complexity analysis that guarantees faster running times for such starting solutions, undoubtedly due to the inevitable combinatorial aspects of the simplex method itself.

* Received by the editors November 4, 1991; accepted for publication (in revised form) December 22, 1993.
† School of Operations Research and Industrial Engineering, College of Engineering, Cornell University, Ithaca, New York 14853-3801. Present address, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, Massachusetts 02142 (rfreund@mit.edu). This research was supported in part by the National Science Foundation, Air Force Office of Scientific Research, and Office of Naval Research through National Science Foundation grant DMS-8920550.

Interior point methods avoid the combinatorial problems of the simplex method, and so have the promise to yield complexity analyses that guarantee better time bounds for such warm start solutions; indeed, this is one of the motivations of this study.

In the case of interior point algorithms for linear programming, the research on algorithms for solving a linear program directly from an infeasible warm start is part of the research on combined Phase I–Phase II methods for linear programming. The underlying strategy in a combined Phase I–Phase II algorithm is to simultaneously work on the Phase I problem (to attain feasibility) and the Phase II problem (to attain optimality). The starting point for such an algorithm then need not be feasible, and a warm start starting point should again serve as an excellent starting point for a combined Phase I–Phase II algorithm. Perhaps the first interior point combined Phase I–Phase II algorithm is de Ghellinck and Vial [9]. Anstreicher [1] also contributed to the early literature in this area; see, also, Todd [15] and Todd and Wang [16]. These approaches all used the strategy of potential reduction and projective transformations, as originally developed by Karmarkar [12]. Other approaches to the problem using trajectories of optimal solutions to parametric families of shifted barrier problems were studied by Gill et al. [10], Freund [7], and Polyak [13]. Later, after direct potential reduction methods were developed by Gonzaga [11], Ye [20], and Freund [6], these methods were extended to the combined Phase I–Phase II problem; see Freund [8], Anstreicher [2], and Todd [17].

While all of these algorithms simultaneously solve both the Phase I and Phase II problems, they are all interior point algorithms and so they are only guaranteed to converge to a solution. The algorithm is terminated in theory after the appropriate gap (feasibility gap for Phase I, duality gap for Phase II) is less than 2^{-L}, where L is the bit-size representation of the problem data, and is terminated in practice when this gap is less than some prescribed small number, e.g., 10^{-6}.

The formulation of the Phase I–Phase II problem that has been developed by Anstreicher [1], [2] is to solve the linear program

$$\begin{array}{lll} \underset{x}{\text{minimize}} & c^T x & \\ \text{s.t.} & Ax & = b \\ & \xi^T x & = 0 \\ & x & \geq 0, \end{array}$$

where we are given an infeasible warm start vector x^0 that is feasible for the Phase I problem

$$\begin{array}{lll} \underset{x}{\text{minimize}} & \xi^T x & \\ \text{s.t.} & Ax & = b \\ & \xi^T x & \geq 0 \\ & x & \geq 0, \end{array}$$

and has the Phase I objective value $\xi^T x^0 > 0$. If z^* is the optimal value of LP, then $c^T x - z^*$ measures the *optimal value gap* and $\xi^T x$ measures the *feasibility gap*.

1.1. The balance of priorities between Phase I and Phase II. One might think that in solving any linear programming problem, that both Phase I and Phase II are equally important, for surely, without feasibility, the problem is not solved, and without optimality, the problem is not optimized. However, in practice, there are many instances where this simple logic breaks down, and different problems naturally lend themselves to very different ways of prioritizing the balance between improving the Phase I objective, i.e., reducing the feasibility gap, and improving the Phase II objective, i.e., improving the optimal value gap. Consider the following list of instances.

(i) In some practical modeling problems, the constraints of the problem are specified easily, but the objective function is not so easy to specify. This may be because of accounting criteria and the problem of ascertaining the true "variable cost" of the activities. Or it may be because it is not clear just what the actual objective is in the practical problem. In these problems, attaining feasibility is important, but attaining exact optimality is not so important, because of a lack of confidence that the linear programming objective function is a good representation of the true objective of the underlying practical problem.

(ii) There are instances of practical situations where the user is primarily interested in obtaining a feasible solution, and the objective function is not very important. In these instances, much more priority should be given to Phase I than to Phase II.

(iii) In other practical modeling problems, a feasible solution that is not optimal may be of no use at all. This type of situation arises frequently when using linear programming to solve partial equilibrium economic models, see, e.g., Wagner [19]. In these models, feasibility may be easy to attain, but the partial equilibrium solution is obtained by looking at both the primal and the prices that arise as the solution to the dual problem. A nonoptimal primal feasible solution conveys virtually no information about the underlying economic model. In this application of linear programming, Phase II should achieve a much higher priority than it would in instances (i) or (ii) above.

(iv) In using linear programming in branch and bound routines for solving mixed integer programming problems, a sequence of linear programs is generated and solved as the branch and bound routine runs its course. When solving a particular one of these linear programs, we may only be interested in looking at the bounds generated by the algorithm. In this case, attaining a feasible solution may be completely unnecessary, and it may suffice to generate a bound that is sufficiently positive to signal that the branch of the underlying tree should be pruned. In this case, attaining feasibility may be unimportant, and should receive much less priority than it would in instances (i) and (ii) above.

These instances suggest that an algorithm for the combined Phase I–Phase II problem should have as a parameter some measure of the relative importance or "balance" between the goals of reducing the feasibility gap (Phase I) and reducing the optimal value gap (Phase II) in solving a given linear programming problem. In this paper, we propose a measure of this balance concept, a parameter for setting this measure for a particular problem, and a polynomial time algorithm for solving the linear programming problem from an infeasible warm start that incorporates this measure and parameter into the algorithm. The notion that we develop herein is denoted as "β-balancing" and is developed as follows.

Let β be a positive scalar constant that is specified by the user, called the "balancing parameter." Given the prespecified balancing parameter $\beta > 0$, the algorithm maintains the following Phase I–Phase II β-balancing constraint throughout:

$$(c^T x - z^*) < \beta \xi^T x, \tag{1.1}$$

where $c^T x$ is the objective function, z^* is the (unknown) optimal objective value of the linear program, and $\xi^T x$ measures the infeasibility of the current iterate x. The left side of (1.1) is the *optimal value gap*, and the right side is β times the *feasibility gap*. Thus (1.1) states that the optimal value gap must be less than or equal to β times the feasibility gap.

If β is set to be very large, then (1.1) does not coerce a very tight optimal value gap. And so even when the feasibility gap is small, the optimal value gap can still be quite large (although when the feasibility is zero, clearly from (1.1) the optimal value gap must also be zero). Thus the larger the value of β, the more Phase II is deemphasized, i.e., the more Phase I is emphasized.

If β is set to be very small, then (1.1) coerces a very tight (or negative) optimal value gap as the feasibility gap is narrowed to zero. Therefore, even for a relatively large infeasibility

gap, the optimal value gap must be small (or even negative) in order to satisfy (1.1). Thus the smaller the value of β, the more Phase II is emphasized in the algorithm.

The algorithm developed in this paper seeks to minimize the feasibility gap $\xi^T x$ of the Phase I problem while maintaining the β-balancing constraint (1.1), thus solving the original linear program as a consequence. The algorithm developed in this paper also has the following other desirable features: (i) the iterate solutions monotonically decrease the infeasibility gap, (ii) the iterate solutions satisfy the β-balancing constraint (1.1), (iii) the iterate solutions achieve constant improvement in both Phase I and Phase II in $O(n)$ iterations, (iv) there is a possibility of finite termination of the Phase I problem (whether or not the objective values are superoptimal), and (v) the algorithm is amenable to acceleration via linesearch of the potential function.

The paper is organized as follows. In §2, the notation used in the paper is presented, the formulation of the warm start problem is presented, and the β-balancing constraint is developed and discussed. Also, we show how to convert any linear programming problem with an infeasible warm start and an initial objective function lower bound into the standard form that also satisfies the β-balancing constraint (1.1). Section 3 contains the development of the potential reduction problem that will be used to solve the linear programming problem and contains convergence properties of the potential reduction problem. Section 4 describes the algorithm that is used to solve (in a limiting sense) the potential reduction problem developed in §3. Section 5 discusses modifications and enhancements to the algorithm of §4 that are designed to speed convergence and give more useful information. In particular, §5 discusses ways to accelerate the algorithm via linesearches, improved dual updates via Fraley's restriction of the dual problem [5], finite termination of the Phase I problem, and obtaining explicit convergence constants related to the potential function.

2. Notation, problem formulation, and conversions. *Notation.* Throughout the paper, e denotes the vector of ones, $e = (1, 1, \ldots, 1)^T$, where the dimension is n. For any vector $\bar{x}$, etc., $\bar{X}$ denotes the diagonal matrix whose diagonal components correspond to $\bar{x}$. If $v \in \mathbb{R}^n$, $\|v\|$ denotes the Euclidean norm, i.e.,

$$\|v\| = \left(\sum_{j=1}^{n} v_j^2 \right)^{1/2}.$$

2.1. Problem formulation and the β-balancing constraint. The combined Phase I–Phase II linear programming problem is usually expressed in the following format.

$$LP: \qquad z^* = \underset{x}{\text{minimize}}\ c^T x \tag{2.1a}$$

$$\text{s.t.} \qquad Ax = b, \tag{2.1b}$$

$$\xi^T x = 0, \tag{2.1c}$$

$$x \geq 0, \tag{2.1d}$$

where we assume that there is a given infeasible warm start vector x^0 that satisfies $Ax^0 = b$ (2.1b), $x^0 \geq 0$ (2.1d), but for which $\xi^T x^0 > 0$; see [1], [2], and [16]. Thus x^0 is "almost feasible" for *LP*, and the extent to which x^0 is infeasible is precisely the quantity $\xi^T x^0$. (At the end of this section, we show how to convert any linear programming problem with an initial

infeasible warm start into an instance of *LP* above.) Considering *LP* as the Phase II problem, the Phase I problem for *LP* then is the following problem.

$$P1: \quad \underset{x}{\text{minimize}}\ \xi^T x \tag{2.2a}$$

$$\text{s.t.} \quad Ax = b \tag{2.2b}$$

$$\xi^T x \geq 0 \tag{2.2c}$$

$$x \geq 0, \tag{2.2d}$$

and now note that x^0 is feasible (but not optimal) for $P1$. We also assume that we are given an initial lower bound B^0 on the optimal value z^* of *LP*, i.e., $B^0 \leq z^*$. Such a bound may be readily available or can be produced by the algorithm given in Todd [17].

In the design of an algorithm for solving *LP* and $P1$, which will produce iterate values $x^1, x^2, x^3, \ldots$, we would like $\xi^T x^k \to 0$ as $k \to \infty$, i.e., the iterates converge to a feasible solution to *LP* (and solve $P1$ to optimality). We also would like $c^T x^k \to z^*$ as $k \to \infty$, i.e., the iterate objective values converge to the optimal objective value. Let $\beta > 0$ be a given (user-specified) balancing parameter that will be used to enforce the following Phase I–Phase II balancing condition at each iteration

$$c^T x^k - z^* < \beta \xi^T x^k. \tag{2.3}$$

The left side of (2.3) is the *optimal objective value gap* at iteration k, and the right side is β times the *feasibility gap*. Thus (2.3) states that the optimal objective value gap must be less than β times the feasibility gap. An alternate way to write (2.3) is

$$\frac{c^T x^k - z^*}{\xi^T x^k} < \beta. \tag{2.4}$$

In this form, we see that the ratio of the optimal objective value gap to the infeasibility gap cannot exceed β.

If β is given and (2.3) is enforced throughout the algorithm, then β acts as a prespecified balancing factor that will bound the optimal objective value gap in terms of the feasibility gap. For example, if feasibility is much more important than optimality, then β can be chosen to be a large number (β = 1,000, for example), whereby from (2.3) we see that the feasibility gap does not coerce a small optimal value gap. If, on the other hand, staying near the optimal objective value is more important, then β can be chosen to be a small number ($\beta = 0.001$, for example). Then from (2.3), the feasibility gap does coerce a small optimal objective value gap. From (2.3), we see that at iteration k, the deviation from the optimal objective value $(c^T x^k - z^*)$ is bounded in terms of the extent of infeasibility $(\xi^T x^k)$ by the constant β, i.e.,

$$c^T x^k - z^* < \beta \xi^T x^k.$$

However, z^* is not known in advance; only a lower bound B^0 on z^* is known in advance. The algorithm developed in this paper will produce an increasing sequence of bounds $B^1, B^2, \ldots$, on z^*, where B^k is the bound produced at iteration k. Since we do not know z^* in advance, the algorithm will enforce the following balancing condition:

$$c^T x^k - B^k < \beta \xi^T x^k, \tag{2.5}$$

where (2.5) is identical to (2.3) except that z^* is replaced by the bound B^k. Note that since $B^k \leq z^*$, then (2.5) implies (2.3), i.e.,

$$c^T x^k - z^* < \beta \xi^T x^k \quad \text{whenever} \quad c^T x^k - B^k < \beta \xi^T x^k.$$

We can rearrange (2.5) into the more standard format

$$(-\beta\xi + c)^T x^k < B^k. \tag{2.6}$$

We refer to (2.6) as the β-balancing constraint at iteration k. To satisfy (2.6) at the start of the algorithm, we need the initial assumption that $(-\beta\xi + c)^T x^0 < B^0$. (At the end of this section, we show how to convert any linear programming problem with an infeasible warm start x^0 into an instance of *LP* for which $(-\beta\xi + c)^T x^0 < B^0$ is satisfied.) We now summarize the data and other assumptions we use for the rest of this study.

Assumption 1. The given data for *LP* is the array $(A, \xi, b, c, x^0, B^0, \beta)$.

Assumption 2. $Ax^0 = b$, $\xi^T x^0 > 0$, $x^0 > 0$, $B^0 \leq z^*$.

Assumption 3. $\beta > 0$ and $(-\beta\xi + c)^T x^0 < B^0$.

Assumption 4. The set of optimal solutions of *LP* is a bounded set.

Assumption 5. $n \geq 3$.

Assumptions 1–3 have been reviewed above. Assumption 4 is a standard (though non-trivial) assumption needed for convergence of all interior point algorithms. (See Vial [18] and Anstreicher [4] for ways to mitigate this assumption.) Assumption 5 is trivial, since for $n \leq 2$ the problem *LP* lends itself to instant analysis. We now show how to convert a linear program satisfying Assumptions 4 and 5 into the standard form *LP* of (2.1) and that satisfies Assumptions 1–5.

2.2. Converting a linear program into an instance of *LP* satisfying Assumptions 1–5. Suppose we want to solve the linear program

$$\widehat{LP}: \qquad z^* = \underset{\hat{x}}{\text{minimize}}\ \hat{c}^T \hat{x} \tag{2.7a}$$

$$\text{s.t.} \qquad \hat{A}\hat{x} = b, \tag{2.7b}$$

$$\hat{x} \geq 0, \tag{2.7c}$$

where $\hat{A}$ is $m \times n$ and it is assumed that $n \geq 3$, and $\hat{x}^0$ is a given warm start that is hopefully near feasible and near optimal. Also suppose that B^0 is a known given lower bound on z^*. Then the given data for the problem $\widehat{LP}$ is the array $(\hat{A}, b, \hat{c}, \hat{x}^0, B^0)$. In a typical situation, $\hat{x}^0$ may be the optimal solution to a previous version of $\widehat{LP}$ that is hopefully a good near feasible and near optimal for the current linear program $\widehat{LP}$. Alternatively, $\hat{x}^0$ may be a basic solution to $\widehat{LP}$ for a basis that is suspected of being close to the optimal basis. Knowledge of B^0 can be given in a number of ways. If L is the size of the array $(\hat{A}, b, \hat{c})$ (i.e., L is the number of bits needed to encode the data $(\hat{A}, b, \hat{c})$ in binary form), then one value of B^0 that can be used is -2^L, but this is not practical. A more practical approach would be to set B^0 to be some large negative number such as -10^{12}. However, if the user has a good knowledge of the program $\widehat{LP}$, he/she may be able to set B^0 fairly accurately. (For example, suppose $\widehat{LP}$ is a refinery problem. Then it is reasonable that a lower bound on the cost of operating the refinery is readily apparent from knowledge of the data that have been used to generate the program $\widehat{LP}$.) It should also be pointed out that an algorithm for generating a reasonable bound B^0 has been developed in Todd [17].

We first assume that $\hat{x}^0$ satisfies (2.7b), i.e., $\hat{A}\hat{x}^0 = b$. This will certainly be the case if $\hat{x}^0$ is a basic solution for a (hopefully near-optimal) basis of $\hat{A}$. If $\hat{x}^0$ does not satisfy (2.7b), then

$\hat{x}^0$ can be projected onto the linear manifold $\{\hat{x}|\hat{A}\hat{x} = b\}$ by choosing any suitable projection, e.g.,

$$\hat{x}^0 \leftarrow D[I - D\hat{A}^T(\hat{A}D^2\hat{A}^T)^{-1}A\hat{D}]\hat{x}^0 + D^2\hat{A}^T(\hat{A}D^2\hat{A}^T)^{-1}b,$$

where D is any positive definite matrix (e.g., $D = I$ or D is a positive diagonal matrix). It is assumed that $\hat{x}^0 \not> 0$, for otherwise $\hat{x}^0$ would be an interior feasible solution to (2.7) and there would be no need for a Phase I procedure to be part of the solution to (2.7).

Now let $h \geq 0$ be any vector that satisfies $\hat{x}^0 + h > 0$. Then our problem $\widehat{LP}$ is equivalent to

$$\widehat{LP}^2: \quad \min_{\hat{x},\hat{w}} c^T\hat{x} \tag{2.8a}$$

$$\text{s.t.} \quad \hat{A}\hat{x} = b, \tag{2.8b}$$

$$\hat{x} + \hat{w}h \geq 0, \tag{2.8c}$$

$$\hat{w} = 0, \tag{2.8d}$$

where we note that $(\hat{x}, \hat{w}) = (\hat{x}^0, 1)$ is feasible for (2.8) except for the last constraint (2.8d), which measures the infeasibilities of $\hat{x}^0$. If $\widehat{LP}^2$ is the Phase II problem, then the Phase I problem can be written as

$$\widehat{LP}^1: \quad \underset{\hat{x},\hat{w}}{\text{minimize}}\ \hat{w} \tag{2.9a}$$

$$\text{s.t.} \quad \hat{A}\hat{x} = b, \tag{2.9b}$$

$$\hat{x} + \hat{w}h \geq 0, \tag{2.9c}$$

$$\hat{w} \geq 0. \tag{2.9d}$$

Notice that $(\hat{x}, \hat{w}) = (\hat{x}^0, 1)$ is feasible for $\widehat{LP}^1$ and, in fact,

$$(\hat{x}, \hat{w}) = (\hat{x}^0, \varepsilon)$$

is feasible for $\widehat{LP}^1$ for any $\varepsilon \geq 1$, due to the fact that $h \geq 0$.

Let β be the prespecified balancing parameter discussed previously. Then if $\hat{w}^0$ is given by

$$\hat{w}^0 = \max\left\{1,\ 1 + \frac{(\hat{c}^T\hat{x}^0 - B^0)}{\beta}\right\}, \tag{2.10}$$

then $(\hat{x}^0, \hat{w}^0)$ will satisfy

$$\hat{A}\hat{x}^0 = b, \tag{2.11a}$$

$$\hat{x}^0 + \hat{w}^0 h > 0, \tag{2.11b}$$

$$(-\beta\hat{w}^0 + \hat{c}^T\hat{x}^0) < B^0. \tag{2.11c}$$

Thus the pair $(\hat{x}^0, \hat{w}^0)$ is feasible for the Phase I problem $\widehat{LP}^1$ (2.9) and also satisfies (2.11c), which is the analog of Assumption 3 for this problem. Also, $(\hat{x}^0, \hat{w}^0)$ satisfies all constraints of $\widehat{LP}^2$ (2.8) except (2.8d).

To convert $\widehat{LP}^2$ (2.8) to an instance of LP (2.1), we proceed as follows. First, let x denote the slack vector

$$x = \hat{x} + \hat{w}h. \tag{2.12}$$

Then $x^0 = \hat{x}^0 + \hat{w}^0 h > 0$ denotes the starting slack variables for the starting solution $(\hat{x}^0, \hat{w}^0)$ of $\widehat{LP}^1$ (2.9).

The next step is to eliminate the variable $\hat{w}$ from the systems (2.8) and (2.9). To do this, assume with no loss of generality that the vectors $\hat{A}h$ and b are not linearly independent. (If this is not the case, a perturbation of $h > 0$ will enforce their linear independence.) Then let $\lambda \in \mathbb{R}^m$ be any vector for which

$$\lambda^T b = 0,$$

$$\lambda^T(\hat{A}h) = 1,$$

(such a λ is simple to compute), and let

$$\xi = \hat{A}^T\lambda.$$

Then note that all x, $\hat{x}$, $\hat{w}$ that satisfy (2.8b) and (2.12) satisfy

$$\xi^T x = \lambda^T \hat{A}x = \lambda^T \hat{A}(\hat{x} + \hat{w}h) = \lambda^T b + \lambda^T \hat{A}h\hat{w} = \hat{w}. \tag{2.13}$$

Thus we can substitute $\hat{x}$ and $\hat{w}$ by x (from 2.12) and $\xi^T x$ (from 2.13) in (2.8) and (2.9). If we define

$$c = \hat{c} - \hat{c}^T h\xi \quad \text{and} \quad A = \hat{A} - \hat{A}h\xi^T,$$

then $\widehat{LP}^2$ transforms to

$$\begin{aligned} LP: \qquad & \underset{x}{\text{minimize}}\ c^T x \\ & \text{s.t.} \qquad Ax = b, \\ & \qquad\quad x \geq 0, \\ & \qquad\quad \xi^T x = 0, \end{aligned}$$

and $\widehat{LP}^1$ transforms to

$$\begin{aligned} P1: \qquad & \underset{x}{\text{minimize}}\ \xi^T x \\ & \text{s.t.} \qquad Ax = b, \\ & \qquad\quad x \geq 0, \\ & \qquad\quad \xi^T x \geq 0. \end{aligned}$$

Also, $x^0 = \hat{x}^0 + \hat{w}^0 h$ satisfies $Ax^0 = b$, $x^0 > 0$, $\xi^T x^0 = \hat{w}^0 > 0$, and (2.11c) transforms to

$$(-\beta\xi + c)^T x^0 < B^0. \tag{2.14}$$

Note that

$$\xi^T x^0 = \hat{w}^0 = \max\left\{1,\ 1 + \frac{(\hat{c}^T\hat{x}^0 - \hat{B}^0)}{\beta}\right\},$$

from (2.10). Also, if $\widehat{LP}^2$ satisfies Assumption 4, then it is easy to verify that LP does as well.

Finally, note that this construction is dependent on the choice of the vector h (h is sometimes referred to as the "shift vector"), which is used to shift the starting point $\hat{x}^0$ so that $\hat{x}^0 + h > 0$. Shift vectors have been studied from a theoretical point of view in Polyak [13], Gill et al. [10], and in Freund [7], [8]. There are many choices of h that can be used, and some computational experiments have shown that the choice of h can have a large impact on the practical performance of an algorithm. It is an open question of how to choose h most efficiently in practice.

3. The potential reduction problem for solving *LP* and convergence properties. In this section we consider solving the standard form problem

$$\begin{aligned} LP: \qquad z^* = \min_x \ & c^T x \\ & Ax = b, \\ & \xi^T x = 0, \\ & x \geq 0, \end{aligned} \tag{3.1}$$

whose dual is

$$\begin{aligned} LD: \qquad \max_{\pi,\theta,s} \ & b^T \pi \\ & A^T \pi + \xi\theta + s = c, \\ & s \geq 0. \end{aligned} \tag{3.2}$$

It is assumed that the data array $(A, \xi, b, c, x^0, B^0, \beta)$ satisfies Assumptions 1–5 of the previous section.

The Phase I problem for LP then is to solve

$$\begin{aligned} P1: \qquad \underset{x}{\text{minimize}} \ & \xi^T x \\ \text{s.t.} \quad & Ax = b, \\ & \xi^T x \geq 0, \\ & x \geq 0. \end{aligned} \tag{3.3}$$

We will not work with this problem ($P1$), but will instead augment $P1$ with the additional balancing constraint involving the lower bound B^0 on the optimal value z^* of LP and the balancing parameter β discussed in §2. Suppose B^0 is the given lower bound on z^* and that β is the balancing parameter for which the starting point x^0 satisfies

$$(-\beta\xi + c)^T x^0 < B^0; \tag{3.4}$$

see Assumption 3. (The method for satisfying (3.4) was discussed in §2.) Now consider the parametric family of augmented Phase I problems

$$\begin{aligned} P_B: \qquad z_B = \underset{x,t}{\text{minimize}} \ & \xi^T x \\ \text{s.t.} \quad & Ax = b, \\ & (-\beta\xi + c)^T x + t = B, \\ & x \geq 0, \ t \geq 0, \end{aligned} \tag{3.5}$$

whose dual is given by

$$
\begin{aligned}
D_B: \qquad z_B = \underset{\pi,\mu,s}{\text{maximize}}\ & b^T\pi - B\mu, \\
& A^T\pi - (-\beta\xi + c)\mu + s = \xi, \\
& s \geq 0, \mu \geq 0.
\end{aligned} \tag{3.6}
$$

The following are elementary properties P_B.

PROPOSITION 3.1. *For all* $B \in [B^0, z^*]$

(i) P_B *is feasible,*

(ii) $z_B > 0$ *if* $B < z^*$, $z_B = 0$ *if* $B = z^*$.

(iii) *The set of optimal solutions to* P_B *is nonempty and bounded.*

(iv) *For all* x *feasible for* P_B,

$$c^T x \leq z^* + \beta\xi^T x.$$

Proof. (i) Let $t^0 = B^0 - (-\beta\xi + c)^T x^0$. From Assumption 2 and (3.4) it follows that (x^0, t^0) is feasible for P_{B^0} and so (x^0, t) is also feasible for P_B for any $B \geq B^0$, where $t = B - (-\beta\xi + c)^T x^0$.

(ii) Suppose $B < z^*$. Then if $z_B \leq 0$, there exists x for which $Ax = b$, $x \geq 0$, $\xi^T x = 0$, $(-\beta\xi + c)^T x \leq B < z^*$, and so $c^T x < z^*$, violating the definition of z^*. Thus $z_B > 0$. If $B = z^*$, then a similar argument establishes that $z_B = 0$.

(iii) Suppose the set of optimal solutions to P_B is not bounded. Then there is a direction $d \neq 0$ that satisfies $\xi^T d = 0$, $Ad = 0$, $d \geq 0$, $(-\beta\xi + c)^T d \leq 0$, and so $c^T d \leq 0$. Therefore d is a nontrivial ray of the optimal solution set of *LP*, violating the assumption that *LP* has a bounded set of optimal solutions.

(iv) $(-\beta\xi + c)^T x \leq B$ implies $c^T x \leq \beta\xi^T x + B \leq \beta\xi^T x + z^*$. □

Now consider the following potential reduction problem related to P_B and D_B.

$$PR: \qquad \underset{x,t,B}{\text{minimize}}\ F(x,t) = q\,\ell n(\xi^T x) - \sum_{j=1}^{n} \ell n\, x_j - \ell n\, t \tag{3.7}$$

$$\text{s.t.} \qquad Ax = b, \tag{3.8a}$$

$$(-\beta\xi + c)^T x + t = B, \tag{3.8b}$$

$$x > 0,\ t > 0, \tag{3.8c}$$

$$B \leq z^*, \tag{3.9}$$

where q is a parameter satisfying $q \geq n + 1$, and (3.8) reflects feasibility for P_B, for $B \leq z^*$, which is given in (3.9). The following lemma relates potential function values to the objective function $\xi^T x$ of problem P_B.

LEMMA 3.1. *Suppose* (x^0, t^0, B^0) *is the starting point of an algorithm for solving* PR *and suppose that* $z^* < +\infty$. *Suppose* $(\bar{x}, \bar{t}, \bar{B})$ *is a feasible point generated by the algorithm, and that* $\xi^T\bar{x} \leq \xi^T x^0$. *Let*

$$\Delta = F(x^0, t^0) - F(\bar{x}, \bar{t}). \tag{3.10}$$

Then

$$\xi^T\bar{x} \leq (\xi^T x^0) C_1 e^{-\Delta/q},$$

where C_1 is computed from the optimal value of the following linear program:

$$PC: \quad (C_1)^{\left(\frac{q}{n+1}\right)} = (n+1)^{-1} \max_{x,t} e^T(X^0)^{-1}x + (t^0)^{-1}t \tag{3.11a}$$

$$\text{s.t.} \qquad Ax = b, \tag{3.11b}$$

$$(-\beta\xi + c)^T x + t \leq z^*, \tag{3.11c}$$

$$\xi^T x \leq \xi^T x^0, \tag{3.11d}$$

$$-\xi^T x \leq 0. \tag{3.11e}$$

Proof. The pair (x^0, t^0) is feasible for PC. If PC is unbounded, then there exists (d, v) satisfying $d \geq 0$, $v \geq 0$, $Ad = 0$, $(-\beta\xi + c)^T d + v \leq 0$, $\xi^T d \leq 0$, $-\xi^T d \leq 0$, and $e^T(X^0)^{-1}d + (t^0)^{-1}v > 0$. Thus $d \geq 0$, $Ad = 0$, $\xi^T d = 0$, $c^T d + v \leq 0$, $v \geq 0$, and so $c^T d \leq 0$ and $d \neq 0$, contradicting the assumption that the set of optimal solutions to LP is a bounded set. Thus C_1 is well defined, and $0 < C_1 < +\infty$.

Since $(\bar{x}, \bar{t})$ is also feasible for PC,

$$e^T(X^0)^{-1}\bar{x} + (t^0)^{-1}\bar{t} \leq (n+1)(C_1)^{\left(\frac{q}{n+1}\right)}.$$

Thus by the arithmetic–geometric mean inequality, it then follows that

$$\sum_{j=1}^{n} \ell n\bar{x}_j + \ell n\bar{t} - \sum_{j=1}^{n} \ell n x_j^0 - \ell n\, t^0 \leq q\ell n C_1. \tag{3.12}$$

Next, from (3.10) and (3.7), we have

$$\begin{aligned} q\ell n(\xi^T\bar{x}) &= q\ell n(\xi^T x^0) + \textstyle\sum_1 \ell n\bar{x}_j + \ell n\bar{t} - \sum_1 \ell n x_j^0 - \ell n t^0 - \Delta \\ &\leq q\ell n(\xi^T x^0) + q\ell n C_1 - \Delta, \end{aligned}$$

where the inequality follows from (3.12). Exponentiating and rearranging yields the result. □

Note that if the size of β, x^0, and t^0 are $O(L)$, then $C_1 \leq 2^L$, provided that $z^* < +\infty$.

We demonstrate an algorithm in §4 that reduces $F(x,t)$ by a fixed constant $\delta \geq \frac{1}{6}$ at each iteration, if $q \geq n + 1 + \sqrt{n+1}$, with the additional property that the values of $\xi^T x$ monotonically decrease at each iteration. This is the basis for the following convergence theorem.

THEOREM 3.1 (Convergence). *Suppose (x^k, t^k, B^k), $k = 0, \ldots,$ is a sequence of feasible solutions to problem PR with the property that $F(x^{k+1}, t^{k+1}) \leq F(x^k, t^k) - \frac{1}{6}$, and $\xi^T x^{k+1} \leq \xi^T x^k$, $k = 0, 1, \ldots$. Suppose that $z^* < +\infty$. Then with C_1 as given in* (3.11),

(i) $0 < \xi^T x^k \leq (\xi^T x^0) C_1 e^{-k/6q}$.

Let (π^*, θ^*, s^*) be any optimal solution to LD. Then

(ii) $-|\theta^*|(\xi^T x^0) C_1 e^{-k/6q} \leq c^T x^k - z^* \leq \beta(\xi^T x^0) C_1 e^{-k/6q}$

(iii) $-|\theta^*|(\xi^T x^0) C_1 e^{-k/6q} \leq c^T x^k - B^k \leq \beta(\xi^T x^0) C_1 e^{-k/6q}$

(iv) $0 \leq z^* - B^k \leq (\beta + |\theta^*|) C_1 e^{-k/6q}$.

Theorem 3.1(i) states that fixed improvement in the Phase I objective value $\xi^T x^k$ is obtained in $O(q)$ iterations. The convergence results in Theorem 3.1(ii) relate the convergence of the

Phase II objective value $c^T x^k$ to the optimal value z^*. Similar convergence results for the lower bounds B^k are given in (iii) and (iv) of the theorem.

Proof of Theorem 3.1. Letting $(\bar{x}, \bar{t}) = (x^k, t^k)$, (i) follows from Lemma 3.1, where from (3.10) $\Delta \geq k/6$. From the convexity properties of linear programming duality, we obtain from Proposition A.3 of Appendix A that

$$c^T x \geq z^* + \theta^* \xi^T x \text{ for any } x \text{ satisfying } Ax = b,\ x \geq 0, \tag{3.13}$$

and so

$$c^T x^k - B^k \geq c^T x^k - z^* \geq \theta^* \xi^T x^k \geq -|\theta^*|(\xi^T x^0) C_1 e^{-k/6q}. \tag{3.14}$$

Furthermore, from (3.8b), we obtain

$$c^T x^k - z^* \leq c^T x^k - B^k \leq \beta \xi^T x^k \leq \beta(\xi^T x^0) C_1 e^{-k/6q}, \tag{3.15}$$

and (3.14) and (3.15) combine to prove (ii) and (iii). (iv) is a consequence of (ii) and (iii). □

4. The algorithm for solving the potential reduction problem *PR*. In this section, we present an algorithm that obtains a decrease of $\delta \geq \frac{1}{6}$ in the potential function $F(x,t)$ of problem *PR* (3.7)–(3.9) at each iteration, and that is monotone decreasing in the values of $\xi^T x$, given $q \geq n + 1 + \sqrt{n+1}$.

Suppose the current iterate values for *PR* is the array $(\bar{x}, \bar{t}, \bar{B})$, which is feasible for PR. As in the standard potential reduction algorithm (see Ye [20], Gonzaga [11], Freund [6], and Anstreicher [3]), we seek to compute a primal direction that will decrease the potential function. Since the primal variables are $(x,t) = (\bar{x}, \bar{t})$, we seek a direction $(\tilde{d}, \tilde{r})$ and a suitable steplength α for which $F(\bar{x} - \alpha\tilde{d}, \bar{t} - \alpha\tilde{r})$ achieves a constant decrease over $F(\bar{x}, \bar{t})$. Analogous to [20], [6], and [3], we let $(\tilde{d}, \tilde{r})$ be the solution to the following optimization problem:

$$Q: \quad \underset{d,r}{\text{maximize}} \left(\frac{q}{\xi^T \bar{x}} \xi^T - e^T \bar{X}^{-1} \right) d - \bar{t}^{-1} r - \frac{1}{2} d^T \bar{X}^{-2} d - \frac{1}{2} \bar{t}^{-2} r^2 \tag{4.1a}$$

$$\text{s.t.} \qquad Ad = 0 \qquad (\pi), \tag{4.1b}$$

$$(-\beta\xi + c)^T d + r = 0 \qquad (-\theta), \tag{4.1c}$$

$$\xi^T d \geq 0 \qquad (\delta), \tag{4.1d}$$

where the quantities (π), $(-\theta)$, (δ) indicated are the dual multipliers on the constraints.

This problem has a strictly concave quadratic objective, and since $(d,r) = (0,0)$ is a feasible solution, it will attain its optimum uniquely. Program Q can be interpreted as the standard rescaled projection of the rescaled gradient of the potential function onto the null space of (3.8a)–(3.8b), with the simple monotonicity constraint (see Anstreicher [3]) added as well in (4.1d). The unique solution $(\tilde{d}, \tilde{r})$ to Q is obtained by solving the following Karush–Kuhn–Tucker conditions for $(\tilde{d}, \tilde{r}, \tilde{\pi}, \tilde{\theta}, \tilde{\delta})$:

$$A\tilde{d} = 0, \tag{4.2a}$$

$$(-\beta\xi + c)^T \tilde{d} + \tilde{r} = 0, \tag{4.2b}$$

$$\xi^T \tilde{d} \geq 0, \tag{4.2c}$$

$$\frac{q}{\xi^T\bar{x}}\xi - \bar{X}^{-1}e - \bar{X}^{-2}\tilde{d} = A^T\tilde{\pi} - (-\beta\xi + c)\tilde{\theta} - \tilde{\delta}\xi, \tag{4.2d}$$

$$-\bar{t}^{-1} - \bar{t}^{-2}\tilde{r} = -\tilde{\theta}, \tag{4.2e}$$

$$\tilde{\delta} \geq 0,\ (\xi^T\tilde{d})\tilde{\delta} = 0. \tag{4.2f}$$

(Note from (4.1) that (4.2) can readily be solved as follows as one of two systems of linear equations. First, one assumes that the monotonicity constraint (4.1d) is nonbinding, and then one solves (4.2) ignoring (4.2c) and (4.2f) and setting $\tilde{\delta} = 0$. Then the resulting system is linear. If (4.2c) is violated in the solution, then solve again (4.2) with (4.2c) at equality, again ignoring (4.2f), and the resulting value of $\tilde{\delta}$ must be nonnegative; see also Anstreicher [2] and Todd [17].) It will be convenient to set

$$\tilde{s} = \bar{X}^{-1}(e + \bar{X}^{-1}\tilde{d}) \tag{4.2g}$$

and to rewrite (4.2e) as

$$\tilde{\theta} = \bar{t}^{-1}(1 + \bar{t}^{-1}\tilde{r}). \tag{4.2h}$$

Next we define

$$\bar{\gamma} = \sqrt{\tilde{d}^T\bar{X}^{-2}\tilde{d} + (\tilde{r}/\bar{t})^2} \tag{4.3}$$

and note from (4.2g) and (4.2h) that

$$\bar{\gamma} = \sqrt{(e - \bar{X}\tilde{s})^T(e - \bar{X}\tilde{s}) + (1 - \bar{t}\tilde{\theta})^2} \tag{4.4}$$

and from (4.2a)–(4.2f) that

$$\frac{q}{\xi^T\bar{x}}\xi^T\tilde{d} - e^T\bar{X}^{-1}\tilde{d} - \bar{t}^{-1}\tilde{r} = \bar{\gamma}^2. \tag{4.5}$$

Just as in [6], for example, we have the following theorem.

THEOREM 4.1 (Primal improvement). *For* $0 \leq \alpha < 1$,

(i) $(\bar{x} - (\alpha/\bar{\gamma})\tilde{d},\ \bar{t} - (\alpha/\bar{\gamma})\tilde{r}, \bar{B})$ *is feasible for* PR,

(ii) $F(\bar{x} - (\alpha/\bar{\gamma})\tilde{d},\ \bar{t} - (\alpha/\bar{\gamma})\tilde{r}) \leq F(\bar{x}, \bar{t}) - \alpha\bar{\gamma} + \frac{\alpha^2}{2(1-\alpha)}$, *and*

(iii) *if* $\bar{\gamma} \geq \frac{4}{5}$ *and* $\alpha = \frac{2}{5}$,

$$F(\bar{x} - (\alpha/\bar{\gamma})\tilde{d},\ \bar{t} - (\alpha/\bar{\gamma})\,\tilde{r}) \leq F(\bar{x}, \bar{t}) - \frac{1}{6}.$$

Proof. (i) Since the only variables that change are x, t, (3.9) is still satisfied. Since $(\tilde{d}, \tilde{r})$ lies in the null space of (3.8a)–(3.8b), it only remains to show that $\bar{x} - (\alpha/\bar{\gamma})\tilde{d} > 0$, $\bar{t} - (\alpha/\bar{\gamma})\tilde{r} > 0$. This will follow from (4.3), which implies that

$$|(\bar{X}^{-1}\tilde{d})_j| < \bar{\gamma},\ j = 1, \ldots, n \quad \text{and} \quad |\tilde{r}/\bar{t}| < \bar{\gamma}.$$

Therefore

$$\bar{x} - (\alpha/\bar{\gamma})\tilde{d} = \bar{X}(e - (\alpha/\bar{\gamma})\bar{X}^{-1}\tilde{d}) > 0 \quad \text{for } \alpha \in [0, 1)$$

and

$$\bar{t} - (\alpha/\bar{\gamma})\tilde{r} = \bar{t}(1 - (\alpha/\bar{\gamma})(\tilde{r}/\bar{t})) > 0 \quad \text{for } \alpha \in [0,1).$$

Furthermore, from Proposition A.2 of Appendix A,

$$\begin{aligned}
(4.6)\ & \sum_{j=1}^{n} \ell n\,(1 - (\alpha/\bar{\gamma})(\bar{X}^{-1}\tilde{d})_j) + \ell n(1 - (\alpha/\bar{\gamma})(\tilde{r}/\bar{t})) \\
& \geq -(\alpha/\bar{\gamma})e^T\bar{X}^{-1}\tilde{d} - \sum_{j=1}^{n} \frac{(\alpha/\bar{\gamma})^2(\bar{X}^{-1}\tilde{d})_j^2}{2(1-\alpha)} - (\alpha/\bar{\gamma})(\tilde{r}/\bar{t}) - \frac{(\alpha/\bar{\gamma})^2(\tilde{r}/\bar{t})^2}{2(1-\alpha)} \\
& = -(\alpha/\bar{\gamma})[e^T\bar{X}^{-1}\tilde{d} + \tilde{r}/\bar{t}] - \frac{\alpha^2}{2(\bar{\gamma}^2)(1-\alpha)}\,((\tilde{d}^T\bar{X}^{-2}\tilde{d}) + (\tilde{r}/\bar{t})^2) \\
& = -(\alpha/\bar{\gamma})[e^T\bar{X}^{-1}\tilde{d} + \tilde{r}/\bar{t}] - \frac{\alpha^2}{2(1-\alpha)} \qquad \text{(from 4.3).}
\end{aligned}$$

$$\begin{aligned}
\text{(ii)}\quad & F(\bar{x} - (\alpha/\bar{\gamma})\tilde{d},\ \bar{t} - (\alpha/\bar{\gamma})\tilde{r}) - F(\bar{x},\bar{t}) \\
& = q\,\ell n\left(1 - (\alpha/\bar{\gamma})\frac{\xi^T\tilde{d}}{\xi^T\bar{x}}\right) - \sum_{j=1}^{n} \ell n(1 - (\alpha/\bar{\gamma})(\bar{X}^{-1}\tilde{d})_j) - \ell n(1 - (\alpha/\bar{\gamma})(\tilde{r}/\bar{t})) \\
& \leq -\frac{q}{\xi^T\bar{x}}\,(\alpha/\bar{\gamma})\xi^T\tilde{d} + (\alpha/\bar{\gamma})[e^T\bar{X}^{-1}\tilde{d} + \tilde{r}/\bar{t}] + \frac{\alpha^2}{2(1-\alpha)}
\end{aligned}$$

(which follows from (4.6) and Proposition A.1 of Appendix A)

$$\begin{aligned}
& = (-\alpha/\bar{\gamma})\left(\frac{q}{\xi^T\bar{x}}\,\xi^T\tilde{d} - e^T\bar{X}^{-1}\tilde{d} - \tilde{r}\bar{t}^{-1}\right) + \frac{\alpha^2}{2(1-\alpha)} \\
& = (-\alpha/\bar{\gamma})\bar{\gamma}^2 + \frac{\alpha^2}{2(1-\alpha)} \qquad \text{(from 4.5)} \\
& = -\alpha\bar{\gamma} + \frac{\alpha^2}{2(1-\alpha)}.
\end{aligned}$$

This proves (ii). Then (iii) follows by direct substitution. □

Theorem 4.1 guarantees a decrease in $F(x,t)$ if the value of $\gamma = \bar{\gamma}$ is sufficiently large, e.g., if $\bar{\gamma} \geq 4/5$. In the case when $\bar{\gamma}$ is small, we can obtain a reduction in the potential function by replacing the bound $\bar{B}$ on z^* by a new bound $\hat{B}$ generated from new dual variables $(\hat{\pi}, \hat{\theta}, \hat{s})$ for LD.

LEMMA 4.1 (Dual improvement). *Suppose* $\bar{\gamma} < 1$ *and* $q \geq n + 1 + \sqrt{n+1}$. *Define*

$$(4.7)\qquad (\hat{\pi}, \hat{\theta}, \hat{s}) = \left(\frac{\tilde{\pi}}{\tilde{\theta}},\ \beta - \frac{\tilde{\delta}}{\tilde{\theta}} - \frac{q}{\tilde{\theta}\xi^T\bar{x}},\ \frac{\tilde{s}}{\tilde{\theta}}\right),$$

where $\tilde{\pi}$, $\tilde{\theta}$, $\tilde{s}$, $\tilde{\delta}$ *are given in the solution to* (4.2). *Then* $(\hat{\pi}, \hat{\theta}, \hat{s})$ *is feasible for* LD, *with dual objective value*

$$(4.8)\qquad \hat{B} \triangleq b^T\hat{\pi} > \bar{B} + \frac{\sqrt{n+1}(1-\bar{\gamma})}{(1+\bar{\gamma})}\bar{t}.$$

Proof. If $\bar{\gamma} < 1$, it follows from (4.4) that $\tilde{s} > 0$ and $\tilde{\theta} > 0$. Therefore $(\hat{\pi}, \hat{\theta}, \hat{s})$ is well defined (since we also have $\xi^T \bar{x} > 0$), and $\hat{s} > 0$. Then it is easily verified from (4.2d) and (4.2g) that $A^T\hat{\pi} + \xi\hat{\theta} + \hat{s} = c$, and so $(\hat{\pi}, \hat{\theta}, \hat{s})$ is feasible for LD, with dual objective value $\hat{B} \triangleq b^T\hat{\pi} = \bar{x}^T A^T \hat{\pi}$

$$
\begin{aligned}
&= c^T\bar{x} - \bar{x}^T\hat{s} - \xi^T\bar{x}\hat{\theta} \\
&= c^T\bar{x} - \frac{1}{\tilde{\theta}} e^T(e + \bar{X}^{-1}\tilde{d}) - \xi^T\bar{x}\left(\beta - \frac{\tilde{\delta}}{\tilde{\theta}} - \frac{q}{\tilde{\theta}\xi^T\bar{x}}\right) \quad \text{(from (4.2g) and (4.7))} \\
&= c^T\bar{x} - \beta\xi^T\bar{x} + \frac{1}{\tilde{\theta}}(-n - e^T\bar{X}^{-1}\tilde{d} + \tilde{\delta}\xi^T\bar{x} + q) \\
&= \bar{B} - \bar{t} + \frac{1}{\tilde{\theta}}(q - n - e^T\bar{X}^{-1}\tilde{d}) + \frac{\bar{\delta}\xi^T\bar{x}}{\tilde{\theta}} \quad \text{(from (3.8b))} \\
&= \bar{B} + \frac{1}{\tilde{\theta}}(q - n - \tilde{\theta}\bar{t} - e^T\bar{X}^{-1}\tilde{d}) + \frac{\bar{\delta}\xi^T\bar{x}}{\tilde{\theta}} \\
&= \bar{B} + \frac{1}{\tilde{\theta}}(q - n - (1 + \tilde{r}/\bar{t}) - e^T\bar{X}^{-1}\tilde{d}) + \frac{\bar{\delta}\xi^T\bar{x}}{\tilde{\theta}} \quad \text{(from (4.2h))} \\
&= \bar{B} + \frac{1}{\tilde{\theta}}(q - (n+1) - [e^T\bar{X}^{-1}d + \tilde{r}/\bar{t}]) + \frac{\bar{\delta}\xi^T\bar{x}}{\tilde{\theta}} \\
&\geq \bar{B} + \frac{1}{\tilde{\theta}}(\sqrt{n+1} - \sqrt{n+1}\,\bar{\gamma}) + \frac{\bar{\delta}\xi^T\bar{x}}{\tilde{\theta}} \quad \text{(from (4.3))} \\
&\geq \bar{B} + \frac{(1-\bar{\gamma})\sqrt{n+1}}{\tilde{\theta}} \geq \bar{B} + \frac{\bar{t}(1-\bar{\gamma})\sqrt{n+1}}{(1+\bar{\gamma})},
\end{aligned}
$$

where the last inequality follows from (4.4), which implies that $\bar{t}\tilde{\theta} \leq (1 + \bar{\gamma})$. □

We now can prove the following theorem.

THEOREM 4.2 (Dual improvement). *Suppose* $\bar{\gamma} < 1$ *and* $q \geq n + 1 + \sqrt{n+1}$. *Define* $(\hat{\pi}, \hat{\theta}, \hat{s})$ *as in* (4.7), $\hat{B}$ *as in* (4.8), *and let*

$$\hat{t} = \bar{t} + \hat{B} - \bar{B}. \tag{4.9}$$

Then

(i) $(\bar{x}, \hat{t}, \hat{B})$ *is feasible for PR and*

$$F(\bar{x}, \hat{t}) \leq F(\bar{x}, \bar{t}) - \ell n\left(1 + \frac{\sqrt{n+1}(1-\bar{\gamma})}{1+\bar{\gamma}}\right).$$

(ii) *If* $\bar{\gamma} < 4/5$, *then*

$$F(\bar{x}, \hat{t}) \leq F(\bar{x}, \bar{t}) - \frac{1}{6}.$$

Proof. Because $(\hat{\pi}, \hat{\theta}, \hat{s})$ is feasible for LD and $\hat{B} = b^T\hat{\pi}$, then $\hat{B} \leq z^*$, and so $\hat{B}$ satisfies (3.9). Also $\bar{x}$ satisfies (3.8a) and $\bar{x} > 0$. Finally, we need to show that $(-\beta\xi + c)^T\bar{x} + \hat{t} = \hat{B}$, but this follows easily since $(-\beta\xi + c)^T\bar{x} + \bar{t} = \bar{B}$ and $\hat{t} = \bar{t} + \hat{B} - \bar{B}$, and so (3.8b) is satisfied. Also, since $\hat{B} > B$ (4.8), $\hat{t} > \bar{t} > 0$, and so (3.8c) is satisfied. Thus $(\bar{x}, \hat{t}, \hat{B})$ is feasible in PR.

To demonstrate the decrease in the potential function, we note

$$\begin{aligned} F(\bar{x},\hat{t}) - F(\bar{x},\bar{t}) &= -\ell n(\hat{t}) + \ell n(\bar{t}) = -\ell n(\hat{t}/\bar{t}) \\ &= -\ell n((\bar{t} + \hat{B} - \bar{B})/\bar{t}) \\ &= -\ell n(1 + (\hat{B} - \bar{B})/\bar{t}) \\ &\leq -\ell n\left(1 + \frac{\sqrt{n+1}(1-\bar{\gamma})}{(1+\bar{\gamma})}\right) \quad \text{(from (4.8))} \\ &\leq -\frac{1}{6}, \end{aligned}$$

because $\bar{\gamma} \leq \frac{4}{5}$ and $n \geq 3$. □

The following algorithm is a summary of the analysis of this section.

ALGORITHM 1 $(A, b, c, \xi, \beta, x^0, t^0, B^0, q, \gamma)$ $(q \geq n + 1 + \sqrt{n+1}, \gamma \leq 1)$.

Step 0. Initialization $k = 0$.

Step 1. Compute primal direction
$(\bar{x}, \bar{t}, \bar{B}) = (x^k, t^k, B^k)$.
Compute $(\tilde{d}, \tilde{r}, \tilde{\pi}, \tilde{\theta}, \tilde{\delta}, \tilde{s})$ from (4.2a)–(4.2h).
Compute $\bar{\gamma}$ from (4.3).

Step 2. Determine whether to take primal step or to update dual bound.
If $\bar{\gamma} \geq \gamma$, go to Step 3.
If $\bar{\gamma} < \gamma$, go to Step 4.

Step 3. Take a primal step.
Set $(\hat{x}, \hat{t}) = (\bar{x} - (\alpha/\bar{\gamma})\tilde{d}, \bar{t} - (\alpha/\bar{\gamma})\tilde{r})$, where $\alpha = 2/5$.
Set $\hat{B} = \bar{B}$.
Go to Step 5.

Step 4. Update dual bound.
Compute $(\hat{\pi}, \hat{\theta}, \hat{s})$ from (4.7).
Compute $\hat{B}$ from (4.8).
Compute $\hat{t}$ from (4.9).
Set $\hat{x} = \bar{x}$.
Go to Step 5.

Step 5. Redefine all variables and return.
$(x^{k+1}, t^{k+1}, B^{k+1}) = (\hat{x}, \hat{t}, \hat{B})$
$k \leftarrow k + 1$.
Go to Step 1.

With $q = n + 1 + \sqrt{n+1}$ and $\gamma = 4/5$, Theorems 4.1 and 4.2 guarantee a decrease in the $F(x,t)$ of at least $\delta = 1/6$ at each iteration of Algorithm 1, yielding the bounds on convergence as stated in Theorem 3.1.

Section 5 discusses ways to accelerate and improve Algorithm 1 and other features as well.

5. Modifications and enhancements to Algorithm 1.

5.1. Use of a linesearch of the potential function. Instead of using a fixed steplength of α in Step 3, α can be determined by a linesearch of the potential function $F(x,t)$. Todd and Burrell [14] have shown that $F(x,t)$ is quasiconvex, and so the linesearch procedure is very simple to execute.

A similar idea can be used to improve the bound $\hat{B}$ in Step 4 of Algorithm 1. Suppose $(\bar{\pi}, \bar{\theta}, \bar{s})$ was a previous solution to the dual *LD* resulting in the previous bound $\bar{B} = b^T\bar{\pi}$.

Then at Step 4, the new dual solution is $(\hat{\pi}, \hat{\theta}, \hat{s})$ with $\hat{B} = b^T\hat{\pi} > \bar{B}$, from (4.8). A min–ratio test can be used to compute the largest value α^* of α for which the affine combination $\alpha(\hat{\pi}, \hat{\theta}, \hat{s}) + (1-\alpha)(\bar{\pi}, \bar{\theta}, \bar{s})$ is feasible for LD, and since $\alpha^* > 1$, $B^* \triangleq b^T(\alpha^*\hat{\pi} + (1-\alpha^*)\bar{\pi}) > \bar{B}$. This new value B^* is a valid lower bound on z^*, and can be used instead of $\hat{B}$ at Step 4. A further enhancement on the choice of B is discussed next.

5.2. Update the lower bound B using Fraley's restriction of the dual. In Fraley [5], a two-dimensional restriction of the dual problem is developed. This restriction has been used to great advantage in Todd [17], for example. Here we motivate this problem and show its use in updating the lower bound B in Algorithm 1. Substituting $b^T = \bar{x}^T A^T = e^T\bar{X}A^T$ in LD (3.2) and multiplying the constraints by $\bar{X}$ yields the equivalent form of LD.

$$LD': \quad \max_{\pi,\theta,s} e^T\bar{X}A^T\pi \tag{5.1a}$$

$$\text{s.t.} \quad \bar{X}A^T\pi + \theta\bar{X}\xi + \bar{X}s = \bar{X}c \tag{5.1b}$$

$$s \geq 0. \tag{5.1c}$$

Note from (5.1b) that (5.1a) is equal to $c^T\bar{x} - \xi^T\bar{x}\theta - \bar{x}^T s$. Also, note that (5.1b) is equivalent to

$$\theta(\bar{X}\xi)_p + (\bar{X}s)_p = (\bar{X}c)_p,$$

where

$$P = [I - \bar{X}A^T(A\bar{X}^2A^T)^{-1}A\bar{X}], \tag{5.2}$$

and the notation v_p denotes the quantity Pv, i.e., $v_p = Pv$.

Thus LD is equivalent to

$$LD'': \quad \max_{\theta,s} c^T\bar{x} - \xi^T\bar{x}\theta - \bar{x}^T s \tag{5.3a}$$

$$\text{s.t.} \quad \theta(\bar{X}\xi)_p + (\bar{X}s)_p = (\bar{X}c)_p, \tag{5.3b}$$

$$s \geq 0. \tag{5.3c}$$

Now consider the equation system

$$\theta(\bar{X}\xi)_p + \bar{X}s + \mu(e - e_p) = (\bar{X}c)_p. \tag{5.4}$$

If (μ, θ, s) solves (5.4), then (θ, s) solves (5.3b), and so the following program is a restriction of LD'' in the sense that the set of feasible solutions (in θ and s) is a subset of those of LD''.

$$FD_{\bar{x}}: \quad z_{\bar{x}} = \max_{\mu,\theta,s} c^T\bar{x} - \xi^T\bar{x}\theta - \bar{x}^T s \tag{5.5a}$$

$$\text{s.t.} \quad \theta(\bar{X}\xi)_p + \bar{X}s + \mu(e - e_p) = (\bar{X}c)_p \tag{5.5b}$$

$$s \geq 0. \tag{5.5c}$$

We denote this linear program as $FD_{\bar{x}}$ for "Fraley's restricted dual" and note its dependence on $\bar{x}$ through (5.5) as well as (5.2). Note also that $FD_{\bar{x}}$ can be solved as a two-dimensional linear program in n inequalities in the variables θ and μ.

Now consider a modification of Algorithm 1 that solves $FD_{\bar{x}}$ at the start of Step 1, and replaces $(\bar{B}, \bar{t})$ by $(z_{\bar{x}}, \bar{t} + z_{\bar{x}} - \bar{B})$ whenever $FD_{\bar{x}}$ has an optimal solution and $z_{\bar{x}} \geq \bar{B}$. (If at some iteration $FD_{\bar{x}}$ is unbounded, i.e., $z_{\bar{x}} = +\infty$, then $z^* = +\infty$ and so LP has no feasible solution.)

We can show that if Fraley's restricted dual is used to update $\bar{B}$ at the start of Step 1, then it will always be the case that $\bar{\gamma} \geq 1$, and so Step 4 of the algorithm will never be encountered. To see this, suppose that Fraley's restricted dual is used to update $\bar{B}$ at the start of Step 1 as indicated above. Then $\bar{B} \geq z_{\bar{x}}$. Consider the following proposition.

PROPOSITION 5.1. *Let* $(\hat{\pi}, \hat{\theta}, \hat{s})$ *be the dual solution to LD at Step* 4 *of Algorithm* 1. *Then* $(\hat{\mu}, \hat{\theta}, \hat{s})$ *is feasible for* $FD_{\bar{x}}$, *where* $\hat{\mu} = -1/\tilde{\theta}$ *and* $\tilde{\theta}$ *is given in* (4.2), *with objective value* $\hat{B}$ (*see* (4.8)), *and* $\hat{B} > \bar{B}$.

Proof. Consider the system (4.2). $A\tilde{d} = 0$, so $A\bar{X}(\bar{X}^{-1}\tilde{d}) = 0$, so $(\bar{X}^{-1}\tilde{d})_p = \bar{X}^{-1}\tilde{d}$. From (4.2d), $\bar{X}^{-1}\tilde{d} = -e - \bar{X}A^T\tilde{\pi} - \left(\tilde{\theta}\beta - \tilde{\delta} - \frac{q}{\xi^T\bar{x}}\right)\bar{X}\xi + \tilde{\theta}\bar{X}c$, and $\bar{X}^{-1}\tilde{d} = (\bar{X}^{-1}\tilde{d})_p = -e_p - \left(\tilde{\theta}\beta - \tilde{\delta} - \frac{q}{\xi^T\bar{x}}\right)(\bar{X}\xi)_p + \tilde{\theta}(\bar{X}c)_p$. Thus from (4.2g) and (4.7),

$$\bar{X}\hat{s} = \frac{1}{\tilde{\theta}}\bar{X}\tilde{s} = \frac{1}{\tilde{\theta}}(e + \bar{X}^{-1}\tilde{d}) = \frac{1}{\tilde{\theta}}(e - e_p) - \left(\beta - \frac{\tilde{\delta}}{\tilde{\theta}} - \frac{q}{\xi^T\bar{x}\tilde{\theta}}\right)(\bar{X}\xi)_p + (\bar{X}c)_p$$
$$= -\hat{\mu}(e - e_p) - \hat{\theta}(\bar{X}\xi)_p + (\bar{X}c)_p,$$

and so $(\hat{\mu}, \hat{\theta}, \hat{s})$ is feasible for $FD_{\bar{x}}$, with objective value $c^T\bar{x} - \hat{\theta}(\xi^T\bar{x}) - \bar{x}^T\hat{s} = b^T\hat{\pi} = \hat{B} > \bar{B}$. □

Now from the Proposition 5.1, if $\bar{\gamma} < 1$, then we would produce a solution to $FD_{\bar{x}}$ with objective $\hat{B} > \bar{B} \geq z_{\bar{x}}$, a contradiction. Thus, if $\bar{B}$ is updated using Fraley's restricted dual, Step 4 of Algorithm 1 will never be encountered.

5.3. Check for finite termination of Phase I. Suppose Algorithm 1 is at Step 3. Instead of setting $\alpha = 2/5$ or determining α by a linesearch of the potential function, one could first test if $(\bar{x} - \alpha\tilde{d})$ solves the Phase I problem for some value of α. Since $A\bar{x} = b$, $\xi^T\bar{x} > 0$, $\xi^T\tilde{d} \geq 0$, $\bar{x} > 0$, this amounts to checking if

$$\bar{x} - \left(\frac{\xi^T\bar{x}}{\xi^T\tilde{d}}\right)\tilde{d} \geq 0$$

in the case when $\xi^T\tilde{d} > 0$. If indeed

$$x' = \bar{x} - \frac{\xi^T\bar{x}}{\xi^T\tilde{d}}\tilde{d} \geq 0,$$

then x' solves the Phase I problem, and then LP can be solved from the feasible point x' by a purely Phase II method, of which many abound. (Of course, the new point x' generated from the above test may have a very poor objective value, in which case this approach may not be advantageous. However, in preliminary computational experiments, we found that finite termination of Phase I via this test has worked very well on small randomly generated problems.)

5.4. A ζ-optimal algorithm for LP. Suppose, instead of solving LP, we are interested in finding a feasible solution $\bar{x}$ to LP whose objective value is within a value $\zeta > 0$ of z^*, i.e., $c^T\bar{x} \leq z^* + \zeta$. (One can easily imagine a variety of situations where this is a reasonable

goal, such as when the objective function is not well specified.) Thus we seek a point $\bar{x}$ that satisfies

$$\begin{aligned} A\bar{x} &= b, \\ \xi^T \bar{x} &= 0, \\ \bar{x} &\geq 0, \\ c^T \bar{x} &\leq z^* + \zeta. \end{aligned}$$

Algorithm 1 can be easily modified to accomplish this goal, as follows: Whenever the bound $\bar{B}$ is updated to $\hat{B}$ in Step 4, replace $\bar{B}$ by $\hat{B} + \zeta$, instead of $\hat{B}$. Then all dual bounds B^k will satisfy $B^k \leq z^* + \zeta$, and hence all points x^k will satisfy

$$(-\beta\xi + c)^T x^k \leq B^k \leq z^* + \zeta.$$

Rearranging gives

$$c^T x^k \leq (z^* + \zeta) + \beta \xi^T x^k.$$

Then Theorem 3.1(i) is still valid and so as $\xi^T x^k \to 0$,

$$\limsup_{k \to \infty} c^T x^k \leq z^* + \zeta.$$

Whenever $\bar{B}$ is updated to $\hat{B}$ in Step 4, $\bar{B}$ increases by at least $\zeta > 0$. Thus Step 4 can only be visited at most

$$\left\lceil \frac{z^* - B^0}{\zeta} \right\rceil$$

times. Furthermore, the fact that the bound $\bar{B}$ is increased by at least ζ should accelerate convergence of the algorithm.

5.5. An explicit convergence constant for Algorithm 1. Theorem 3.1, together with Lemma 3.1, states that all iterates (x^k, t^k, B^k) of Algorithm 1 must satisfy

$$(\xi^T x^k) \leq (\xi^T x^0) C_1 e^{-k/6q},$$

where C_1 is given in (3.11). Although it is easy to see that $C_1 \leq 2^L$ when z^* is finite and β, x^0, and t^0 have size $O(L)$, it is impractical to compute C_1. Knowing C_1 is nevertheless important from the point of view of a prior guarantee that $\xi^T x^k$ will be no larger than a certain value after a certain number of iterations. Below we show that if an *upper* bound $\bar{U}$ on z^* is known in advance, then C_1 can be replaced by a known value $\tilde{C}_1$ (derived below) whenever the algorithm visits Step 4.

LEMMA 5.1. (Computing a substitute value of C_1). *Suppose Algorithm* 1 *is in Step* 4 *at iteration k. Then let*

$$\tilde{D} = \frac{1}{1 - \bar{\gamma}} \left[n + 1 - \bar{\gamma}^2 + \frac{(1 + \bar{\gamma})}{\bar{t}} [\bar{U} - \bar{B}] + \frac{q}{\xi^T \bar{x}} \xi^T \tilde{d} \right]$$

and

$$\tilde{C}_1 = [(n+1)^{-1} \tilde{D}]^{\left(\frac{n+1}{q}\right)}.$$

Then

(i) *for all subsequent iterations* $i > k$,

$$\xi^T x^i \leq (\xi^T x^k)\tilde{C}_1 e^{-\left(\frac{i-k}{6q}\right)}$$

and

(ii) *if* $\gamma = \frac{4}{5}$, *then* $\bar{\gamma} < \frac{4}{5}$ *and* $\tilde{D} \leq 5[q + \frac{2}{\bar{t}}(\bar{U} - \bar{B})]$.

Proof. Suppose at iteration k that the algorithm is in Step 4. Then $\bar{\gamma} < \gamma \leq 1$. Consider the linear program

$$\begin{aligned} \overline{PB}: \qquad \bar{T} = \max_{x,t} \; & e^T \bar{X}^{-1} x + \bar{t}^{-1} t \\ \text{s.t.} \quad & Ax = b, \\ & (-\beta\xi + c)^T x + t \leq \bar{U}, \\ & \xi^T x \leq \xi^T \bar{x}, \\ & -\xi^T x \leq 0. \end{aligned}$$

Note that since $\bar{U} \geq z^* \geq B^i$, $i = k, \ldots$, then all iterates (x^i, t^i) will be feasible for $\overline{PB}$ for $i \geq k$. The dual of $\overline{PB}$ is

$$\overline{DB}: \qquad \bar{T} = \min_{\lambda,\theta,\delta,\mu} \; b^T\lambda + \bar{U}\theta + \xi^T\bar{x}\delta \tag{5.6a}$$

$$\text{s.t.} \quad A^T\lambda + \theta(c - \beta\xi) + (\delta - \mu)\xi \geq \bar{X}^{-1}e, \tag{5.6b}$$

$$\theta \geq \bar{t}^{-1}, \tag{5.6c}$$

$$\theta, \delta, \mu \geq 0. \tag{5.6d}$$

Suppose we are at Step 4 of the algorithm, and so $\bar{\gamma} < \gamma \leq 1$. Then $\tilde{s} > 0$, and upon setting

$$(\lambda', \theta', \delta', \mu') = \left(\frac{-\tilde{\pi}}{1-\bar{\gamma}}, \frac{\tilde{\theta}}{1-\bar{\gamma}}, \frac{q}{\xi^T\bar{x}(1-\bar{\gamma})} + \frac{\tilde{\delta}}{(1-\bar{\gamma})}, 0\right),$$

it can be verified by rearranging (4.2d) that $(\lambda', \theta', \delta', \mu')$ is feasible for $\overline{DB}$. To see this, note from (4.3) that $\bar{X}^{-1}\tilde{d} \geq -\bar{\gamma}e$ and $\bar{t}^{-1}\tilde{r} \geq -\bar{\gamma}$, and from (4.2d) that $A^T(-\tilde{\pi}) + \tilde{\theta}(-\beta\xi + c) + \left(\tilde{\delta} + \frac{q}{\xi^T\bar{x}}\right)\xi = \bar{X}^{-1}(e + \bar{X}^{-1}d) \geq \bar{X}^{-1}e(1 - \bar{\gamma})$. Dividing through by $(1 - \bar{\gamma})$ shows that $(\lambda', \theta', \delta', \mu')$ solves (5.6b). Also, from (4.2h), $\theta' = \frac{\tilde{\theta}}{1-\bar{\gamma}} = \frac{1+\bar{t}^{-1}\tilde{r}}{\bar{t}}\left(\frac{1}{1-\bar{\gamma}}\right) \geq \frac{1-\bar{\gamma}}{1-\bar{\gamma}}\left(\frac{1}{\bar{t}}\right) = \bar{t}^{-1}$, so (5.6c) is satisfied. Finally, note from (4.2h) and (4.3) that θ', δ', $\mu' \geq 0$, so (5.6d) is satisfied. Then $\bar{T} \leq b^T\lambda' + \bar{U}\theta' + \delta'\xi^T\bar{x}$

$$\begin{aligned} &= \frac{1}{1-\bar{\gamma}}\left[-\bar{x}^t A^T\tilde{\pi} + \bar{U}\tilde{\theta} + q + \tilde{\delta}\xi^T\bar{x}\right] \\ &= \frac{1}{1-\bar{\gamma}}\left(-(\bar{B} - \bar{t})\tilde{\theta} - \tilde{\delta}\xi^T\bar{x} - q + n + e^T\bar{X}^{-1}\tilde{d} + \bar{U}\tilde{\theta} + q + \tilde{\delta}\xi^T\bar{x}\right) \quad \text{(from (4.2d))} \\ &= \frac{1}{1-\bar{\gamma}}\left(n + \bar{t}\tilde{\theta} + (\bar{U} - \bar{B})\tilde{\theta} + e^T\bar{X}^{-1}\tilde{d}\right) \\ &= \frac{1}{1-\bar{\gamma}}\left(n + 1 + \bar{t}^{-1}\tilde{r} + (\bar{U} - \bar{B})\tilde{\theta} + e^T\bar{X}^{-1}\tilde{d}\right) \quad \text{(from (4.2h))} \\ &= \frac{1}{1-\bar{\gamma}}\left(n + 1 + (\bar{U} - \bar{B})\tilde{\theta} + \frac{q}{\xi^T\bar{x}}\xi^T\tilde{d} - \bar{\gamma}^2\right) \quad \text{(from (4.5))} \\ &\leq \tilde{D}, \end{aligned}$$

since $\tilde{\theta} \leq (1+\bar{\gamma})/\bar{t}$ from (4.2h). Finally, for any $i > k$, (x^i, t^i) is feasible in $\overline{PB}$, so that

$$e^T \bar{X}^{-1} x^i + \bar{t}^{-1} t^i \leq \bar{T} \leq \tilde{D}.$$

It then follows from the arithmetic–geometric mean inequality that

$$\sum_{j=1}^{n} \ell n x_j^i + \ell n t^i - \sum \ell n \bar{x}_j - \ell n \bar{t} \leq (n+1)\ell n(\tilde{D}/(n+1)).$$

The proof of (i) then follows as in Lemma 3.1 and Theorem 3.1(i). To see (ii), note from (4.5) that

$$\begin{aligned} \frac{q}{\xi^T \bar{x}} \xi^T \tilde{d} &= \bar{\gamma}^2 + e^T \bar{X}^{-1} \tilde{d} + \bar{t}^{-1} \tilde{r} \\ &\leq \bar{\gamma}^2 + \sqrt{n+1}\,\bar{\gamma} \leq \bar{\gamma}^2 + \sqrt{n+1}. \end{aligned}$$

Then with $\bar{\gamma} < \gamma = \frac{4}{5}$,

$$\tilde{D} \leq 5\left[q + \frac{2}{\bar{t}}(\bar{U} - \bar{B})\right]. \qquad \square$$

Appendix A. PROPOSITION A.1. *If* $x > -1$, $\ell n(1+x) \leq x$.
PROPOSITION A.2. *If* $|x| \leq \alpha < 1$, *then* $\ell n(1+x) \geq x - \frac{x^2}{2(1-\alpha)}$.
Proofs of the above two inequalities can be found in [6].
PROPOSITION A.3. *Consider the dual linear programs*

$$\begin{array}{lll} LP_r\colon \quad z^*(r) = \min_x c^T x & & LD_r\colon \quad \max_{\pi,\theta} b^T\pi + r\theta \\ \text{s.t.} \quad Ax = b, & & \text{s.t.} \quad A^T\pi + \xi\theta \leq c. \\ \quad\ \xi^T x = r, & & \\ \quad\ x \geq 0. & & \end{array}$$

Suppose (π^*, θ^*) *solves* LD_0, *and let* $z^* = z^*(0)$. *Then for any* x *feasible for* LP_r, $c^T x \geq z^* + \theta^* r$.

Proof. Because (π^*, θ^*) is feasible for the dual LD_r for any r,

$$z^*(r) \geq b^T\pi^* + r\theta^* = z^*(0) + r\theta^* = z^* + r\theta^*.$$

Therefore, if x is feasible for LP_r, $c^T x \geq z^*(r) \geq z^* + r\theta^*$. $\square$

Acknowledgment. I would like to acknowledge Michael Todd for many stimulating discussions regarding the warm start LP problem that had an influence on the research herein.

REFERENCES

[1] K. M. ANSTREICHER, *A combined phase* I–*phase* II *projective algorithm for linear programming*, Math. Programming, 43 (1989), pp. 209–223.
[2] ———, *A Combined Phase* I–*Phase* II *Scaled Potential Algorithm for Linear Programming*, CORE Discussion Paper 8939, CORE Catholic University of Louvain, Belgium, 1989.
[3] ———, *On monotonicity in the scaled potential algorithm for linear programming*, Linear Algebra Appl., 152 (1991), pp. 223–232.
[4] ———, *On interior algorithms for linear programming with no regularity assumptions*, Oper. Res. Lett., 11 (1992), pp. 209–212.

[5] C. FRALEY, *Linear updates for a single-phase projective method*, Oper. Res. Lett., 9 (1990), pp. 169–174.
[6] R. M. FREUND, *Polynomial-time algorithms for linear programming based only on primal scaling and projected gradients of a potential function*, Math. Programming, 51 (1991), pp. 203–222.
[7] ———, *Theoretical efficiency of a shifted barrier function algorithm for linear programming*, Linear Algebra Appl., 152 (1991), pp. 19–41.
[8] ———, *A Potential-Function Reduction Algorithm for Solving a Linear Program Directly from an Infeasible "Warm Start,"* Working Paper 3079-89-MS, Sloan School of Management, Massachusetts Institute of Technology, September 1989; Math. Programming, to appear.
[9] G. DE GHELLINCK AND J.-P. VIAL, *A polynomial Newton method for linear programming*, Algorithmica, 1 (1986), pp. 425–453.
[10] P. E. GILL, W. MURRAY, M. A. SAUNDERS, AND M. H. WRIGHT, *Shifted Barrier Methods for Linear Programming*, Tech. Report SOL 88-9, Department of Operations Research, Stanford University, Stanford, CA, 1988.
[11] C. C. GONZAGA, *Polynomial affine algorithms for linear programming*, Math. Programming, 49 (1990), pp. 7–21.
[12] N. KARMARKAR, *A new polynomial time algorithm for linear programming*, Combinatorica, 44 (1984), pp. 373–395.
[13] R. A. POLYAK, *Modified Barrier Functions*, Report RC 14602, IBM T. J. Watson Research Center, Yorktown Heights, NY, 1989.
[14] M. J. TODD AND B. BURRELL, *An extension of Karmarkar's algorithm for linear programming using dual variables*, Algorithmica, 1 (1986), pp. 409–424.
[15] M. J. TODD, *On Anstreicher's Combined Phase* I–*Phase* II *Projective Algorithm for Linear Programming*, Tech. Report No. 776, School of OR&IE, Cornell University, Ithaca, NY, 1988; Math. Programming, to appear.
[16] M. J. TODD AND Y. WANG, *On Combined Phase* 1–*Phase* 2 *Projective Methods for Linear Programming*, Tech. Report No. 877, School of OR&IE, Cornell University, Ithaca, NY, 1989; Algorithmica, to appear.
[17] M. J. TODD, *Combining phase* I *and phase* II *in a potential reduction algorithm for linear programming*, Math. Programming, 59 (1993), pp. 133–150.
[18] J. P. VIAL, *A projective algorithm for linear programming with no regularity condition*, Oper. Res. Lett., 12 (1992), pp. 1–2.
[19] M. H. WAGNER, *Supply-demand decomposition of the national coal model*, Oper. Res., 29 (1981), pp. 1137–1153.
[20] Y. YE, *An* $O(n^3 L)$ *potential reduction algorithm for linear programming*, Math. Programming, 50 (1991), pp. 239–258.

SIAM J. OPTIMIZATION
Vol. 5, No. 2, pp. 269–285, May 1995

004

AN IMPLICIT FILTERING ALGORITHM FOR OPTIMIZATION OF FUNCTIONS WITH MANY LOCAL MINIMA *

P. GILMORE† AND C. T. KELLEY‡

Abstract. In this paper we describe and analyze an algorithm for certain box constrained optimization problems that may have several local minima. A paradigm for these problems is one in which the function to be minimized is the sum of a simple function, such as a convex quadratic, and high frequency, low amplitude terms that cause local minima away from the global minimum of the simple function. Our method is gradient based and therefore the performance can be improved by use of quasi-Newton methods.

Key words. filtering, projected gradient algorithm, quasi-Newton method

AMS subject classifications. 65H10, 65K05, 65K10

1. Introduction.

In this paper we describe and analyze an algorithm for bound constrained optimization problems that may have several local minima. The type of problem we have in mind is one in which the function to be minimized is the sum of a simple function, such as a convex quadratic, and high frequency, low amplitude terms that cause the local minima. Of particular interest is the case in which the amplitude of the high frequency components decays to zero near the local minima of the simple function. This algorithm, at various stages of its development, has been applied to such problems by a group in the Departments of Mathematics and Electrical and Computer Engineering at North Carolina State University to a variety of optimization problems that arise in computer-aided design of microwave devices [17], [16], [19], [20]. The algorithm is an extension of the projected gradient method [1] and as such is simple to implement and its performance can be improved by application of quasi-Newton methods. The purpose of this theoretical paper is to analyze the convergence properties of the method. The algorithm discussed in this paper was designed for the specific applications described fully in [17], [16], [19], and [20]. These papers put the numerical properties of the algorithm in context.

An example of the type of problem we have in mind is plotted in Fig. 1.1, taken from [20], which is a graph of a negative of the power-added efficiency of a simulated semiconductor device against the real and imaginary parts of the second harmonic of load impedance, which are constrained to lie in the interval $[0, 80]$. The small amplitude, high frequency perturbation that dies off near the optimal point, $(0, 0)$, is clearly visible.

In this section we begin by discussing in general terms the class of problems we seek to solve, then mention some other possibilities, and give a brief description of our approach. We formally describe the basic form of our algorithm in § 2. In § 3 we relate the output of the algorithm to a class of problems like that represented in

* Received by the editors January 13, 1993; accepted for publication (in revised form) September 2, 1993.

† National High Magnetic Field Laboratory, Florida State University, Tallahassee, Florida 32306-3016.

‡ North Carolina State University, Center for Research in Scientific Computation and Department of Mathematics, Box 8205, Raleigh, North Carolina 27695-8205 (tim_kelleyncsu.edu). This research was supported by National Science Foundation grant DMS-9024622, U. S. Army grant DAAL0189K0906, and Air Force Office of Scientific Research grant AFOSR-FQ8671-9101094. Computing activity was partially supported by an allocation of time from the North Carolina Supercomputing Center.

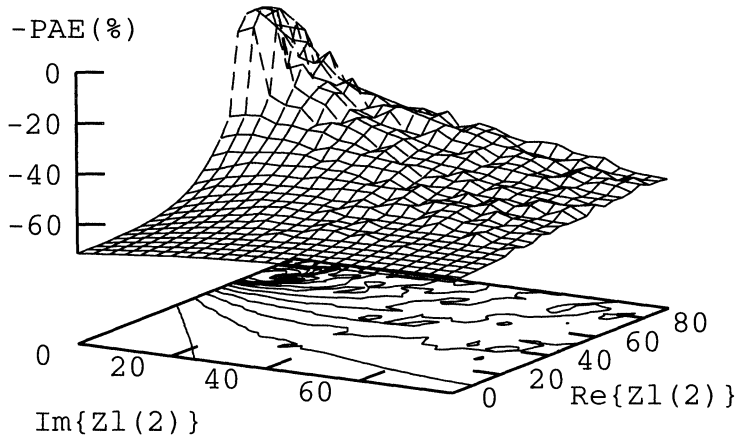


FIG. 1.1. *Power added efficiency.*

Fig. 1.1. In §§ 4 and 5 we state and prove some convergence results.

We seek to minimize a function f subject to simple bound constraints:

$$x \in \Omega = \{x \mid l^i \leq x^i \leq u^i\},$$

where x^i denotes the ith component of the vector $x \in R^N$, but that we can only observe $\hat{f} = f + \phi$, where ϕ is small in magnitude relative to f but has high frequency oscillations that cause local minima. We do not require that ϕ be smooth or even continuous.

One way to avoid such local minima is to filter the high frequency components from some expansion of $\hat{f}$, by means of a discrete Fourier transform, for example. In this way one might expect that the filtered form of $\hat{f}$ is a good approximation to f and does not have as many local minima as $\hat{f}$ does. By applying a conventional minimization algorithm to the filtered form of $\hat{f}$, one might find the minimizer of f up to the accuracy allowed by the noise in the observation. By changing the filter as an iteration progresses, to admit higher frequencies near the minimizer, or by restarting the iteration with a filter that admits higher frequencies after convergence, one might hope to even avoid local minima in f itself. Another advantage of refining the filter as the iteration progresses is to deal with problems, such as the one represented in Fig. 1.1, for which the perturbation ϕ is much smaller near the minimizer of f than elsewhere. Such problems were encountered in [17], [16], [19], and [20] in which noise from model errors was reduced near the solution of the optimization problem.

The disadvantage of applying a filter to $\hat{f}$ is that one must sample the entire variable space to use the filter. An example of such an algorithm has been reported recently in [12]. From the point of view of this paper, but not from that of [12], an advantage of such an algorithm is that large amplitude high frequency terms may be eliminated and therefore the iterates may avoid steep valleys. In the work reported in [17], [16], [19], and [20], steep valleys in the objective could be attributed to errors.

Filtering algorithms might not be appropriate in situations where steep valleys are significant.

Another approach is stochastic smoothing [13], where f is replaced by an average and the averages are changed as the iteration progresses. Almost sure convergence to the global minimum is proved in [13]. We do not prove such a strong result for the algorithm we propose. The advantages that our algorithm offers are simplicity and efficiency of implementation in that there is no preprocessing of the objective function and the analysis is completely deterministic. The price paid for this efficiency and simplicity is that some features of the objective, such as steep valleys, may be missed. This is reflected in the convergence results, which, in broad terms, assert that a local minimum of the unperturbed function will be identified up to the accuracy permitted by the perturbation ϕ. However, for the types of problems considered in [17], [16], [19], and [20], the decay of ϕ near the minimum allowed for high accuracy. We quantify this in Theorem 3.1.

We should also mention the multidirectional search algorithm for unconstrained problems proposed in [18]. This algorithm is based on the Nelder–Mead simplex algorithm. The multidirectional search algorithm uses a simplex that is rotated, expanded, and contracted as the algorithm proceeds. The size of this simplex corresponds to the length of the scales used in the implicit filtering algorithm. At the beginning of the optimization process the initial simplex is taken to be relatively large. This could, in principal, allow the multidirectional search algorithm to avoid local minima caused by the low amplitude high frequency term, although this was not the purpose of its design and is not covered by its analysis. As the algorithm proceeds the size of the simplex decreases allowing the structure of the merit function to be resolved to a finer level of resolution. Torczon proved convergence of this algorithm to local minima for continuous functions in [18]. The cost of application of the algorithm in [18] is roughly the same as ours if centered differences are used to approximate gradients. In [9], Dennis and Torczon showed how the multidirectional search algorithm could be efficiently implemented on parallel processor computers.

Rather than sample the variable space as a true filtering algorithm would, we propose the use of a finite-difference gradient-based method, for example the projected gradient method [1], with the step size in the difference chosen as it would be if ϕ were floating point round-off; for example, in the case of forward differences $h \approx \sqrt{\|\phi\|_\infty}$. Since $\|\phi\|_\infty$ is not known, we apply the finite-difference gradient-based method, decrease h after convergence, and apply the finite-difference gradient-based method again. We could terminate this outer iteration after a predetermined smallest value of h is reached or if we determine that no further progress is being made. We refer to the algorithm as *implicit filtering* because we use the differencing to "step over" the noise ϕ at varying levels of resolution, hence implicitly filtering the objective. This algorithm is, therefore, deterministic in both its implementation and its analysis.

A significant difference from the alternative approaches listed above is that the performance in the terminal phase of the iteration can be accelerated by a quasi-Newton method. This was very important from the point of view of the applications discussed in [17], [16], [19], and [20]. In those papers the SR1 update was found to be very useful and roughly 10% more efficient than the Broyden-Fletcher, Goldfarb, Shanno (BFGS) update. In § 4 we present some simple examples using the secant update for one-dimensional problems that illustrate this point.

2. Specification of the algorithm. As a basic algorithm we use the projected-gradient-Armijo algorithm from [1]. The form we use employs finite difference gradients. We let ∇_h be some finite difference gradient using, for example, forward or centered differences, computed with step h. The algorithm takes as its input the function to be minimized, an initial iterate x that is overwritten with an approximation to the minimizer, a step size h for the finite difference, a function $\tau(h)$ used for termination, a minimum step size $\bar{\alpha}$ for the line search, and a (small) parameter σ used to measure sufficient decrease. In the specification of Algorithm `projgrad` we use the notation

$$x(\alpha, h, f) = \mathcal{P}(x - \alpha \nabla_h f), \tag{2.1}$$

where $\mathcal{P}$ denotes the projection onto the feasible set

$$\mathcal{P}(x)^i = \begin{cases} u^i & x^i > u^i, \\ x^i & l^i \le x^i \le u^i, \\ l^i & x^i < l^i. \end{cases}$$

In the description of the algorithms that follow and in the discussion in §5, the Euclidean norm is denoted by $\|\cdot\|$ and the ℓ^∞ norm $\|\cdot\|_\infty$ with a subscript.

ALGORITHM 2.1. Algorithm `projgrad`$(f, x, h, \tau, \bar{\alpha}, \sigma)$

1. k = 0
2. Compute $\nabla_h f$.
   (a) If $\|x - x(1, h, f)\| \le \tau(h)$ terminate successfully.
3. Set $\alpha = 1$.
4. (a) If $\alpha < \bar{\alpha}$ terminate unsuccessfully.
   (b) Compute $f(x(\alpha, h, f))$.
   (c) If

$$f(x) - f(x(\alpha, h, f)) \ge \frac{\sigma \|x - x(\alpha, h, f)\|^2}{\alpha},$$

   set $x = x(\alpha, h, f), k = k + 1$ and go to step 2.
   (d) $\alpha = \beta\alpha$.

The roles of the parameters σ and β are the same as in standard discussions of the Armijo rule [1], [8], [14]. The role of $\bar{\alpha}$ is that of a safeguard to keep from reducing the step size too often when $-\nabla_h f$ is not a descent direction and to determine when no further reduction in h should be done. We have used $\bar{\alpha} = \beta^{10}$ in the application work reported in [17], [16], [19], and [20]. The analysis in § 5 can be directly extended to more general line search rules, such as the polynomial models in [8]. The cubic model was used in [17], [16], [19], and [20]. Extension to a trust region approach such as that in [5] or [15] should also be possible.

The basic form of the algorithm we propose in this paper requires a decreasing, finite sequence of difference steps $\{h_i\}_{i=1}^m$ called *scales* and consists of the repeated application of Algorithm `projgrad` to $\hat{f}$.

ALGORITHM 2.2. Algorithm `imfilter`$(\hat{f}, x, \{h_i\}, \tau, \bar{\alpha}, \sigma)$

- for $i = 1, \ldots m$
  call `projgrad`$(\hat{f}, x, h_i, \tau, \bar{\alpha}, \sigma)$.

Passing through the decreasing sequence of scales is intended to have the effect of changing the filter as the iteration progresses to admit higher frequencies in $\hat{f}$. However, this is only a heuristic, as we do no actual filtering. Our strategy for selection of the sequence $\{h_i\}$ is also a heuristic. We make some more remarks on that later in this section. Note that h_i, like the temperature in an annealing algorithm, the simplex size in multidirectional search, or the time step in the algorithm in [12], in a sense measures the resolution of the optimizer. The important difference in Algorithm `imfilter` from these algorithms is that it is gradient based. Therefore it has the simultaneous advantages that it does not require sampling of the variable space, can be analyzed with Taylor series methods, is simple to implement, and can be accelerated by quasi-Newton methods.

At this point we make a few remarks on the goals and properties of Algorithm `imfilter`. As is the case with other filtering algorithms, such as that given in [12], our algorithm may miss global minima caused by large amplitude "spikes" in ϕ. We view this as desirable and do not think of our algorithm as a global optimization method, but rather as a method for dealing with a particular class of noisy functions. In the work reported in [17], [16], [19], and [20], spikes represented error and were best avoided.

A proper criterion for determining if Algorithm `imfilter` has succeeded is also a question. Even if each call to `projgrad` terminates successfully with a solution x, there is no guarantee that a second call to `imfilter` would leave x invariant since ϕ could change the output from the early calls to `projgrad`. Therefore it might be necessary to restart `imfilter`. We show in § 3 that such restarts are not necessary if ϕ decays sufficiently rapidly near a global minimum of f. We set as our goal the computation of x such that x is a minimum at every scale.

DEFINITION 2.1. *x is a* minimum at all scales *if* `projgrad` *leaves x invariant for all $h = h_1, \ldots, h_m$.*

We compute a minimum at all scales by Algorithm `imfilter` and restarting, if necessary, until each call to `projgrad` leaves x invariant.

ALGORITHM 2.3. Algorithm `allscale`$(f, x, h, \tau, \bar{\alpha}, \sigma)$

- Until each call to `projgrad` leaves x invariant:
  call `imfilter`$(\hat{f}, x, \{h_i\}, \tau, \bar{\alpha}, \sigma)$.

The central theoretical contributions of this paper are to show how a minimum at all scales is related to a global minimum for functions of the type plotted in Fig. 1.1, where the perturbation decays near a minimum of f, and to give conditions on f, ϕ, τ, and $\bar{\alpha}$ under which Algorithm `allscale` terminates in finitely many steps and returns a minimum at all scales. We do these things in §§ 3, 4, and 5.

3. Minima at all scales: examples and characterization. The idea, first advocated in [17], [16], [19], and [20], that a minimum at all scales, and not a global minimum or a minimum of an explicitly filtered function, should be the goal of the iteration is central to the algorithm and reflects the motivating problems where the amplitude of ϕ decays near the minimum. We begin with a theorem that illustrates the relationship between a minimum at all scales and a global minimum. We follow that with two examples to illustrate the ideas. A feature of these theorems and their proofs is the variety of subtle relations between the size of the perturbation and the curvature of f and the parameters $\{h_i\}$, σ, $\bar{\alpha}$ in the specification of the algorithm. These relationships are further explored in §§ 4 and 5.

In the following theorem, as in [17], [16], [19], and [20], we use $\tau(h) = \bar{\tau}h$, for some $\bar{\tau} > 0$ and $h_k = \mu^k h_0$ for some $\mu \in (0,1)$. We assume that ∇_h is computed in Ω is a way that is at least first order accurate. We must make precise our assumptions on f and the way that ϕ decays near the global minimum of f.

ASSUMPTION 3.1. *Assume that $\hat{f} = f + \phi$, that f has a global minimum in Ω at x^*, ∇f is Lipschitz continuous in Ω with Lipschitz constant L. Assume that*

$$\Omega = \{x \,|\, l^i \le x^i \le u^i\} \subset R^N,$$

diameter $D_\Omega = \max(u^i - l^i) < \infty$. Assume that ∇_h is computed with forward, backward, or centered differences. Finally assume that there are c_0, c_1, e_-, and $M_\phi > 0$ such that for all $x \in \Omega$

$$\begin{aligned} &\|x - \mathcal{P}(x - \nabla f(x))\| \ge c_0 \|x - x^*\|, \\ &\|\nabla_h f - \nabla f(x)\| \le c_1 h, \\ &|\phi(x)| \le M_\phi \max\{\|x - x^*\|^2, e_-^2\}. \end{aligned} \tag{3.1}$$

THEOREM 3.1. *Assume that Assumption 3.1 holds. Then if M_ϕ is sufficiently small there are C_1, C_2, and $K \ge 1$, such that if*

$$h_0 \ge \frac{2C_1 M_\phi D_\Omega}{c_0 - C_2 M_\phi}, \tag{3.2}$$

$h_k = \mu^k h_0$, for some $\mu \in (0,1)$ and

$$\|x - x(1, h_k, \hat{f})\| = \|x - \mathcal{P}(x - \nabla_{h_k} \hat{f}(x))\| \le \bar{\tau} h_k$$

for $k = 1, \ldots, m$, then

$$\|x - x^*\| \le K h_-,$$

where $h_- = \max(e_-, h_m)$.

Conversely, there is $\bar{\tau}_0$ such that if $\bar{\tau} \ge \bar{\tau}_0$ and $h_m \ge e_- \ge \|x - x^*\|$ then x is a minimum at all scales.

Proof. We note that Assumption 3.1 implies that there are C_1 and C_2 such that if $\bar{\phi} = C_1 M_\phi$ and $\nu = C_2 M_\phi$ then either $\|x - x^*\| < e_-$ or

$$\|\nabla_h \phi(x)\| \le \bar{\phi} \|x - x^*\|^2 / h + \nu \max(e, h). \tag{3.3}$$

In fact, for forward or backward differences we have

$$\|\nabla_h \phi(x)\| \le 2 M_\phi \sqrt{N} (e + h)^2 / h,$$

leading to $C_1 = 2\sqrt{N}$ and $C_2 \le 6\sqrt{N}$. Using second order centered differences would reduce C_1 and C_2 by factors of two but not eliminate ν entirely. Perhaps because of this reduction of $\bar{\phi}$ and ν, centered differences were found to reduce the sensitivity of the algorithm on the size of ϕ in [17]. We have, for M_ϕ sufficiently small, that

$$\bar{\phi} \le \frac{c_0 - \nu}{4} \text{ and } 0 < \nu < c_0. \tag{3.4}$$

We let $e = \|x - x^*\|$. Our assumptions imply that either $e \leq e_-$ or for each $k = 0, \ldots, m$

$$\begin{aligned}
\bar{\tau} h_k &\geq \|x - \mathcal{P}(x - \nabla_h \hat{f}(x))\| \\
&\geq \|x - \mathcal{P}(x - \nabla_h f(x))\| - \|\nabla_h \phi(x)\| \\
&\geq \|x - \mathcal{P}(x - \nabla f(x)\| - \|\nabla f(x) - \nabla_h f(x)\| - \|\nabla_h \phi(x)\| \\
&\geq \|x - \mathcal{P}(x - \nabla f(x))\| - c_1 h_k - \left(\frac{\bar{\phi} e^2}{h_k} + \nu \max(e, h_k) \right) \\
&\geq c_0 e - c_1 h_k - \left(\frac{\bar{\phi} e^2}{h_k} + \nu \max(e, h_k) \right).
\end{aligned}$$

Therefore, either $e \leq e_-$ or for each $k = 0, \ldots, m$

$$c_0 e \leq \|x - \mathcal{P}(x - \nabla_{h_k} f(x))\| \leq (\bar{\tau} + c_1) h_k + \frac{\bar{\phi} e^2}{h_k} + \nu \max(e, h_k). \tag{3.5}$$

Setting $k = 0$ and using (3.2) implies that either $e \leq h_0$ or

$$(c_0 - \nu) e \leq (\bar{\tau} + c_1) h_k + \frac{(c_0 - \nu) e}{2}$$

and hence

$$e \leq \kappa h_0 = \mu^{-1} \kappa h_1, \tag{3.6}$$

where

$$\kappa = 2 \frac{\bar{\tau} + c_1}{c_0 - \nu}.$$

We now assume that M_ϕ is small enough so that

$$\frac{\bar{\phi} \max(\mu^{-1} \kappa, 2)}{c_0 - \nu} \leq \frac{1}{2}, \tag{3.7}$$

which is usually stronger (3.4).

We use (3.6) to induct on k and show that either $e \leq e_-$ or $e \leq \max(\kappa, 1) h_k$. Assume that $e \leq \kappa h_{k-1} \leq \mu^{-1} \kappa h_k$, which we have verified above for $k = 1$. The goal of the induction step is to show that either $e \leq e_-$, $e \leq h_k$ or $e \leq \kappa h_k$. Assume that $e > e_-$ and $e > h_k$, then we may apply (3.5) and (3.7) to conclude

$$e \leq \kappa h_k / 2 + \frac{\bar{\phi} e^2}{(c_0 - \nu) h_k} \leq \kappa h_k / 2 + \frac{\bar{\phi} \mu^{-1} \kappa}{c_0 - \nu} e \leq \kappa h_k / 2 + e/2.$$

Hence, either $e \leq e_-$, $e \leq h_k$ or $e \leq \kappa h_k$ and the induction is complete. This completes the proof of the forward part of the result with $K = \max(\kappa, 1)$.

To prove the converse, we note that if $h \geq e_- \geq e$ then

$$\|x - x(\alpha, h, \hat{f})\| \leq L e_- + c_1 h + \bar{\phi} e_- + \nu h \leq \bar{\tau} h$$

if $\bar{\tau} \geq \bar{\tau}_0 = L + c_1 + \bar{\phi} + \nu$. □

It is important in the proof of Theorem 3.1 that h_0 be large enough to begin the induction. Heuristically, an initial scale that is too small could lead to entrapment in a local minimum and that possibility is eliminated by (3.2). The same failure could be caused by selection of a value of $\bar{\tau}$ that is too small. The perturbation itself must be small, which is the role of assumption (3.7). Left unresolved is the issue of whether the outer iteration or the line search in `projgrad` will terminate. A deeper examination of the relation of the termination criteria, i.e., $\bar{\tau}$, and the line search, i.e., the parameter σ, to the size of ϕ is done in §§ 4 and 5.

The parameter $\bar{\tau}_0$ could be so large that many scales will be rejected; that is Algorithm `projgrad` will terminate on entry, before the iteration begins to make progress. The search for a good heuristic for the choice of $\bar{\tau}$ is an open problem.

Note that the accuracy of the finite difference plays no direct role in the ultimate accuracy of the iteration because of the presence of the $\nu \max(e, h)$ term in (3.3). To see how this term arises, assume that there is C such that

$$\phi(x) \leq C\|\phi\|_\infty \|x - x^*\|^2.$$

For forward differences we have

$$\|\nabla_h \phi(x)\| \leq 2C\sqrt{N}\|\phi\|_\infty (e + h)^2/h$$

leading to the estimates

$$\bar{\phi} \leq 2C\sqrt{N}\|\phi\|_\infty \quad \text{and} \quad \nu \leq 6C\sqrt{N}\|\phi\|_\infty.$$

Using second order centered differences would reduce the estimates of $\bar{\phi}$ and ν by factors of two but not eliminate ν entirely. Perhaps because of this reduction of $\bar{\phi}$ and ν, centered differences were found to reduce the sensitivity of the algorithm on the size of ϕ in [17], [16], [19], and [20].

As an example of the type of function we consider, we take $f = x^2$ and $\phi = \epsilon x^2 \cos(80x)$ on the interval $[-2, 2]$. Here $e_- = 0$. We take $\bar{\tau} = 2$, $\mu = 1/2$, and apply forward differencing. Using the notation in the proof we see that

$$c_0 = 2, \quad c_1 = 1, \quad \bar{\phi} = 2\epsilon, \quad \text{and} \quad \nu = 6\epsilon.$$

Hence $\kappa = 6/(2 - 6\epsilon)$ Hence (3.7) holds if

$$24\epsilon/(2 - 6\epsilon)^2 \leq 1/2,$$

which holds for any $\epsilon \leq 1/20$. As $D_\Omega = 4$, setting $\epsilon = 1/20$ and $h_0 = 2$ will satisfy the assumptions of Theorem 3.1. A plot of $\hat{f}$ with $\epsilon = 1/20$ is in Fig. 3.1.

In the applications it was rare that $e_- = 0$. Typically the noise in $\hat{f}$ decayed near the minimum, but not to zero. In this case the minimum could only be resolved to a level of size proportional to the square root of the minimum noise. As an example of such a function consider $\hat{f}_2 = x^2(x) + \phi_2(x)$ where $\Omega = [-2, 2]$ and

$$\phi_2(x) = .75x^2 \cos(80x)/20 + .25 * \cos(100x)^2/20.$$

$\hat{f}_2$ satisfies the assumptions of Theorem 3.1 with $e_- = .25/20$. We plot e_- in Fig. 3.2. The perturbations in these examples appear small, but clearly result in substantial

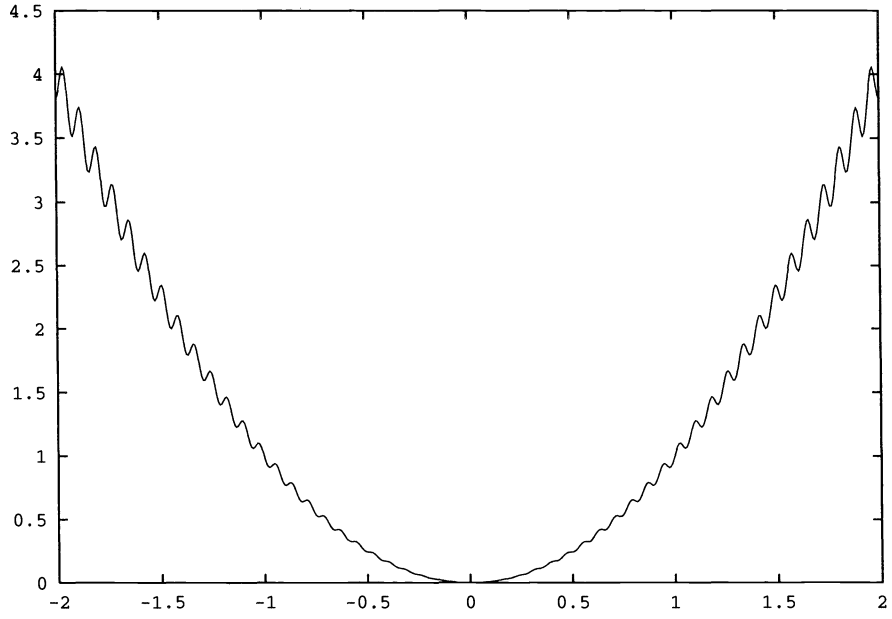


FIG. 3.1. $\hat{f} = f + \phi, e_- = 0.$

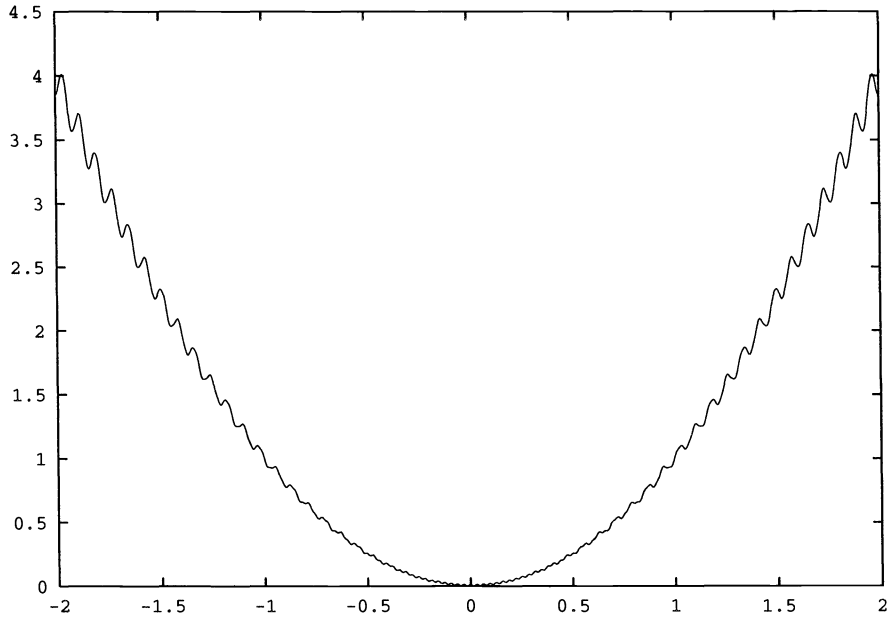


FIG. 3.2. $\hat{f}_2 = f + \phi_2, e_- \neq 0.$

local minima.

The proof of Theorem 3.1 also provides insight into how Algorithm 2.2 is intended to function. For functions satisfying the hypothesis of Theorem 3.1, application of `projgrad`$(\hat{f}, x, h_i, \tau, \bar{\alpha})$ return x_i with $\|x_i - x*\| \leq K \max(e_-, h_i)$. This should reduce the size of ϕ for the next iterate and allows for the reduction of h.

4. Convergence results. In this section we apply a technical result proved in § 5 to give conditions under which Algorithm `imfilter` will converge to a minimum at all scales for objectives that satisfy Assumption 3.1. We then give some simple examples to illustrate the behavior of the algorithm and show how quasi-Newton methods can improve the performance of the algorithm.

In § 5 we prove the following result.

THEOREM 4.1. *Let Assumption* 3.1 *hold and assume that* $M_\phi \leq 1$. *Let* $\beta \leq 1/2$ *and* $\sigma \in (0, 1/4)$ *be given. Let* $\tau(h) = \bar{\tau}h$. *Then there are* $\bar{\tau}_1$ *and* $\eta > 0$ *so that if* $\bar{\tau} \geq \bar{\tau}_1$, $M_\phi^{1/3}/\bar{\tau}^2 < \eta$, $\|x - x^*\| \geq e_-$, *and* $h \geq M_\phi^{1/3}\|x - x^*\|$, *then Algorithm* `projgrad`$(\hat{f}, x, h, \tau, \bar{\alpha}, \sigma)$ *will terminate successfully and return* x *such that* $\|x - x(1, h, \hat{f})\| \leq \bar{\tau}h$. *Moreover, either* x *is unchanged or* $\hat{f}$ *is reduced by at least* $\sigma\bar{\alpha}\bar{\tau}^2h^2$.

From the technical result Theorem 4.1 we can directly obtain conditions that insure that Algorithm `imfilter` will terminate successfully with a minimum at all scales. We consider sequences of scales of the form $h_k = \mu^k h_0$ with $\mu \in (0, 1)$.

THEOREM 4.2. *Assume that Assumption* 3.1 *holds, that* $M_\phi \leq 1$, *that* $M_\phi^{1/3}/\bar{\tau}^2 < \eta$, *and that* $\bar{\tau} > \bar{\tau}_0$. *Let* $\mu \in (0, 1)$ *be such that* $M_\phi^{1/3}K \leq \mu$. *Then if* $h_0 \geq D_\Omega/2$, $h_k = \mu^k h_0$, *and* $h_m \geq e_-$, *then Algorithm* `imfilter` *will terminate and return a minimum at all scales.*

Proof. Either $\|x - x^*\| \leq e_-$ and we terminate with a minimum at all scales by Theorem 3.1 or by Theorem 4.1, Algorithm `projgrad`$(\hat{f}, x, h_0, \tau, \bar{\alpha}, \sigma)$ will terminate with $\|x - x(1, h_0, \hat{f})\| \leq \bar{\tau}h_0$. We may then conclude from Theorem 3.1 that

$$e = \|x - x^*\| \leq Kh_0.$$

If we require that $M_\phi^{1/3}K \leq \mu$ then either $e < e_-$ or

$$\mu h_0 \geq M^{1/3}Kh_0 \geq M_\phi^{1/3}e,$$

and we may apply Algorithm `projgrad`$(\hat{f}, x, h, \tau, \bar{\alpha}, \sigma)$ with $h = \mu h_0$.

We may continue this, replacing h by μh as the iteration progresses. Eventually the iteration will terminate at a minimum at all scales since the reduction in h implies that there is a reduction in the bound for e, and any x such that $e \leq e_-$ is a minimum at all scales. □

Note that the restart feature of Algorithm `allscale` was not used in Theorem 4.2. It was not needed in the work reported in [17], [16], [19], and [20] either. A corollary of Theorem 4.2 is that if `projgrad` fails, as it can if there are too many step size reductions, then $e \leq e_-$. Hence monitoring the Armijo rule is a way to detect the limit for the scales.

COROLLARY 4.1. *Let the assumption of Theorem* 4.2 *hold. If Algorithm* `imfilter` *is initiated as in the statement of that result and* μh_m *is the first scale for which Algorithm* `projgrad` *fails, then* $Kh_m \geq e_-$.

While Corollary 4.1 does not guarantee that $Kh_m \geq e_- \geq K\mu h_m$, we have used it with success in the applications to determine when to terminate the iteration when no a priori knowledge of the size of ϕ was available.

We applied algorithm **imfilter** to the function

$$\hat{f}(x) = 2x^2 + x^2\cos(80x)/6$$

on the interval $[-2,2]$. We used $x_0 = -1.75$, $h_0 = 2$, and $h_i = 2^{-i}h_0$ for $i = 1,\ldots,12$. We used $\bar{\tau} = 2$. In Table 4.1 we tabulate h, the iteration counter i_p for Algorithm **projgrad** for that h, the number of step size reductions i_A at that iteration, x, $\nu(x) = |x - \mathcal{P}(x - \nabla_h\hat{f}(x))|$, and $\hat{f}(x)$. The last entry for each value of h corresponds to the terminal iterate for Algorithm **projgrad**. After evaluation of f at the initial iterate, the cost in function evaluations is like that for the projected gradient method. For each value of h and i_p one function evaluation is performed to compute $\nabla_h\hat{f}$, $f(x)$ being provided by the termination at the previous value of h or i_p i_A counts the number of additional function evaluations for each value of i_p required to obtain sufficient decrease in the Armijo rule. For Table 4.1 a total of 39 function evaluations are performed. An interesting feature of the table is how the iterates can remain

TABLE 4.1
One sweep of implicit filtering. $e_- = 0$.

h	i_p	i_A	x	$\nu(x)$	$f(x)$
2	0	0	-1.750e+0	2.947e+0	6.024e+0
1	0	1	-1.750e+0	3.750e+0	6.024e+0
1	1	2	7.442e-1	2.744e+0	1.016e+0
1	2	0	-5.564e-1	2.895e-1	6.637e-1
5.000e-1	0	1	-5.564e-1	1.315e+0	6.637e-1
5.000e-1	1	3	1.011e-1	1.336e+0	2.004e-2
5.000e-1	2	0	-6.595e-2	6.735e-1	9.087e-3
2.500e-1	0	0	-6.595e-2	2.221e-1	9.087e-3
1.250e-1	0	0	-6.595e-2	1.685e-2	9.087e-3
6.250e-2	0	1	-6.595e-2	1.450e-1	9.087e-3
6.250e-2	1	4	6.536e-3	1.603e-1	9.161e-5
6.250e-2	2	0	-3.480e-3	1.111e-1	2.616e-5
3.125e-2	0	0	-3.480e-3	4.603e-2	2.616e-5
1.562e-2	0	0	-3.480e-3	1.809e-2	2.616e-5
7.812e-3	0	0	-3.480e-3	1.834e-3	2.616e-5
3.906e-3	0	0	-3.480e-3	6.596e-3	2.616e-5
1.953e-3	0	1	-3.480e-3	1.081e-2	2.616e-5
1.953e-3	1	2	1.925e-3	1.252e-2	8.020e-6
1.953e-3	2	0	-1.204e-3	9.849e-4	3.140e-6
9.766e-4	0	1	-1.204e-3	3.100e-3	3.140e-6
9.766e-4	1	3	3.461e-4	3.614e-3	2.595e-7
9.766e-4	2	0	-1.056e-4	1.658e-3	2.418e-8

unchanged as the scale is reduced. It is normal behavior for algorithm **imfilter** to pass through more than one scale with the termination criterion satisfied on entry before finally taking a step and changing the iterate. The iteration terminated at a minimum at all scales and hence a full application of Algorithm **allscale** would terminate as well.

We also applied Algorithm **imfilter** to

$$\hat{f}(x) = 2x^2 + .75x^2\cos(80x)/6 + .25\cos(100x)/6.$$

On the interval $[-2,2]$, $x_0 = -1.75$, $h_0 = 2$, and $h_i = 2^{-i}h_0$ for $i = 1,\ldots,12$, and $\bar{\tau} = 2$. Here $e_- \approx .25/6$. Since $e_- \neq 0$ we might expect Algorithm **projgrad**, and

TABLE 4.2
One sweep of implicit filtering. $e_- \neq 0$.

h	i_p	i_A	x	$\nu(x)$	$f(x)$
2	0	0	-1.750e+0	2.948e+0	6.064e+0
1	0	1	-1.750e+0	3.750e+0	6.064e+0
1	1	2	7.354e-1	2.735e+0	1.041e+0
1	2	0	-5.934e-1	3.195e-1	6.996e-1
5.000e-1	0	1	-5.934e-1	1.281e+0	6.996e-1
5.000e-1	1	13	4.701e-2	1.268e+0	4.200e-3
5.000e-1	2	17	4.686e-2	1.267e+0	4.195e-3
5.000e-1	3	0	4.686e-2	1.267e+0	4.195e-3

TABLE 4.3
One sweep of secant method implicit filtering. $e_- = 0$.

h	i_p	i_A	x	$\nu(x)$	$f(x)$
2	0	0	-1.750e+0	2.947e+0	6.024e+0
1	0	1	-1.750e+0	3.750e+0	6.024e+0
1	1	0	7.442e-1	2.744e+0	1.016e+0
1	2	0	-5.291e-1	7.538e-2	5.559e-1
5.000e-1	0	1	-5.291e-1	1.109e+0	5.559e-1
5.000e-1	1	3	2.520e-2	1.065e+0	1.224e-3
5.000e-1	2	0	-8.757e-3	9.626e-1	1.631e-4
2.500e-1	0	0	-8.757e-3	4.999e-1	1.631e-4
1.250e-1	0	0	-8.757e-3	1.970e-1	1.631e-4
6.250e-2	0	0	-8.757e-3	8.672e-2	1.631e-4
3.125e-2	0	0	-8.757e-3	2.655e-2	1.631e-4
1.562e-2	0	0	-8.757e-3	3.974e-3	1.631e-4
7.812e-3	0	1	-8.757e-3	2.063e-2	1.631e-4
7.812e-3	1	1	1.560e-3	2.319e-2	5.272e-6
7.812e-3	2	0	-1.169e-3	1.173e-2	2.961e-6
3.906e-3	0	0	-1.169e-3	3.390e-3	2.961e-6
1.953e-3	0	0	-1.169e-3	8.344e-4	2.961e-6
9.766e-4	0	1	-1.169e-3	2.950e-3	2.961e-6
9.766e-4	1	1	3.056e-4	3.439e-3	2.024e-7
9.766e-4	2	0	-9.131e-5	1.720e-3	1.806e-8

hence Algorithm `imfilter`, to fail when $e < e_-$. This happens as we can see from the large number of step size reductions in the latter phases of the iteration reported in Table 4.2. The version of Algorithm `imfilter` used in the applications would have terminated the iteration at when $h = .5$ and $i_p = 2$ (since $\bar{\alpha} = 2^{-10}$), finding a minimum at all scales for the scales $\{h_i\}_{i=0}^{6}$.

The example in Table 4.2 illustrates the heuristic we use to estimate e_-, i.e., the point where further reductions in h give no advantage. When the Armijo rule in Algorithm `projgrad` fails, we conclude that the nature of the problem has changed and that a minimum scale has been found. While this heuristic is certainly far from a theorem, and examples can easily be constructed for which it fails, we found it to be very useful in the applications reported in [17], [16], [19], and [20].

We close this section with remarks on enhancements that improve the performance of Algorithm `allscale` in practice. First of all, for the work reported in [17], [16], [19], and [20] experience showed that it was not necessary to call `imfilter` more than once, hence the loop in Algorithm `allscale` that tests the invariance of `imfilter`, which is necessary for the theoretical results, was not used in the practical results reported in [17], [16], [19], and [20]. Convergence in the final phases of the iteration was improved

TABLE 4.4
One sweep of secant method implicit filtering. $e_- \neq 0$.

h	i_p	i_A	x	$\nu(x)$	$f(x)$
2	0	0	−1.750e+0	2.948e+0	6.064e+0
1	0	1	−1.750e+0	3.750e+0	6.064e+0
1	1	0	7.354e-1	2.735e+0	1.041e+0
1	2	0	−5.489e-1	2.269e−1	6.405e−1
5.000e−1	0	1	−5.489e−1	1.269e+0	6.405e−1
5.000e−1	1	1	8.569e−2	1.238e+0	3.338e−2
5.000e−1	2	0	−7.101e−2	6.495e−1	3.008e−2
2.500e−1	0	0	−7.101e−2	1.893e−1	3.008e−2

by using a projected SR1 [3] iteration such as that proposed in [6]. The SR1 update performed somewhat better than the BFGS update in the preliminary experiments for our work in [17], [16], [19], and [20]. This is consistent with other reports [7], [11], [10], and we used it in the computations reported in [17], [16], [19], and [20]. We point out that for problems of moderate size, a projected Newton formulation [2] would be equally desirable if the Hessian could be computed accurately and cheaply, but that was certainly not the case for the problems considered in [17], [16], [19], and [20] where function evaluations were quite expensive making Hessian evaluation too expensive.

To illustrate the benefits of incorporation of a quasi-Newton update into Algorithm **projgrad** we applied a secant update to the examples tabulated in Tables 4.1 and 4.2 and report the results in Tables 4.3 and 4.4. Twenty-eight function evaluations were required for the computation reported in Table 4.3, an improvement over the 39 for the results in Table 4.1. The conclusions from a comparison of Tables 4.2 and 4.4 are not so clear. While a substantial reduction in function evaluations is provided by the secant approach, the final value of the objective function is larger.

One could also use a different forward difference step for each component of $\nabla_h f$. Our implementation in [17], [16], [19], and [20] did this by scaling the feasible set to the unit cube in R^N. We let the largest scale $h_0 = .5$ be half the diameter of the cube and let $h_i = .5h_{i-1}$. Selection of the smallest scale is problem dependent and was determined in [17], [16], [19], and [20] by physical estimates of the error in the objective function.

5. Proof of Theorem 4.1. Throughout this section we assume that the assumptions of Theorem 4.1 hold. As in §1 $\nabla_h \hat{f}(x)$ represent a forward, backward, or centered difference approximation to $\nabla f(x)$. In this section, since the context will be clear, we abbreviate $x(\alpha, h, \hat{f})$ by $x(\alpha)$.

We require several lemmas.

LEMMA 5.1. *Let Assumption* 3.1 *hold. Let* $h \geq M_\phi^{1/3}\|x - x^*\|$. *Then there is* M_1 *such that*

$$\|\nabla_h \phi\| \leq M_1 M_\phi^{1/3} h. \tag{5.1}$$

Proof. As in the proof of Theorem 3.1,

$$\|\nabla_h \phi(x)\| \leq 2M_\phi \sqrt{N}(e + h)^2/h$$

and hence, for forward differences,

$$\|\nabla_h \phi(x)\| \leq 2M_\phi^{1/3}\sqrt{N}(M_\phi^{1/3} + 1)^2 h.$$

This completes the proof with $M_1 = 8\sqrt{N}$. □

LEMMA 5.2. *Let Assumption* 3.1 *hold. Let* $h \geq M_\phi^{1/3}\|x - x^*\|$. *Then there is* M_2 *such that*

$$\|\nabla_h \hat{f}(x) - \nabla f(x)\| \leq M_2 h. \tag{5.2}$$

Proof. Equation (5.2) is a direct consequence of Lemma 5.1 and Assumption 3.1 with $M_2 = c_1 + (c_0 + M_1)M_\phi^{1/3}$. □

The next lemma, which we give without proof, is a direct consequence of the projection theorem [4].

LEMMA 5.3. *Let* $x \in \Omega$, $0 < \alpha \leq 1$, *and* $x(\alpha) = x(\alpha, h, \hat{f})$. *Then*

1. $\|x - x(\alpha)\| \leq \alpha\|\nabla_h \hat{f}(x)\|$.
2. *For any* $i \in \{1, \ldots, n\}$, *if* $x^i(1) = x^i - \nabla_h \hat{f}(x)$ *then* $x^i(\alpha) = x^i - \alpha\nabla_h \hat{f}(x)$.
3. *If* $\|x - x(1)\| \geq \bar{\tau}h$ *then* $\|x - x(\alpha)\| \geq \alpha\bar{\tau}h$.
4. $\alpha\nabla_h \hat{f}(x)^T(x - x(\alpha)) \geq \|x - x(\alpha)\|^2$.

Next we prove a lemma that specifies an interval for the step size for which the criteria for sufficient decrease given in the description of Algorithm `projgrad` is always satisfied.

LEMMA 5.4. *Let Assumption* 3.1 *hold and assume that* $M_\phi \leq 1$. *Let* $\beta \leq 1/2$, $\bar{\alpha} \leq 1/(2L + 8)$, *and* $\sigma \in (0, 1/4)$ *be given. Let* $\tau(h) = \bar{\tau}h$. *Then there are* $\eta > 0$ *so that if* $M_\phi^{1/3}/\bar{\tau}^2 < \eta$, $\|x - x^*\| \geq e_-$, *and* $h \geq M_\phi^{1/3}\|x - x^*\|$, *then if* $\|x - x(1)\| \geq \bar{\tau}h$ *then the* generalized Armijo step size rule

$$\hat{f}(x) - \hat{f}(x(\alpha)) \geq \sigma\frac{\|x_k - x_k(\alpha)\|^2}{\alpha} \tag{5.3}$$

is satisfied for all α *with*

$$\frac{1}{2L + 8M_\phi} \leq \alpha \leq \frac{3}{2L + 8M_\phi}. \tag{5.4}$$

Hence there is m *such that* (5.3) *is satisfied with* $\alpha = \beta_m$.

Proof. Let $\delta(\alpha) = x - x(\alpha)$ and assume that $\bar{\tau} \geq \bar{\tau}_0$, so that the converse part of Theorem 3.1 holds. Therefore $e > e_-$.

Using the definition of $\hat{f}(x)$, and the fundamental theorem of calculus we obtain,

$$\begin{aligned}
\hat{f}(x) - \hat{f}(x(\alpha)) &= f(x) - f(x(\alpha)) + \phi(x) - \phi(x(\alpha)) \\
&= \delta(\alpha)^T \int_0^1 \nabla f(x - t\delta(\alpha))\, dt + \phi(x) - \phi(x(\alpha)) \\
&= \delta(\alpha)^T \left(\nabla f(x) + \int_0^1 (\nabla f(x - t\delta(\alpha)) - \nabla f(x))\, dt\right) \\
&\quad + \phi(x) - \phi(x(\alpha)) \qquad (5.5) \\
&\geq \delta(\alpha)^T \nabla f(x) - \frac{L\|(x - x(\alpha))\|^2}{2} + \phi(x) - \phi(x(\alpha).
\end{aligned}$$

Now assume that $\|x - x(1)\| \geq \bar{\tau}h$, and hence $e \geq e_-$. We estimate the parts of the right-hand side of (5.5) in turn. First, by Lemmas 5.2 and 5.3,

$$\delta(\alpha)^T \nabla f(x) \geq \frac{\|\delta(\alpha)\|^2}{\alpha} - \frac{M_2\|\delta(\alpha)\|^2}{\alpha\bar{\tau}} \geq \left(1 - \frac{M_2}{\bar{\tau}}\right)\frac{\|\delta(\alpha)\|^2}{\alpha}.$$

We set $\bar{\tau}_1 = \max(\bar{\tau}_0, 4M_2)$ and require $\bar{\tau} \geq \bar{\tau}_1$ to obtain

$$\delta(\alpha)^T \nabla f(x) \geq \frac{3\|\delta(\alpha)\|^2}{4\alpha}. \tag{5.6}$$

Using Assumption 3.1 we have, as we assume $e \geq e_-$,

$$|\phi(x(\alpha))| \leq M_\phi \|x(\alpha) - x^*\|^2 \leq 2M_\phi(\|\delta(\alpha)\|^2 + e^2).$$

By the estimates

$$M_\phi^{1/3} e \leq h \leq \|\delta(\alpha)\|/(\alpha\bar{\tau})$$

and Lemma 5.3, we have

$$|\phi(x(\alpha))| \leq 2M_\phi(\|\delta(\alpha)\|^2 + M_\phi^{-2/3}h^2)$$

$$\leq \frac{\|\delta(\alpha)\|^2}{\alpha}\left(2\alpha M_\phi + \frac{2M_\phi^{1/3}}{\alpha\bar{\tau}^2}\right).$$

Since

$$|\phi(x)| \leq M_\phi e^2 \leq M_\phi^{1/3}h^2 \leq \frac{\|\delta(\alpha)\|^2}{\alpha}\frac{M_\phi^{1/3}}{\alpha\bar{\tau}^2}.$$

We have, since $\alpha \leq 1$,

$$|\phi(x) - \phi(x(\alpha))| \leq \frac{\|\delta(\alpha)\|^2}{\alpha}\left(2M_\phi + \frac{3M_\phi^{1/3}}{\alpha\bar{\tau}^2}\right). \tag{5.7}$$

Using (5.6) and (5.7) in (5.5) yields

$$\hat{f}(x) - \hat{f}(x(\alpha)) \geq \frac{\|\delta(\alpha)\|^2}{\alpha}(3/4 - D_0/\alpha - D_2\alpha) \tag{5.8}$$

where

$$D_0 = \frac{3M_\phi^{1/3}}{\bar{\tau}^2} \quad \text{and} \quad D_2 = 2M_\phi + L/2.$$

If $M_\phi^{1/3}/\bar{\tau}^2$ is small enough so that

$$D_0 D_2 \leq 1/64, \tag{5.9}$$

then

$$-D_2\alpha^2 + \alpha/2 - D_0 \geq 0,$$

for all $\alpha \in [1/(4D_0), 3/(4D_0)]$. Hence $3/4 - D_0/\alpha - D_2\alpha \geq \sigma$ holds for all $\sigma \in (0, 1/4)$ and all $\alpha \in [1/(4D_0), 3/(4D_0)]$. This completes the proof. □

To complete the proof of Theorem 4.1 we note that Lemma 5.4 implies that if $\|x - x(1)\| > \bar{\tau}h$, then the line search in Algorithm `projgrad`$(f, x, h, \tau, \bar{\alpha}, \sigma)$ will return $x(\alpha)$ and, using Lemma 5.3,

$$\hat{f}(x) - \hat{f}(x(\alpha)) \geq \sigma \frac{\|x - x(\alpha)\|^2}{\alpha} \geq \sigma \frac{\alpha^2 \bar{\tau}^2 h^2}{\alpha} \geq \sigma \bar{\alpha} \bar{\tau}^2 h^2.$$

Boundedness of Ω and continuity of f then imply that Algorithm `projgrad` must terminate successfully.

Acknowledgments. The authors would like to thank Griff Bilbro, Bob Trew, and Tom Winslow of the Department of Electrical and Computer Engineering at North Carolina State University and Dan Stoneking of M/A-COM Corporation for their input to the development of the algorithm. We also appreciate Winslow's creation of Fig. 1.1.

REFERENCES

[1] D. B. BERTSEKAS, *On the Goldstein–Levitin–Polyak gradient projection method*, IEEE Trans. Autom. Control, 21 (1976), pp. 174–184.

[2] ———, *Projected Newton methods for optimization problems with simple constraints*, SIAM J. Control Optim., 20 (1982), pp. 221–246.

[3] C. G. BROYDEN, *Quasi-Newton methods and their application to function minimization*, Math. Comp., 21 (1967), pp. 368–381.

[4] P. CIARLET, *Introduction to numerical linear algebra and optimization*, Cambridge University Press, New York, NY, 1988.

[5] A. R. CONN, N. I. M. GOULD, AND P. L. TOINT, *Global convergence of a class of trust region algorithms for optimization problems with simple bounds*, SIAM J. Numer. Anal., 25 (1988), pp. 433–460.

[6] A. R. CONN, N. I. M. GOULD, AND P. L. TOINT, *Testing a class of methods for solving minimization problems with simple bounds on the variables*, Math. Comp., 50 (1988), pp. 399–430.

[7] ———, *Convergence of quasi-Newton matrices generated by the symmetric rank one update*, Math. Programming A, 50 (1991), pp. 177–195.

[8] J. E. DENNIS AND R. B. SCHNABEL, *Numerical Methods for Nonlinear Equations and Unconstrained Optimization*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[9] J. E. DENNIS AND V. TORCZON, *Direct search methods on parallel machines*, SIAM J. Optim., 1 (1991), pp. 448–474.

[10] C. T. KELLEY, E. W. SACHS, AND B. WATSON, *A pointwise quasi-Newton method for unconstrained optimal control problems*, II, J. Optim. Theory Appl., 71 (1991), pp. 535–547.

[11] H. KHALFAN, R. H. BYRD, AND R. B. SCHNABEL, *A theoretical and experimental study of the symmetric rank one update*, Tech. Report CU-CS-489-90, University of Colorado at Boulder, December 1990.

[12] J. KOSTROWICKI AND L. PIELA, *Diffusion equation method of global minimization: Performance for standard test functions*, J. Optim. Theory Appl., 69 (1991), pp. 269–284.

[13] J. KREIMER AND R. Y. RUBINSTEIN, *Smoothed functionals for optimization problems*, SIAM J. Numer. Anal., 25 (1988), pp. 470–487.

[14] J. ORTEGA AND W. RHEINBOLDT, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[15] T. STEIHAUG, *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., 20 (1983), pp. 626–637.

[16] D. STONEKING, G. BILBRO, R. TREW, P. GILMORE, AND C. T. KELLEY, *Yield optimization using a GaAs process simulator coupled to a physical device model*, in Proceedings IEEE/Cornell Conference on Advanced Concepts in High Speed Devices and Circuits, IEEE, 1991, pp. 374–383.

[17] ———, *Yield optimization using a GaAs process simulator coupled to a physical device model*, IEEE Transactions on Microwave Theory and Techniques, 40 (1992), pp. 1353–1363.

[18] V. TORCZSON, *On the convergence of the multidimensional direct search*, SIAM J. Optim., 1 (1991), pp. 123–145.

[19] T. A. WINSLOW, R. J. TREW, P. GILMORE, AND C. T. KELLEY, *Doping profiles for optimum class* B *performance of GaAs mesfet amplifiers*, in Proceedings IEEE/Cornell Conference on Advanced Concepts in High Speed Devices and Circuits, IEEE, 1991, pp. 188–197.

[20] ———, *Simulated performance optimization of GaAs MESFET amplifiers*, in Proceedings IEEE/Cornell Conference on Advanced Concepts in High Speed Devices and Circuits, IEEE, 1991, pp. 393–402.

SIAM J. OPTIMIZATION
Vol. 5, No. 2, pp. 286–313, May 1995

005

INDEFINITE TRUST REGION SUBPROBLEMS AND NONSYMMETRIC EIGENVALUE PERTURBATIONS*

RONALD J. STERN† AND HENRY WOLKOWICZ‡

Abstract. This paper extends the theory of trust region subproblems in two ways: (i) it allows indefinite inner products in the quadratic constraint, and (ii) it uses a two-sided (upper and lower bound) quadratic constraint. Characterizations of optimality are presented that have no gap between necessity and sufficiency. Conditions for the existence of solutions are given in terms of the definiteness of a matrix pencil. A simple dual program is introduced that involves the maximization of a strictly concave function on an interval. This dual program simplifies the theory and algorithms for trust region subproblems. It also illustrates that the trust region subproblems are implicit convex programming problems, and thus explains why they are so tractable.

The duality theory also provides connections to eigenvalue perturbation theory. Trust region subproblems with zero linear term in the objective function correspond to eigenvalue problems, and adding a linear term in the objective function is seen to correspond to a perturbed eigenvalue problem. Some eigenvalue interlacing results are presented.

Key words. indefinite trust region subproblems, existence and optimality conditions, numerical solutions, hard case, matrix pencils, nonsymmetric eigenvalue perturbation theory

AMS subject classifications. 49M37, 65K05, 65K10, 90C30

1. Introduction.

Calculation of the step between iterates in *trust region* numerical methods for minimization problems involves the minimization of a quadratic objective function subject to a norm constraint. This trust region subproblem is

$$(\tilde{P}) \qquad \begin{array}{ll} \min & \mu(y) := y^t B y - 2\psi^t y \\ \text{subject to} & Ay = b, \\ & y^t D y \leq \delta,\ y \in \Re^n, \end{array}$$

where $\psi \in \Re^n$; $B \in \Re^{n\times n}$ is symmetric, A is $m \times n$; $b \in \Re^m$, D is a positive definite scaling matrix, and $\delta > 0$ is the trust region radius. The objective function μ provides a quadratic model of a merit function, while the linear constraint $Ay = b$ is a linear model of possibly nonlinear constraints. Note that the trust region quadratic constraint has the implicit, or hidden, constraint $0 \leq y^t D y$, while a positive δ yields the standard generalized Slater constraint qualification of convex programming.

By representing the linear constraint $Ay = b$ as $y = \hat{y} + Zw$, where the range of Z is equal to the null space of A, and $\hat{y}$ is a particular solution of $Ay = b$, we can eliminate this linear constraint. Moreover, we can also eliminate the scaling matrix D and use complementary slackness to get the simplified problem

$$(\tilde{P}_E) \qquad \begin{array}{ll} \min & \mu(y) := y^t B y - 2\psi^t y \\ \text{subject to} & y^t y = 1,\ y \in \Re^n. \end{array}$$

Trust region problems have proven to be very successful and important in both unconstrained and constrained optimization. The theory, algorithms, and applications

* Received by the editors January 27, 1993; accepted for publication (in revised form) September 22, 1993. This research was supported by the Natural Sciences Engineering Research Council Canada.

† Concordia University, Department of Mathematics and Statistics, Montreal, Quebec H4B 1R6, Canada (stern@vax2.concordia.ca).

‡ University of Waterloo, Department of Combinatorics and Optimization, Waterloo, Ontario N2L 3G1, Canada(hwolkowi@orion.uwaterloo.ca).

have been described in many papers and textbooks; see, e.g., [3], [9], [6], [11]–[13], [23], [24], [30], [31], [33]. A well-known algorithm for numerically approximating a global minimum is given in [13] and [26]. Other numerical algorithms are presented in [15], [12]. Recently, the trust region subproblem, with the additional linear constraint $Ax = b$, has been employed as the basic step in the affine scaling variation of interior point methods for solving linear programming problems; see, e.g., [7], [2], [35]. Affine scaling methods for general quadratic programming problems, which solve a trust region subproblem at each step, are given in [16]. In addition, many continuous relaxations of discrete optimization problems result in norm constraints and therefore trust region subproblems arise; see, e.g., [22] for a survey.

Generalizations of $(\tilde{P})$ are also important. Subproblems with two trust region constraints appear in sequential quadratic programming (SQP) algorithms; see, e.g., [4], [39], [37]. In [37], an algorithm is presented that treats the two trust region problem by restricting it to two dimensions. More recently, Zhang [40] treated the two trust region problem using a parametric approach and assuming positive definiteness of the objective function. (In both [39] and [40], the condition that $B - \lambda C$ is positive definite for some λ, where C is the Hessian for the second trust region constraint, is very important. This condition is studied here for the indefinite case and shown to be equally important.) Two trust region subproblems also appear in parametric identification problems; see, e.g., [21], [17]. Moreover, it is often useful to consider modelling the general nonlinear programming problem using quadratic approximations for both the objective function as well as for the constraints; see, e.g., [5], [27]. Such problems have up to now been considered too difficult to solve without further modelling using linear approximations for the constraints. One reason for this is that the quadratic approximations can result in indefinite Hessians for the objective function as well as for the constraints, resulting in possible unboundedness and infeasibility problems.

The success of trust region methods depends in part on the fact that one can characterize, and hence numerically approximate, the global minimum of the subproblem $(\tilde{P})$. The characterization, which has no gap between necessity and sufficiency, is independent of any convexity assumptions on the quadratic function μ; that is, B can be indefinite. The choice of the scaling matrix D can be very important. It is currently restricted to be positive definite in order to maintain tractability of the subproblem, but it would be advantageous and important to allow a larger class of matrices in order to obtain scale invariance; see, e.g., [9, p. 59]. Of more interest and importance is the fact that the feasible set $\{y : y^t y = 1\}$ in $(\tilde{P}_E)$ being nonconvex does not present a problem in the characterization of optimality. Note that we can add $k(y^t y - 1)$, $k > 0$, to the objective function without changing the optimum. Thus if k is large, then the objective function becomes convex. This means that we can assume that the objective function is convex if desired. However, this is no longer true if the constraint $y^t y = 1$ is changed to an indefinite constraint.

In case $\psi = 0$ (no linear term) the stationary points of the trust region subproblem correspond to the eigenvalues of B. In [32], the authors related stationarity properties of $(\tilde{P})$ to spectral properties of the parametric border perturbation of B given by

$$A(t) = \begin{pmatrix} B & -\psi \\ \psi^t & t \end{pmatrix}. \tag{1.1}$$

Hence, the above perturbation of B has, as an analog, the perturbation of the purely quadratic function $y^t B y$ by the linear term $-2\psi^t x$ in $(\tilde{P})$. Other connections between trust region problems and eigenvalue problems are known in the literature.

If one considers a symmetric perturbation in (1.1), then connections with the trust region problem are studied in [29] and show up in the theory of divide and conquer algorithms for symmetric eigenvalue problems; see, e.g., [1]. Moreover, the algorithms in [13] and [26] are based on finding a Lagrange multiplier smaller than the smallest eigenvalue of B, and therefore guaranteeing positive definiteness of the Hessian of the Lagrangian. The success and importance of trust region methods in both unconstrained and constrained optimization can be attributed to the fact that the subproblems can be solved very efficiently and robustly, which can be attributed to their being implicit eigenvalue problems.

In this paper we consider generalizing $(\tilde{P})$ in two ways and relating these trust region subproblems to eigenvalue perturbation theory. The ellipsoidal constraint $y^tDy \leq \delta$ is replaced by a two-sided constraint, while the positive definite scaling matrix D is replaced by a possibly indefinite matrix C. Specifically, we consider the problem

$$(P) \qquad \begin{array}{ll} \min & \mu(y) = y^tBy - 2\psi^ty \\ \text{subject to} & \beta \leq y^tCy \leq \alpha,\ y \in \Re^n, \end{array}$$

where B and C are symmetric matrices with no definiteness assumed, and $-\infty \leq \beta \leq \alpha \leq \infty$. The motivation for this paper is to extend the existing theory of trust region subproblems (in light of the above discussion on applications) in the hope that this will be a step in the direction of solving general problems with quadratic objectives and quadratic constraints. Note that unlike the definite case, a change of variables will not reduce the problem to the form $(\tilde{P})$. Moreover, it is not clear that solving the equality constrained problem is equivalent to solving the inequality constrained problem, along with a complementary slackness condition. For example, if

$$B = \begin{bmatrix} -1 & 0 \\ 0 & 3 \end{bmatrix}, \qquad C = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix},$$

then the equality constrained problem $y^tCy = 1$ is bounded below while the inequality case, with $\beta = -\infty$, $\alpha = 1$, is unbounded.

Indefinite quadratic constraints arise when considering indefinite inner product spaces or Minkowski spaces; see, e.g., [14], [8]. In this case, the generalized distance function, or norm, arising from the indefinite inner product, can be zero and/or complex valued. The two-sided constraint is a step toward the solution of problems with two quadratic constraints and generalizes the standard problem where the left-hand side constraint is implicitly understood to be ≥ 0.

The paper is organized as follows. In §2 we give necessary and sufficient optimality conditions for (P), as well as a general existence theorem. Then in §3 a further analysis is undertaken. We transform (P) to a "standard form" where the matrix pencil $B - \lambda C$ satisfies a certain regularity condition, and use this form to catalog the various conditions under which an optimum for (P) can exist.

In §4, we apply our results to obtain spectral information regarding the completely general parametric border perturbation of B given by

$$(1.2) \qquad A(t) = \begin{pmatrix} B & u \\ v^t & t \end{pmatrix},$$

under the assumption that the spectral decomposition of B is known.

In §5 we present a general dual program for (P). This dual program is a true concave maximization problem and shows that these trust region subproblems are

implicitly convex. Moreover, the dual program provides bounds on the optimal value of (P). This provides stopping criteria for algorithms for (P) based on duality gap considerations.

We conclude with an appendix to show how the algorithm and results in [13] and [26] can be extended to our more general two-sided indefinite trust region subproblems. Note that an interior point primal-dual algorithm, based on the duality theory given here, is presented in [28].

1.1. Notations. $M \succ 0$ means that a real symmetric matrix M is *positive definite*, while $M \succeq 0$ indicates that M is *positive semidefinite*. (The reverse notations $M \prec 0$, $M \preceq 0$ will be used to denote *negative definiteness* and *negative semidefiniteness*, respectively.) $\mathcal{R}(M)$ denotes the *range space* of M; while $\mathcal{N}(M)$ denotes the *null space* of M. $M^\dagger$ is the *Moore–Penrose generalized inverse* of M. For

$$\lambda \in \Re,\ (\lambda)_+ := \begin{cases} \lambda & \text{if } \lambda \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

2. Optimality conditions. Our results will generally be stated for the minimization problem

$$(P) \qquad \begin{array}{ll} \min & \mu(y) = y^t B y - 2\psi^t y \\ \text{subject to} & \beta \leq y^t C y \leq \alpha,\ y \in \Re^n, \end{array}$$

where $-\infty \leq \beta \leq \alpha \leq \infty$, and both B and C may be indefinite. The maximization versions of the results will always be analogous in an obvious way.

We have the following theorem, which extends a result of Gay [13] and Sorensen [30], where C was assumed to be positive definite and $\beta = 0 < \alpha$ is implicitly assumed; see, also, Fletcher [9]. Our theorem does not tell us when problem (P) possesses a minimizing point, but rather, it tells us when a given feasible point yields a minimum. There is no gap between the necessary and sufficient optimality conditions and there is no assumption on boundedness of the feasible set or the objective function. The three optimality conditions are, respectively, stationarity, nonnegative definiteness, and complementary slackness and multiplier sign.

THEOREM 2.1. *Let y be a feasible point for (P). Then y gives the global minimum for (P) if there exists a Lagrange multiplier $\lambda \in \Re$ such that*

$$(B - \lambda C)y = \psi, \tag{2.1}$$

$$B - \lambda C \succeq 0, \tag{2.2}$$

and

$$\lambda(\beta - y^t C y) \geq 0 \geq \lambda(y^t C y - \alpha). \tag{2.3}$$

Furthermore, if

$$B - \lambda C \succ 0, \tag{2.4}$$

then y is the unique minimizing point. Moreover, suppose that the following constraint qualification holds:

$$Cy = 0 \ \text{ implies } \beta < 0 < \alpha. \tag{2.5}$$

Then y solves (P) if and only if the conditions (2.1)–(2.3) *hold, for some* $\lambda \in \Re$.

Proof. First consider sufficiency. Let y, λ satisfy the above optimality conditions, where y is feasible. There are three cases to consider. □

Case (i). Suppose that $\beta < y^tCy < \alpha$. Then, the optimality condition (2.3) implies that $\lambda = 0$. That μ is convex now follows from (2.2). Thus y is a global unconstrained minimum of the convex function μ and y solves (P).

Case (ii). Suppose that

$$y^tCy = \alpha \tag{2.6}$$

By (2.1),(2.2), we see that y minimizes the *Lagrangian function*

$$L(z, \lambda) := \mu(z) - \lambda(z^tCz - \alpha)$$

over $\Re^n$. That is,

$$\mu(y) = L(y, \lambda) \leq L(z, \lambda) \qquad \forall z \in \Re^n.$$

Since $\beta < y^tCy$ implies $\lambda \leq 0$, it follows that $\lambda(z^tCz - \alpha) \geq 0$, for all feasible z. This in turn yields $\mu(y) \leq \mu(z)$, for all feasible z.

Case (iii). Suppose that $y^tCy = \beta$. Then the conclusion follows similarly to Case (ii).

This proves the if part. The furthermore part of the theorem now follows easily.

Now consider the necessity part of the statement. If $Cy = 0$, then the constraint qualification implies that we have an unconstrained problem and the optimality conditions hold trivially with $\lambda = 0$. Otherwise, we again need to consider the same three cases. For Case (i), we again conclude that the quadratic function μ must be convex. Therefore we can choose $\lambda = 0$ to satisfy the optimality conditions. For Case (ii), we associate with the constraint the (isotropic) cone

$$K = \{w \in \Re^n : w^tCw = 0\}.$$

(Note that the standard linear independence constraint qualification holds, since $Cy \neq 0$ by the constraint qualification assumption.) Suppose that y solves (P). By differentiating the Lagrangian function with respect to y, we obtain the Lagrange equation (2.1) as a first-order necessary condition for optimality. Hence there exists $\lambda \leq 0$ such that (2.1) holds, and it only remains to verify the second-order condition (2.2). Let us denote by T_y the set of tangent directions to the constraint at y; that is,

$$T_y = \{w \in \Re^n : w^tCy = 0\}.$$

The standard second-order conditions state that $B - \lambda C$ is positive semidefinite on T_y. Now let $v \in \Re^n$ be a direction such that

$$v \notin K \cup T_y. \tag{2.7}$$

For each such v, we can construct a feasible point $z = y + \theta v$, where $\theta \neq 0$ and $z^tCz = \alpha$. In order to accomplish this, consider the solvability of the equation

$$(y + \theta v)^tC(y + \theta v) = \alpha. \tag{2.8}$$

This becomes

$$y^tCy + 2\theta v^tCy + \theta^2 v^tCv = \alpha, \tag{2.9}$$

and from (2.6), this in turn becomes

$$\theta[2v^tCy + \theta v^tCv] = 0. \tag{2.10}$$

Now in view of (2.7), we see that

$$\theta = \frac{-2v^tCy}{v^tCv} \tag{2.11}$$

has the required properties.

Note that the value of the Lagrangian at a feasible point satisfying (2.6) is equal to the value of the objective function at that point. Moreover, the Lagrangian is a quadratic and so the second-order Taylor expansions are exact:

$$\begin{aligned} L(z,\lambda) &= L(y,\lambda) + (z-y)^t\left[\nabla L(y,\lambda) + \frac{1}{2}\nabla^2 L(y,\lambda)(z-y)\right] \\ &= L(y,\lambda) + (z-y)^t\frac{1}{2}\nabla^2 L(y,\lambda)(z-y). \end{aligned}$$

This means that

$$\mu(z) - \mu(y) = (z-y)^t(B-\lambda C)(z-y). \tag{2.12}$$

Thus the optimality of y implies

$$v^t(B-\lambda C)v \geq 0 \qquad \forall v \notin K \cup T_y. \tag{2.13}$$

Since the set K has no interior points, by analyticity of the function y^tCy, we see that (2.13), the standard second-order conditions on T_y, and a continuity argument yield (2.2).

Case (iii) with $\beta = y^tCy$ follows similarly.

Remark 1. One can use homogenization to apply Theorem 2.1 to more general quadratic constraints, namely, $y^tCy + \zeta^t y$, where C is nonsingular.

Remark 2. The optimality conditions (2.1),(2.2),(2.3) are a compact version of the usual optimality conditions with two constraints that involve two multipliers.

Terminology. If λ is such that the Lagrange equation (2.1) holds for a feasible y, then λ is called a *Lagrange multiplier* and we say that y is a *stationary point* belonging to λ. The set of all such y is denoted by $\Sigma(\lambda)$, while the set of all Lagrange multipliers is denoted by Λ.

In view of Theorem 2.1, we get the following necessary condition on the symmetric matrix pencil $B - \lambda C$ for (P) to possess a minimizing point.

COROLLARY 2.2. *Suppose that C is nonsingular and* $\max\{|\alpha|, |\beta|\} > 0$. *If y solves (P), then*

$$\exists\ \hat{\lambda} \in \Re \ \text{ s.t. } \ B - \hat{\lambda}C \succeq 0. \tag{2.14}$$

Proof. If $y \neq 0$, then the result follows from the nonsingularity of C and Theorem 2.1. If $y = 0$, then necessarily we have $\psi = 0$, the optimal value $\mu^* = 0$, and

$\beta \leq 0 \leq \alpha$. Since we cannot have both α and β equal to 0, we can assume without loss of generality that $\beta < 0$. Therefore optimality implies that the system

$$y^t C y < 0, \; y^t B y < 0$$

is inconsistent. The result now follows from the theorem of the alternative in Lemma 2.3 in [38]. □

Before stating our main existence result (Theorem 2.4 below), we distinguish between two subcases of (2.14).

• We say that we are in the *regular case* or the *positive definite pencil case* provided that (P) is feasible and

$$\exists\, \hat{\lambda} \in \Re \text{ s.t. } B - \hat{\lambda} C \succ 0. \tag{2.15}$$

• We are in the *irregular case* or the *positive semidefinite pencil case* if (P) is feasible and (2.14) holds, but for no $\lambda \in \Re$ do we have $B - \lambda C \succ 0$.

Remark 3. Characterizations of various definiteness properties for matrix pencils were given by Hershkowitz and Schneider [18] and by Tsing and Uhlig [34]. See also [14]. We do not use those results in this paper, however.

In the next section we see that in the regular case, the set

$$J := \{\lambda \in \Re : B - \lambda C \succ 0\}$$

is an open subinterval of the real line which is bounded if C is indefinite and unbounded if C is definite. On the other hand, in the irregular case with a nonsingular pencil $B - tC$, the number $\hat{\lambda}$ is unique. This is taken up in the following lemma. (Recall that a pencil being singular means that $\det(B - tC) \equiv 0$. The pencil is nonsingular, for example, if either B or C is a nonsingular matrix.)

LEMMA 2.3. *Suppose that the irregular case holds and the function* $\det(B - tC)$ *is not identically* 0 *in* t. *Then there is only one value* $\hat{\lambda}$ *such that* (2.14) *holds.*

Proof. Suppose that $\delta \neq \hat{\lambda}$ is such that $B - \delta C \succeq 0$. Then any convex combination of $B - \hat{\lambda} C$ and $B - \delta C$ is positive semidefinite. Hence

$$B - \hat{\lambda} C - \alpha(\delta - \hat{\lambda}) C \succeq 0 \qquad \forall \alpha \in [0, 1]. \tag{2.16}$$

Now consider the analytic function

$$h(\alpha) = \det[B - \hat{\lambda} C - \alpha(\delta - \hat{\lambda}) C].$$

Then $h(0) = \det(B - \hat{\lambda} C) = 0$, and the assumption on the determinant implies that there exists $\beta \in \Re$ such that $h(\beta) \neq 0$. Hence analyticity implies that $h(\alpha) \neq 0$ for all sufficiently small $\alpha > 0$. But then by (2.16), for such α we would have

$$B - \hat{\lambda} C - \alpha(\delta - \hat{\lambda}) C \succ 0,$$

which contradicts being in the irregular case. □

We now have the following result regarding the existence of a minimizing point for problem (P).

THEOREM 2.4. *Consider problem* (P) *with* C *nonsingular.*

1. *If* (P) *possesses a minimizing point and* $\max\{|\alpha|, |\beta|\} > 0$, *then condition* (2.14) *holds.*

2. *Conversely, assume that* (P) *is feasible,* (2.14) *holds, and both* α *and* β *are finite. Then we have the following cases.*

(a) regular. (P) *possesses a minimizing point.*

(b) irregular. (P) *possesses a minimizing point if and only if* (2.1) – (2.3) *are consistent, in which case* y *is a minimizing point with associated Lagrange multiplier* $\hat{\lambda}$.

Proof. Part 1 follows immediately from Corollary 2.2. To prove Part 2(a) assume that (2.15) holds and suppose, to the contrary, that (P) does not possess a minimum. Suppose that y is a feasible point. Then there would exist a sequence of feasible vectors $\{y_i\}_{i=1}^{\infty}$ such that

$$\mu(y_i) \leq \mu(y) \quad \forall i, \tag{2.17}$$

and

$$||y_i|| \longrightarrow \infty \quad \text{as} \quad i \longrightarrow \infty. \tag{2.18}$$

Without loss of generality we can assume that

$$\frac{y_i}{||y_i||} \longrightarrow d \quad \text{as} \quad i \longrightarrow \infty. \tag{2.19}$$

We claim that

$$d^t B d \leq 0. \tag{2.20}$$

If this did not hold, then $d^t B d = a > 0$ would imply that

$$\frac{y_i^t B y_i}{||y_i||^2} > a/2, \tag{2.21}$$

for all sufficiently large i. But then (2.18) yields $\mu(y_i) \longrightarrow \infty$, contradicting (2.17). Now (2.20) and the regularity condition (2.15) together imply

$$\hat{\lambda} d^t C d < b < 0, \tag{2.22}$$

for some b. It then follows that

$$\frac{\hat{\lambda} y_i^t C y_i}{||y_i||^2} < b/2, \tag{2.23}$$

for all sufficiently large i. From (2.18) we then obtain $|y_i^t C y_i| \longrightarrow \infty$ as $i \longrightarrow \infty$, which contradicts the feasibility of the sequence $\{y_i\}_{i=1}^{\infty}$. This completes the proof of Part 2(a).

To prove Part 2(b), suppose that the optimality conditions are satisfied. Since we have an irregular pencil, i.e., λ is unique in (2.14) and $B - \lambda C$ is singular, then from the lemma in [20, p. 408], the two systems

$$\begin{aligned} Bu &= 0, \quad u^t C u < 0, \\ Bv &= 0, \quad v^t C v > 0, \end{aligned}$$

must be consistent. Therefore, if $y^t C y < \beta$, we can find a feasible point using $y + tv$, since $(y + tv)^t C (y + tv) > \beta$, for sufficiently large t. Similarly, we can use $y + tu$ if $y^t C y > \alpha$. These points satisfy the stationarity conditions and so are optimal. □

The following is an example of the irregular case, with the necessary and sufficient optimality conditions in Part 2(b) of the previous theorem not holding, i.e., with the problem being unbounded. Take $B = \text{diag}(2, -2)$, $C = \text{diag}(-1, 1)$, and $\psi = (1, 2)$. It is readily checked that there does not exist a minimizing y for problem (P). Now note that $\hat{\lambda} = -2$ and $B - \hat{\lambda} C = 0$. Hence the equation $(B - \hat{\lambda} C) y = \psi$ is inconsistent.

3. Further analysis($\alpha = \beta = 1$). In this section we treat the special case of (P) where C is nonsingular and $\alpha = \beta = 1$, i.e., we have the single constraint problem

$$\begin{array}{ll} \min & \mu(y) := y^t B y - 2\psi^t y \\ \text{subject to} & y^t C y = 1,\ y \in \Re^n. \end{array}$$

The results are used in our analysis of eigenvalue perturbations.

3.1. The regular case. The condition (2.15) implies that there exists a nonsingular real $n \times n$ matrix T such that $T^t B T = D$ and $T^t C T = S$ are both diagonal. (We here are utilizing a well-known result on simultaneous diagonalization via congruence; see, e.g., Theorem 7.6.4 in Horn and Johnson [19].) By building a permutation and a scaling into T if necessary, we can without loss of generality assume that the matrices D and S are of the forms

$$D = \operatorname{diag}(D^a, D^b) = \operatorname{diag}(d_1^a, d_2^a, \ldots, d_{n_a}^a, d_1^b, d_2^b, \ldots, d_{n_b}^b)$$

and

$$S = \operatorname{diag}(-I^a, I^b),$$

where

$$d_1^a \geq d_2^a \geq \cdots \geq d_{n_a}^a,$$

$$d_1^b \geq d_2^b \geq \cdots \geq d_{n_b}^b,$$

and where I^a and I^b denote identity matrices of orders n_a and n_b, respectively. Here

$$n_a + n_b = n,$$

where possibly $n_a = 0$. By Sylvester's Theorem of Inertia, see, e.g., [19], feasibility of (P) is equivalent to $n_b > 0$.

We now introduce the problem

$$(P_T) \qquad \begin{array}{ll} \min & \mu_T(x) := x^t D x - 2\eta^t x \\ \text{subject to} & x^t S x = 1. \end{array}$$

Upon identifying

$$y = Tx$$

and

$$\eta = T^t \psi,$$

it is easy to check that

$$\mu(y) = \mu_T(x)$$

and

$$y^t C y = x^t S x.$$

Furthermore, it is clear that in the regular case, the problems (P) and (P_T) have the same Lagrange multiplier set Λ, and for each Lagrange multiplier λ we have

$$\Sigma(\lambda) = T\Sigma_T(\lambda),$$

where $\Sigma_T(\lambda)$ denotes the set of stationary points of problem (P_T) belonging to λ. Finally, it will be convenient to write

$$\eta \doteq \begin{pmatrix} \eta^a \\ \eta^b \end{pmatrix},$$

where the number of components of η^a and η^b are n_a and n_b, respectively.

Whenever the regular case holds, we can accomplish this transformation of (P) to (P_T), which we say is a problem in *standard form*. The regular case of the standardized problem will now be discussed. Hence we shall assume that (P_T) is feasible (i.e., $n_b > 0$) and

$$\exists\, \hat{\lambda} \in \Re \text{ s.t. } D - \hat{\lambda}S \succ 0. \tag{3.1}$$

Two subcases of (3.1) are going to be considered. These will be referred to as the "easy" and "hard" subcases. Our analysis of these subcases generalizes that found in Gander, Golub, and Von Matt [12], where it was assumed that $n_a = 0$; that is, $S = I$. (See also [33].)

3.1.1. The easy subcase. In this subcase of (3.1), we assume feasibility and

$$\eta \text{ is not orthogonal to } \mathcal{N}(D - \lambda S) \quad \text{for all } \lambda \text{ s.t. } \mathcal{N}(D - \lambda S) \neq 0. \tag{3.2}$$

Equivalently, if $\mathcal{I} = \{i : (D - \lambda S)_{ii} = 0\}$, then there exists at least one component $i \in \mathcal{I}$ such that $\eta_i \neq 0$. It is important to note that for fixed $\lambda \in \Re$ we then have

$$\eta \in \mathcal{R}(D - \lambda S) \Longrightarrow \operatorname{rank}(D - \lambda S) = n. \tag{3.3}$$

In other words, (3.2) implies that consistency of the first-order condition

$$(D - \lambda S)x = \eta \tag{3.4}$$

yields invertibility of $D - \lambda S$. (Likewise, consistency of (2.1) implies invertibility of $B - \lambda C$ when (3.2) holds.) For $\lambda \in \Lambda$, denote the unique solution to (3.4) by

$$x_\lambda = (D - \lambda S)^{-1}\eta. \tag{3.5}$$

Let us now introduce the function

$$f_T(\lambda) := 1 - \eta^t(D - \lambda S)^{-2}S\eta. \tag{3.6}$$

(Note that $f_T(\lambda) := 1 - y_\lambda^t C y_\lambda$, where $y_\lambda = (B - \lambda C)^{-1}\psi$). Since

$$D - \lambda S = S(SD - \lambda I), \tag{3.7}$$

it follows that the singularities of $f_T(\cdot)$ are the eigenvalues of SD. We call $f_T(\cdot)$ the *secular function* for problem (P_T). It reduces to the secular function in [12] when $n_a = 0$. Define

$$\Gamma_a := \{i : \eta_i^a \neq 0\}$$

and

$$\Gamma_b := \{j : \eta_j^b \neq 0\}.$$

One can readily check that

$$f_T(\lambda) = 1 + \sum_{\substack{i=1 \\ i\in\Gamma_a}}^{n_a} \frac{(\eta_i^a)^2}{(d_i^a + \lambda)^2} - \sum_{\substack{j=1 \\ i\in\Gamma_a}}^{n_b} \frac{(\eta_j^b)^2}{(d_j^b - \lambda)^2}. \tag{3.8}$$

Remark 4. Note the use of the distinct subscripts i and j in (3.8). This is adopted here, and in what follows, for notational convenience.

Feasibility of x_λ for (P_T) is characterized by the equivalence

$$x_\lambda^t S x_\lambda = 1 \iff f_T(\lambda) = 0. \tag{3.9}$$

It is now clear that in the easy subcase of (3.1),

$$\lambda \in \Lambda \iff f_T(\lambda) = 0, \tag{3.10}$$

in which case x_λ, as given by (3.5), is the unique associated stationary point.

Since $n_b > 0$ (feasibility of (P_T)), the assumption that (3.1) holds implies that either

$$n_a > 0 \text{ and } -d_{n_a}^a < d_{n_b}^b \tag{3.11}$$

or

$$n_a = 0. \tag{3.12}$$

(See Figs. 1 and 2 for plots of g_T for the above two cases, respectively.) If (3.11) holds, then

$$B - \lambda C \succ 0 \iff D - \lambda S \succ 0 \iff \lambda \in (-d_{n_a}^a, d_{n_b}^b), \tag{3.13}$$

while if (3.12) holds, then

$$B - \lambda C \succ 0 \iff D - \lambda S \succ 0 \iff \lambda \in (-\infty, d_{n_b}^b). \tag{3.14}$$

We summarize the above discussion in the following lemma.

LEMMA 3.1. *Problem (P_T) is feasible if and only if C is not negative semidefinite. Moreover, the set of t where $B - tC$ is positive definite is an open interval which is bounded if and only if C is indefinite.*

We now introduce the function

$$g_T(\lambda) := \lambda - \eta^t (D - \lambda S)^{-1} \eta, \tag{3.15}$$

which is called the *secular antiderivative function* for problem (P_T). The implicit form of this function is

$$g(\lambda) := \lambda - \psi^t (B - \lambda C)^{-1} \psi. \tag{3.16}$$

The singularities of $g_T(\cdot)$ are those of $f_T(\cdot)$ and, what is more,

$$g_T'(\lambda) = f_T(\lambda) \tag{3.17}$$

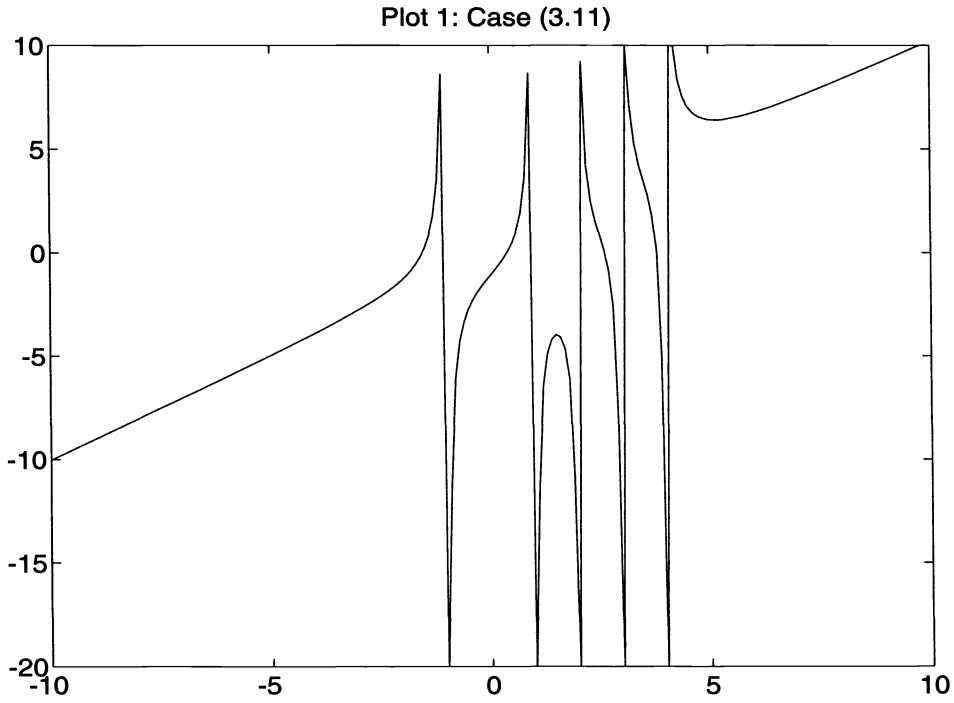


FIG. 1. *Secular antideriviative.*

at all real numbers λ which are not singularities. It is readily verified that

$$g_T(\lambda) = \lambda - \sum_{\substack{i=1 \\ i\in\Gamma_a}}^{n_a} \frac{(\eta_i^a)^2}{(d_i^a + \lambda)} - \sum_{\substack{j=1 \\ i\in\Gamma_a}}^{n_b} \frac{(\eta_j^b)^2}{(d_j^b - \lambda)}. \tag{3.18}$$

At nonsingular points λ we also have

$$g_T''(\lambda) = -2\eta^t(D - \lambda S)^{-3}\eta. \tag{3.19}$$

Now suppose that (3.11) holds. Then, by (3.19), $g_T(\cdot)$ is strictly concave on the interval $(-d_{n_a}^a, d_{n_b}^b)$, which by (3.13) is where $B - \lambda C \succ 0$. Furthermore,

$$g_T(\lambda) \downarrow -\infty \quad \text{as} \quad \lambda \downarrow -d_{n_a}^a$$

and

$$g_T(\lambda) \downarrow -\infty \quad \text{as} \quad \lambda \uparrow d_{n_b}^b.$$

Then there exists a unique $\lambda^* \in (-d_{n_a}^a, d_{n_b}^b)$ such that $g_T'(\lambda^*) = f_T(\lambda^*) = 0$. It follows that $\lambda^* \in \Lambda$, and x_{λ^*} minimizes (P_T). Also, for indices $i < n_a$ we have

$$g_T(\lambda) \downarrow -\infty \quad \text{as} \quad \lambda \downarrow -d_i^a$$

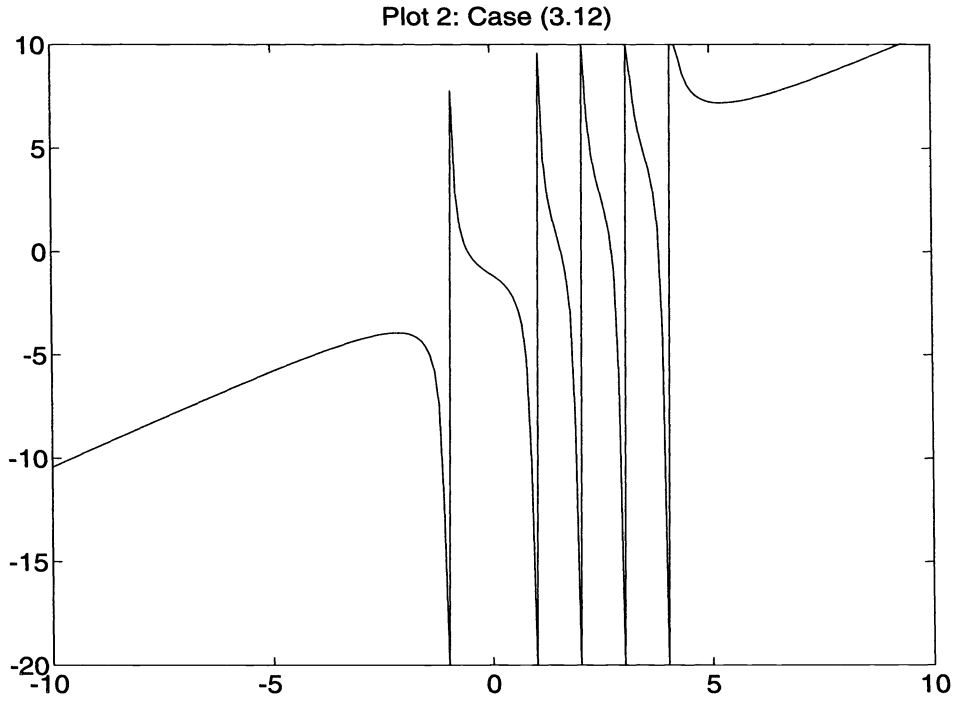


FIG. 2. *Secular antideriviative.*

and

$$g_T(\lambda) \uparrow \infty \quad \text{as} \quad \lambda \uparrow -d_i^a,$$

while for indices $j < n_b$ we have

$$g_T(\lambda) \downarrow -\infty \quad \text{as} \quad \lambda \uparrow d_j^b$$

and

$$g_T(\lambda) \uparrow \infty \quad \text{as} \quad \lambda \downarrow d_j^b,$$

Now suppose that (3.12) holds. Then $g_T(\cdot)$ is strictly concave on $(-\infty, d_{n_b}^b)$,

$$g_T(\lambda) \downarrow -\infty \quad \text{as} \quad \lambda \downarrow -\infty$$

and

$$g_T(\lambda) \downarrow -\infty \quad \text{as} \quad \lambda \uparrow d_{n_b}^b.$$

Hence there exists a unique $\lambda^* \in (-\infty, d_{n_b}^b)$ such that $g_T'(\lambda^*) = f_T(\lambda^*) = 0$. Then $\lambda^* \in \Lambda$, and x_{λ^*} minimizes problem (P_T). For indices $j < n_b$ we have the same behavior as when (3.11) holds.

Remark 5. If one analyzes the function $g_T(\cdot)$ in the special case where $n_a = 0$, additional graphical properties may be obtained. In particular, one can exploit the fact that $g_T'''(\lambda) < 0$ at every nonsingular point λ; see [32] for the details.

We now can give the following existence and uniqueness result for the easy case. It includes a necessary and sufficient condition for the simultaneous existence of a maximizing point and a minimizing point for (P).

THEOREM 3.2. *Consider the easy subcase of the regular case of problem* (P); *that is,* (P) *is feasible (i.e.,* $n_b > 0$*), and* (3.1), (3.2) *hold. Then we have the following:*

1. *The set of Lagrange multipliers* Λ *is finite*;
2. (P) *possesses a unique minimizing point*;
3. (P) *possesses a maximizing point if and only if* $n_a = 0$.

Proof. We can without loss of generality assume that (P) is in the standard form (P_T). Parts 1 and 2 of the theorem follow from the discussion above. (In Part 1 we used the rational property of $f_T(\cdot)$ on any open interval that does not contain a singularity, i.e., using a common denominator reduces the problem to finding the zeros of a polynomial in λ, since the denominator is positive on the open interval.) In order to prove Part 3, assume first that $n_a > 0$. For there to exist a maximizing point, there would necessarily exist $\tilde{\lambda} \in \Lambda$ such that

$$(D - \tilde{\lambda}S) \leq 0. \tag{3.20}$$

This implies

$$d_1^b \leq -d_1^a, \tag{3.21}$$

which contradicts (3.11). Sufficiency in Part 3 follows from compactness of the feasible set and continuity. □

Remark 6. Let the hypotheses of Theorem 3.2 hold, with $n_a > 0$. In view of the preceding discussion, we see that there exists at least one Lagrange multiplier $\tilde{\lambda}$ (that is, a critical point of $g_T(\cdot)$) such that $g_T''(\tilde{\lambda}) > 0$. In particular, there must be such a number in the interval (d_1^b, ∞). However, in view of the preceding theorem, the corresponding stationary point $x_{\tilde{\lambda}}$ does not give a maximum for problem (P_T). In fact, in Example 4.1 in §4, it will be seen that $x_{\tilde{\lambda}}$ need not even give a local maximum.

We conclude the discussion of the easy subcase with a key lemma that will be used in the following sections. The lemma also provides a (concave) dual program. The lemma asserts that in the easy case, the values of the secular antiderivative at its critical points (which are the Lagrange multipliers) equal the values of the objective functions of (P) and its standardization at the corresponding set of stationary points. A variant of this result can be found in [11]; see also [32]. The proof is by direct substitution, and is omitted.

LEMMA 3.3. *Let the hypotheses of Lemma* 3.2 *hold, and let* $\lambda \in \Lambda$. *Then*

$$g_T(\lambda) = \mu_T(x_\lambda) = \mu(y_\lambda), \tag{3.22}$$

where $y_\lambda = (B - \lambda C)^{-1}\psi$.

The above results yield the following dual program, i.e., the optimal values of the primal and dual are equal. The details are presented in §5.

$$\text{DUAL PROGRAM} \qquad \begin{array}{ll} \max & g_T(\lambda) \\ \text{subject to} & B - \lambda C \succeq 0. \end{array}$$

3.1.2. The hard subcase. In this subcase of (3.1), we assume feasibility and the following condition:

$$\eta \text{ is orthogonal to } \mathcal{N}(D - \lambda S) \neq 0 \quad \text{for some } \lambda \in \Re. \tag{3.23}$$

As in the easy case, there is an equivalent statement concerning components of η that are now 0 and correspond to the 0 components of $D - \lambda S$. We again use the (possibly empty) index sets

$$\Gamma_a = \{i : \eta_i^a \neq 0\}$$

and

$$\Gamma_b = \{j : \eta_j^b \neq 0\}.$$

The secular function for problem (P_T) is

$$f_T(\lambda) = 1 + \sum_{\substack{i=1 \\ i\in\Gamma_a}}^{n_a} \frac{(\eta_i^a)^2}{(d_i^a + \lambda)^2} - \sum_{\substack{j=1 \\ j\in\Gamma_b}}^{n_b} \frac{(\eta_j^b)^2}{(d_j^b - \lambda)^2}, \tag{3.24}$$

and the secular antiderivative correspondingly becomes

$$g_T(\lambda) = \lambda - \sum_{\substack{i=1 \\ i\in\Gamma_a}}^{n_a} \frac{(\eta_i^a)^2}{(d_i^a + \lambda)} - \sum_{\substack{j=1 \\ j\in\Gamma_b}}^{n_b} \frac{(\eta_j^b)^2}{(d_j^b - \lambda)}. \tag{3.25}$$

Since we are still in the regular case, the interval J of real numbers λ for which the matrix pencil $D - \lambda S \succ 0$ is given by

$$J = \begin{cases} (-\infty, d_{n_b}^b) & \text{when } n_a = 0, \\ (-d_{n_a}^a, d_{n_b}^b) & \text{when } n_a > 0. \end{cases}$$

The following discussion deals with both forms of J at once. We introduce the index sets

$$\Delta_a = \{i : d_i^a = d_{n_a}^a\}$$

and

$$\Delta_b = \{j : d_j^b = d_{n_b}^b\}.$$

1. If J contains a critical point λ^* of $g_T(\cdot)$, then necessarily $\eta \neq 0$ and $g_T{}'' < 0$. This implies that we have an isolated local maximum of $g_T(\cdot)$ and $\lambda^* \in \Lambda$, since a unique minimizing point x for problem (P_T) can be obtained by solving $(D - \lambda^* S)x = \eta$. Necessarily then $x_i \neq 0$ for all $i \in \Gamma_a$, $x_j \neq 0$ for all $j \in \Gamma_b$, and automatically $x^t S x = 1$ since $f_T(\lambda^*) = 0$.

2. Now suppose that J does not contain a critical point of $g_T(\cdot)$. Then since $g_T(\cdot)$ is concave on J, we see that $g_T(\cdot)$ is monotone on J. We need to consider both the monotone-increasing and monotone-decreasing possibilities.

(a) If $g_T'(\lambda) > 0$ on J, then

$$\Delta_b \cap \Gamma_b = \phi,$$

for otherwise the assumed monotonicity is violated. (Here ϕ denotes the empty set.) Therefore $g_T(\lambda)$ has no pole for $\lambda = d_{n_b}^b$. Also,

$$\lambda^* = d_{n_b}^b \in \Lambda,$$

because a minimizing vector x for (P_T) can be found by simultaneously solving $(D - \lambda^* S)x = \eta$ and $x^t S x = 1$. Necessarily then

$$x_i = 0 \qquad \forall i \notin \Gamma_a$$

and

$$x_j = 0 \qquad \forall j \notin \Delta_b \cup \Gamma_b.$$

Note that selected components x_j can be nonzero for $j \in \Delta_b$, $j \notin \Gamma_b$. The set of vectors x thusly obtained is a submanifold of $\Re^{n-1}$. However, x will be unique in the special case where $g_T'(d^b_{n_b}) = 0$.

(b) If $g_T'(\lambda) < 0$ on J, then monotonicity yields

$$\Delta_a \cap \Gamma_a = \phi.$$

Recall that $n_a > 0$. Also,

$$\lambda^* = -d^a_{n_a} \in \Lambda,$$

because now a minimizing vector x for (P_T) can be found by simultaneously solving $(D - \lambda^* S)x = \eta$ and $x^t S x = 1$. Then

$$x_j = 0 \qquad \forall j \notin \Gamma_b$$

and

$$x_i = 0 \qquad \forall i \notin \Delta_a \cup \Gamma_a.$$

Since certain components x_i may be nonzero for $i \in \Delta_a$, $i \notin \Gamma_a$, the set of vectors x obtained in this way is a submanifold of $\Re^{n-1}$, with x being unique in the special case where $g_T'(d^a_{n_a}) = 0$.

3.2. The irregular case. In the irregular case it may be that (P) cannot be transformed into standard form. Nevertheless, we will study the irregular case of the standard form problem (P_T). We therefore assume that

$$\exists\, \hat{\lambda} \in \Re \text{ s.t. } D - \hat{\lambda} S \geq 0, \tag{3.26}$$

and (by Lemma 2.3) that $\hat{\lambda}$ is unique. We have the following lemma.

LEMMA 3.4. *In the irregular case of the feasible problem (P_T), we have*

$$n_a > 0 \tag{3.27}$$

and

$$-d^a_{n_a} = d^b_{n_b} = \hat{\lambda}. \tag{3.28}$$

Proof. If $n_a = 0$, then we would have $D - \lambda S \succ 0$ on the interval $(-\infty, d^b_{n_b})$, violating the uniqueness of $\hat{\lambda}$. Hence (3.27) holds. Similarly, we must have $-d^a_{n_a} \leq d^b_{n_b}$ for (3.6) to hold, and if $-d^a_{n_a} < d^b_{n_b}$, then $D - \lambda S \succ 0$ on the interval $(-d^a_{n_a}, \hat{d}^b_{n_b})$, which also violates the uniqueness of $\hat{\lambda}$. Consequently, (3.28) holds. □

What follows is an existence theorem for the irregular case of the standard form.

THEOREM 3.5. *Assume that we are in the irregular case of problem* (P_T). *Then* (P_T) *possesses a minimizing point if and only if*

$$\Delta_a \cap \Gamma_a = \phi \tag{3.29}$$

and

$$\Delta_b \cap \Gamma_b = \phi. \tag{3.30}$$

Proof. Suppose that (P_T) possesses a minimizing point x. We first verify (3.29). It must be shown that

$$\eta_i = 0 \qquad \forall i \in \Delta_a.$$

To this end, let $i \in \Delta_a$. Then for the necessary condition (3.4) to hold, we must have

$$(d_i^a + \hat{\lambda})x_i = \eta_i.$$

Since $d_i^a + \hat{\lambda} = 0$, (3.29) follows. Condition (3.4) leads to (3.30) in a similar way.

Now suppose that (3.29) and (3.30) hold. If $g_T'(\hat{\lambda}) \geq 0$, then $\hat{\lambda} \in \Lambda$, since a minimizing vector x^* may be constructed by simultaneously solving $(D - \lambda^* S)x = \eta$ and $x^t S x = 1$. Then

$$x_i = 0 \qquad \forall i \notin \Gamma_a$$

and

$$x_j = 0 \qquad \forall j \notin \Delta_b \cup \Gamma_b.$$

Since selected components x_j can be nonzero for $j \in \Delta_b$, $j \notin \Gamma_b$, it follows that the set of vectors x determined in this way is a submanifold of $\Re^{n-1}$, with x being unique in the special case where $g_T'(\hat{\lambda}) = 0$. The analysis for the possibility $g_T'(\hat{\lambda}) \leq 0$ is similar. □

4. Nonsymmetric eigenvalue perturbations. We wish to obtain spectral information about the real $n \times n$ parametric border perturbation of B given by (1.2); that is

$$A(t) = \begin{pmatrix} B & \alpha \\ \beta^t & t \end{pmatrix},$$

where B is a symmetric $(n-1) \times (n-1)$ matrix. We assume that the spectral decomposition of B is known. In other words, we know an orthogonal matrix P such that $P^t B P$ is diagonal. Then (after including a permutation in P, if necessary) we have a unitary matrix

$$\hat{P} = \begin{pmatrix} P & 0 \\ 0 & 1 \end{pmatrix}$$

such that

$$\hat{A}(t) := \hat{P}^t A(t) \hat{P} = \begin{pmatrix} D_0 & 0 & 0 & \alpha^0 \\ 0 & D_+ & 0 & \alpha^+ \\ 0 & 0 & D_- & \alpha^- \\ (\beta^0)^t & (\beta^+)^t & (\beta^-)^t & t \end{pmatrix}, \tag{4.1}$$

where

$$\begin{aligned} D_0 &= \operatorname{diag}(\gamma_1^0, \gamma_2^0, \ldots, \gamma_{n_0}^0), \\ D_- &= \operatorname{diag}(\gamma_1^-, \gamma_2^-, \ldots, \gamma_{n_-}^-), \\ D_+ &= \operatorname{diag}(\gamma_1^+, \gamma_2^+, \ldots, \gamma_{n_+}^+), \\ \alpha_i^0 \beta_i^0 &= 0 \quad \forall i = 1, 2, \ldots, n_0, \\ \alpha_i^+ \beta_i^+ &> 0 \quad \forall i = 1, 2, \ldots, n_+, \\ \alpha_i^- \beta_i^- &< 0 \quad \forall i = 1, 2, \ldots, n_-, \end{aligned}$$

and

$$n_0 + n_+ + n_- = n - 1.$$

Note that we allow n_0, n_+, or n_- to be zero.

The spectrum of $\hat{A}(t)$ consists of the n_0 numbers γ_i^0 along with the spectrum of

$$\bar{A}(t) := \begin{pmatrix} D_+ & 0 & \alpha^+ \\ 0 & D_- & \alpha^- \\ (\beta^+)^t & (\beta^-)^t & t \end{pmatrix}. \tag{4.2}$$

Hence we focus attention on $\bar{A}(t)$.

Without loss of generality we can assume the diagonal orderings

$$\gamma_1^+ \geq \gamma_2^+ \geq \ldots \gamma_{n_+}^+$$

and

$$\gamma_1^- \geq \gamma_2^- \geq \ldots \gamma_{n_-}^-.$$

Define

$$S = \operatorname{diag}(-1, -1, \ldots, -1, 1, 1, \ldots, 1), \tag{4.3}$$

where the number of -1's is n_+ and the number of 1's is n_-.

We associate with $\bar{A}(t)$ the following problem in $\Re^{\bar{n}}$, where $\bar{n} = n_+ + n_-$:

$$(\bar{P}) \qquad \begin{array}{ll} \min & \bar{\mu}(x) := \bar{x}^t S D \bar{x} - 2\bar{\eta}^t \bar{x} \\ \text{subject to} & \bar{x}^t S \bar{x} = 1. \end{array}$$

Here

$$\bar{\eta} = \begin{pmatrix} \eta^+ \\ \eta^- \end{pmatrix},$$

$$\begin{aligned} D &= \operatorname{diag}(D_+, D_-), \\ \eta_i^+ &= (\alpha_i^+ \beta_i^+)^{1/2} \quad \forall i = 1, 2, \ldots, n_+, \end{aligned}$$

and

$$\eta_i^- = (-\alpha_i^- \beta_i^-)^{1/2} \quad \forall i = 1, 2, \ldots, n_-.$$

It is readily checked that the secular antiderivative associated with $(\bar{P})$ is given by

$$\bar{g}(\lambda) = \lambda + \sum_{i=1}^{n_+} \frac{(\eta_i^+)^2}{(\gamma_i^+ - \lambda)} - \sum_{j=1}^{n_-} \frac{(\eta_j^-)^2}{(\gamma_j^- - \lambda)}. \tag{4.4}$$

We require the following lemma. The proof, which relies on Schur complements, is similar to that of Lemma 3.1 in [32], and is therefore omitted.

LEMMA 4.1. *The real eigenvalues of $\bar{A}(t)$ which differ from the $\bar{n}$ diagonal entries γ_i^+ and γ_j^- are the solutions of*

$$\bar{g}(\lambda) = t. \tag{4.5}$$

The next theorem follows from the discussion in §3.1.1. (Only the minimization version is stated here; the maximization version is analogous.) The theorem gives sufficient conditions for realness of the spectrum of $A(t)$ and describes the associated interlacing.

THEOREM 4.2. *Assume that problem $(\bar{P})$ is feasible; that is, $n_- > 0$, and that either*

$$n_+ > 0 \quad \text{and} \quad \gamma_1^+ < \gamma_{n_-}^- \tag{4.6}$$

or

$$n_+ = 0. \tag{4.7}$$

Then problem $(\bar{P})$ possesses a unique minimizing point, and the following hold.

1. *The matrix $A(t)$ has $n-2$ real eigenvalues, including all the eigenvalues of D_0 and $\bar{n}-1$ eigenvalues of $\bar{A}(t)$ that interlace the $n-1$ eigenvalues of B.*

2. *Suppose that (4.6) holds. Let λ_α denote the unique critical point of $\bar{g}(\cdot)$ in the interval $(\gamma_1^+, \gamma_{n_-}^-)$. A sufficient condition for the other two eigenvalues of $\bar{A}(t)$, say $\delta_a \le \delta_b$, to be real is that*

$$t \le \bar{g}(\lambda_\alpha). \tag{4.8}$$

If the inequality is strict, we get the interlacing

$$\gamma_1^+ < \delta_a < \lambda_\alpha < \delta_b < \gamma_{n_-}^-. \tag{4.9}$$

If the inequality is not strict, then

$$\gamma_1^+ < \delta_a = \lambda_\alpha = \delta_b < \gamma_{n_-}^-. \tag{4.10}$$

3. *Now suppose that (4.7) holds, and let λ_α denote the unique critical point of $\bar{g}(\cdot)$ in the interval $(-\infty, \gamma_{n_-}^-)$. A sufficient condition for the other two eigenvalues of $\bar{A}(t)$, again denoted $\delta_a \le \delta_b$, to be real, is that (4.8) holds. In case (4.8) holds strictly, we obtain the interlacing*

$$-\infty < \delta_a < \lambda_\alpha < \delta_b < \gamma_{n_-}^-. \tag{4.11}$$

If the inequality (4.8) is not strict, then

$$-\infty < \delta_a = \lambda_\alpha = \delta_b < \gamma_{n_-}^-. \tag{4.12}$$

Remark 7. (a) Suppose that problem $(\bar{P})$ is infeasible. (This includes the case of a purely symmetric border perturbation.) Then

$$\bar{g}(\lambda) = \lambda + \sum_{i=1}^{n_+} \frac{(\eta_i^+)^2}{(\gamma_i^+ - \lambda)}. \tag{4.13}$$

From the graphical analysis of this function, one can prove the classical result that for every value t, the spectrum of B interlaces that of $A(t)$. (See Wilkinson [36, §2.39].)

(b) A specialized version of Theorem 4.2 appears in [32], where it was assumed that $n_+ = 0$.

The following example illustrates Lemma 4.1 and the preceding theorem.

Example 4.1. Let

$$A(t) = \bar{A}(t) = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 2 & 1 \\ 1 & -1 & t \end{pmatrix}.$$

Then

$$\bar{g}_T(\lambda) = \lambda + \frac{1}{(1-\lambda)} - \frac{1}{(2-\lambda)}.$$

In view of Theorem 3.2, there is no maximizing point for problem $(\bar{P})$. There is a minimizing point, however, with corresponding Lagrange multiplier $\lambda_\alpha = 1.5310$ and critical value -2.4844. The other critical point is $\lambda_\beta = 2.8832$ with critical value 3.4844. (If one uses MATLAB to graph $\mu_T(\cdot)$, then it is seen that λ_β does not correspond to a local maximum, even though one might suspect this from the graph of $\bar{g}(\cdot)$.) The eigenvalues of $A(t)$ are real if $t \leq -2.4844$ or if $t \geq 3.4844$. For the selected value $t = -3$, the spectrum of $A(-3)$ is $\{-3.0489, 1.3569, 1.6920\}$. The interlacing is of type (4.9).

5. A general dual program. We now return to studying the general program (P):

$$(P) \qquad \begin{array}{ll} \min & \mu(y) = y^t B y - 2\psi^t y \\ \text{subject to} & \beta \leq y^t C y \leq \alpha, \ y \in \Re^n. \end{array}$$

In this section we derive a dual problem for (P) which is a true concave maximization programming problem. This illustrates that (P) is an implicit convex program and shows why the global minimum can be characterized and found. In fact, it is also shown that Lagrangian duality holds without any duality gap.

THEOREM 5.1. *Suppose that y^* solves (P) with optimal value $\mu^* = \mu(y^*)$ and Lagrange multiplier λ^*. Let*

$$L(y, \nu, \omega) = \mu(y) + \nu(\alpha - y^t C y) + \omega(y^t C y - \beta) \tag{5.1}$$

denote the Lagrangian function for (P); let

$$\phi(\nu, \omega) = \inf_y L(y, \nu, \omega) \tag{5.2}$$

denote the Lagrange dual functional; and let

$$h(\nu, \omega) = \nu\alpha - \omega\beta - \psi^t (B - \nu C + \omega C)^{-1} \psi \tag{5.3}$$

denote the quadratic dual functional. Then the optimal value of (P) *satisfies*

$$\mu^* = \max_{\nu \le 0, \omega \le 0} \phi(\nu, \omega), \tag{5.4}$$

while if the regular case holds, then in addition we have

$$\mu^* = \sup_{\substack{B - \nu C + \omega C \succ 0 \\ \nu \le 0, \omega \le 0}} h(\nu, \omega). \tag{5.5}$$

Moreover, the maximum in (5.4) *is attained by*

$$\nu^* = -(-\lambda^*)_+ \quad \text{and} \quad \omega^* = -(\lambda^*)_+, \tag{5.6}$$

where

$$(\lambda)_+ = \begin{cases} \lambda & \text{if } \lambda \ge 0, \\ 0 & \text{otherwise.} \end{cases}$$

Proof. If $B - \nu C + \omega C \succ 0$, then $\phi(\nu, \omega)$ in (5.2) is finite. Moreover, $L(y, \nu, \omega) = \phi(\nu, \omega)$ for $y = (B - \nu C + \omega C)^{-1}\psi$. Substituting for y in L yields

$$h(\nu, \omega) = \phi(\nu, \omega). \tag{5.7}$$

Now if z is feasible for (P), then for all nonpositive ν, ω we have $\phi(\nu, \omega) \le L(z, \nu, \omega) \le \mu(z)$. We now have

$$\begin{aligned} \mu^* &= \min_{z \text{ feasible}} \mu(z) \\ &\ge \sup_{\nu \le 0, \omega \le 0} \phi(\nu, \omega), \\ &= \sup_{\substack{B - \nu C + \omega C \succeq 0 \\ \nu \le 0, \omega \le 0}} \phi(\nu, \omega) \\ &\ge \sup_{\substack{B - \nu C + \omega C \succ 0 \\ \nu \le 0, \omega \le 0}} h(\nu, \omega). \end{aligned} \tag{5.8}$$

Now, from Theorem 2.1, there exists a Lagrange multiplier λ^*. Let ν^* and ω^* be chosen as in (5.6). Then

$$\begin{aligned} \mu^* &= L(y^*, \nu^*, \omega^*) \\ &= \phi(\nu^*, \omega^*) \\ &\le \max_{\nu \le 0, \omega \le 0} \phi(\nu, \omega), \end{aligned}$$

i.e., this and (5.8) imply that (5.4) holds.

If the easy case holds, i.e., $y^* = (B - \lambda^* C)^{-1}\psi$, then (5.7) implies

$$h(\nu^*, \omega^*) = L(y^*, \nu^*, \omega^*) = \mu^*,$$

so (5.8) implies that (5.5) holds. Now suppose that the hard case holds. If $B - \lambda^* C \succ 0$, then (5.4) holds by the above. Now suppose that the regular case holds and $B - \lambda^* C$ is singular. Equivalently, $D - \lambda^* S$ is singular, where $T^t B T = D$ and $T^t C T = S$ are both diagonal, T nonsingular. Let $\bar{y} = T(D - \lambda^* S)^{\dagger} T^t \psi$, where $\dagger$ denotes the

Moore–Penrose generalized inverse. From the optimality conditions, we have that $\psi \in \mathcal{R}(B - \lambda^* C)$. Let $\lambda_k \to \lambda^*$ with $B - \lambda_k C$ positive definite (see Lemma 3.1). Let ν_k, ω_k correspond to λ_k as ν^*, ω^* corresponds to λ^*. Then, from the simultaneous diagonalization, we conclude that

$$y_k = (B - \lambda_k C)^{-1}\psi = T(D - \lambda_k S)^{-1}T^t\psi \to \bar{y}. \tag{5.9}$$

Moreover, $y^* = \bar{y} + z$ for some z in the null space of $B - \lambda^* C$, and we also have $z \perp \psi$. Now $L(y, \nu, \omega) = y^t(B - \nu C + \omega C)y - 2\psi^t y + \nu\alpha - \omega\beta$. So

$$h(\nu_k, \omega_k) = L(y_k, \nu_k, \omega_k) \to L(\bar{y}, \nu^*, \omega^*) = L(y^*, \nu^*, \omega^*) = \mu^*,$$

i.e., this and (5.8) yields (5.5). Attainment follows directly from Theorem 2.1. □

The equality (5.4) provides the standard Lagrangian dual program without any duality gap, while the second equality (5.5) provides a quadratic program type dual. Both duals have no duality gap and both duals are maximizing a concave function over a convex set and so illustrate that the trust region subproblems are implicit convex programs. The constraint qualification avoids trivial exceptional cases such as minimizing x subject to $x^2 \leq 0$. Unfortunately, it can rule out cases where α or β is 0 and 0 is optimal for (P) as well as being an unconstrained minimum for μ, e.g., when $\alpha = 0$, $\psi = 0$, $B \geq 0$. The key observation is that there is no duality gap for the above dual programs. Therefore, we can use a dual algorithm to find the optimal Lagrange multiplier and then worry about the primal optimum point.

The hard case illustrates the difficulty that can arise in duality, i.e., a Lagrange multiplier may exist such that the dual is attained, but the infimum of the Lagrangian may not be attained at a feasible point of the original primal problem.

We can obtain a duality result with only one multiplier.

COROLLARY 5.2. *Suppose that we are in the regular case and y solves (P) with optimal Lagrange multiplier λ. Define*

$$\bar{h}(\lambda) = -(-\lambda)_+\alpha + (\lambda)_+\beta - \psi^t(B - \lambda C)^\dagger\psi. \tag{5.10}$$

Then the optimal value of (P) satisfies

$$\mu^* = \sup_{B - \lambda C \succ 0,} \bar{h}(\lambda). \tag{5.11}$$

Moreover, in the easy case, the maximum is attained, while in the hard case it is attained for λ with $B - \lambda C$ positive semidefinite and possibly singular.

Proof. From the three cases in Theorem 2.1, we see that at least one side of the constraint of (P) can be discarded. Therefore we can assume that at least one of ν or ω is 0 in Theorem 5.1. This yields (5.11). Attainment also follows directly from Theorem 2.1. □

COROLLARY 5.3. *Suppose that $C = I$ and $\beta < 0 < \alpha$, i.e., (P) is the standard trust region subproblem. Let*

$$\bar{g}(\lambda) = \lambda\alpha - \psi^t(B - \lambda C)^\dagger\psi. \tag{5.12}$$

Then

$$\mu^* = \sup_{\substack{B - \lambda C \succ 0 \\ \lambda \leq 0}} \bar{g}(\lambda). \tag{5.13}$$

Moreover, the maximum is attained in the easy case, while it is attained for λ, with $B-\lambda C \succeq 0$ and possibly singular, in the hard case. In addition, if the hard case holds and the Hessian of the Lagrangian at the optimum λ is singular, then

$$\mu^* = \sup_{\substack{B-\lambda C \succ 0 \\ \lambda \leq 0}} \bar{g}(\lambda) = \max_{\substack{B-\lambda C \succeq 0 \\ \lambda \leq 0}} \bar{g}(\lambda). \tag{5.14}$$

Proof. The proof is similar to that of Corollary 5.2. The final statement follows from Theorem 2.1 since the optimum multiplier λ^* is the only point where $B-\lambda C$ is positive semidefinite and singular. □

Note that in the standard version of (P), we could just as well choose $\beta < 0$, which implies that the constraint qualification is automatically satisfied.

6. Appendix. We now follow some of the development in [26] and outline an algorithm for (P) that exploits the Cholesky factorization of $B-\lambda C$. (See Algorithm 6.1.) We assume that C is nonsingular and that the regular case holds, i.e., there exits λ such that $B-\lambda C \succ 0$. In our framework, the algorithm is a primal-dual type algorithm. We maximize the dual function in order to solve the dual problem. Therefore each such iteration provides an improved Lagrange multiplier estimator λ and, by weak duality, an improved lower bound on the optimal value. In addition, if the corresponding solution x_λ is feasible, we get an upper bound on the optimal value. This upper bound is then further improved by moving along a direction of negative curvature toward the boundary of the feasible set. When the gap between lower and upper bounds is small enough, the algorithm stops. Convergence of the algorithm follows immediately from the concavity of the dual function.

This framework also simplifies the description of the algorithm in [26], where the special case that $C = I$ and $0 < \alpha$ is treated. (β can be set to any negative number.) Note that in this case, feasibility of x_λ, i.e., $x_\lambda C x_\lambda \leq \alpha$, is a necessary condition of the hard case and is used as an indicator that the hard case might have occurred. The Newton step in the hard case will generally be too large, which results in slow convergence. However, only in this case do we get the added improvement in the upper bound. A log barrier penalty function can be added to avoid the large step. Thus it appears that the hard case might actually be preferable.

Many of the statements and results are straightforward extensions from [26] and we include some of them for completeness. We include results involving our dual function (see (5.10))

$$\bar{h}(\lambda) = \begin{cases} \lambda\alpha - \psi^t (B-\lambda C)^\dagger \psi & \text{if } \lambda < 0, \\ \lambda\beta - \psi^t (B-\lambda C)^\dagger \psi & \text{if } \lambda \geq 0, \end{cases}$$

and we discuss some of the advantages that occur by using this dual. This dual function is concave on the interval where $B-\lambda C$ is positive definite. It is differentiable if $\lambda \neq 0$ with derivative

$$\bar{h}'(\lambda) = \begin{cases} \alpha - x_\lambda^t C x_\lambda & \text{if } \lambda < 0, \\ \beta - x_\lambda^t C x_\lambda & \text{if } \lambda > 0. \end{cases}$$

(Recall that $x_\lambda = (B-\lambda C)^{-1}\psi$.) The subdifferential at $\lambda = 0$ is the interval

$$\partial \bar{h}(0) = [\beta - x_0^t C x_0, \alpha - x_0^t C x_0].$$

The signs of $\bar{h}'(\lambda)$ and λ tell us which side of the trust region constraint is becoming active. For example, if $\lambda < 0$ and $\alpha > 0$, then we maximize $\bar{h}(\lambda)$ by solving

$$0 = \bar{h}'(\lambda) = \alpha - x_\lambda^t C x_\lambda, \tag{6.1}$$

and exploit the rational structure of this equation by applying Newton's method to solve

$$\phi(\lambda) := \frac{1}{\sqrt{\alpha}} - \frac{1}{\sqrt{x_\lambda^t C x_\lambda}} = 0, \; x_\lambda = (B - \lambda C)^{-1}\psi, \tag{6.2}$$

i.e., we iterate using $\lambda \leftarrow \lambda - \frac{\phi(\lambda)}{\phi'(\lambda)}$. The function ϕ is almost linear but has a singularity where $x_\lambda^t C x_\lambda = 0$. The algorithm is based on solving for feasibility of x_λ, while maintaining the optimality conditions. In our framework we are solving the simple dual problem, which means that we are equivalently maximizing the function $\bar{h}(\lambda)$ rather than just solving (6.1). The dual function does not have the singularity at $x_\lambda^t C x_\lambda = 0$. By using implicit differentiation on the Lagrange equation (2.1), we see that

$$\frac{\partial x_\lambda}{\partial \lambda} = (B - \lambda C)^{-1} C x_\lambda,$$

$$\phi'(\lambda) = \frac{\partial \phi(\lambda)}{\partial \lambda} = \frac{x_\lambda^t C (B - \lambda C)^{-1} C x_\lambda}{(x_\lambda^t C x_\lambda)^{\frac{3}{2}}},$$

$$\bar{h}''(\lambda) = \frac{\partial \bar{h}'(\lambda)}{\partial \lambda} = -2 x_\lambda^t C (B - \lambda C)^{-1} C x_\lambda,$$

and

$$\frac{\phi(\lambda)}{\phi'(\lambda)} = \frac{((x_\lambda^t C x_\lambda)^{\frac{1}{2}} - \sqrt{\alpha})}{\sqrt{\alpha}} \frac{x_\lambda^t C x_\lambda}{x_\lambda^t C (B - \lambda C)^{-1} C x_\lambda}.$$

If both α and $x_\lambda^t C x_\lambda$ are negative, then we can replace them by their negative values in the definition of ϕ. We exploit the Cholesky factorization of the positive definite pencil $B - \lambda C = R^t R$, where R is upper triangular. The following algorithm applies Newton's method to update λ.

ALGORITHM 6.1.

Let λ and x be given with $B - \lambda C = R^t R$ positive definite and $R^t R x = \psi$.
Let $y = Cx$ and $\gamma = y^t x$. Solve $R^t q = y$.
If $\lambda < 0$ or $(\lambda = 0$ and $\gamma > \alpha)$,

If $\alpha\gamma > 0$, let $\lambda = \lambda - \frac{(|\gamma|^{\frac{1}{2}} - \sqrt{|\alpha|})}{\sqrt{|\alpha|}} \frac{\gamma}{q^t q}$.

If $\alpha\gamma \leq 0$, let $\lambda = \lambda - (\gamma - \alpha)\frac{1}{2q^t q}$.

Else if $\lambda > 0$ or $(\lambda = 0$ and $\gamma < \beta)$,

If $\beta\gamma > 0$, let $\lambda = \lambda - \frac{(|\gamma|^{\frac{1}{2}} - \sqrt{|\beta|})}{\sqrt{|\beta|}} \frac{\gamma}{q^t q}$.

If $\beta\gamma \leq 0$, *let* $\lambda = \lambda - (\gamma - \beta)\frac{1}{2q^tq}$.
End

If C is positive definite, then α is positive and this algorithm reduces to that presented in [26]. Note that the algorithm stops if $\psi = 0$. However, this is not a failure, since this indicates that we have solved the dual problem if we solve an eigenvalue problem, e.g., if $\alpha \geq \beta > 0$ we need to solve $\sup_{B-\lambda C \succ 0} \lambda$. We have therefore found the optimal value of the primal problem.

The following lemma is a generalization of Lemma 3.4 in [26]. We have modified the results and added comments to include the role of our dual function. We include the proof for completeness. Note that for $\lambda < 0$, the dual function satisfies

$$\bar{h}(\lambda) = \lambda\alpha - ||Rx||^2,$$

with $\bar{h}(\lambda) = h(\lambda)$ if $B - \lambda C \succ 0$. (The case $\lambda > 0$ follows similarly.)

LEMMA 6.1. *Let* $0 < \sigma < 1$ *be given and suppose that*

$$B - \lambda C = R^tR, \quad (B - \lambda C)x = \psi, \quad \lambda < 0,$$

where $x = (B - \lambda C)^\dagger\psi$ *when* $B - \lambda C$ *is singular. Let* $z \in \Re^n$ *satisfy*

$$(x+z)^tC(x+z) = \alpha, \ ||Rz||^2 \leq \sigma |\ ||Rx||^2 - \lambda\alpha\ | \ (= \sigma|\bar{h}(\lambda)|). \tag{6.3}$$

Then

$$\bar{h}(\lambda) = \lambda\alpha - ||Rx||^2 \leq \mu^* \leq \mu(x+z) \leq \bar{h}(\lambda) + \sigma|\bar{h}(\lambda)|, \tag{6.4}$$

where μ^* *is the optimal value of* (P).

Proof. For any $z \in \Re^n$ we have

$$\mu(x+z) = -(||Rx||^2 - \lambda(x+z)^tC(x+z)) + ||Rz||^2. \tag{6.5}$$

Then for any z which satisfies (6.3), we have

$$\mu(x+z) = \bar{h}(\lambda) + ||Rz||^2 \leq \bar{h}(\lambda) + \sigma|\bar{h}(\lambda)|.$$

Moreover, if $\mu^* = \mu(x+z^*)$, where $x + z^*$ is feasible, then (6.5) implies

$$\mu(x+z^*) \geq -(||Rx||^2 - \lambda\alpha) = \bar{h}(\lambda),$$

i.e., weak duality holds. The last two inequalities yield the lemma. □

From the lemma we now conclude that if $\mu^* \leq 0$, then $-\mu(x+z) \geq (1-\sigma)(-\bar{h}(\lambda))$ and so

$$|\mu(x+z) - \mu^*| = \mu(x+z) - \mu^* \leq \sigma(-\mu^*) = \sigma|\mu^*|.$$

Similarly, if $\mu^* > 0$, then for good approximations λ, we have $\bar{h}(\lambda) > 0$ and $-\mu(x+z) \geq (1+\sigma)(-\bar{h}(\lambda)) \geq (1+\sigma)(-\mu^*)$. Therefore, in both cases we conclude that

$$|\mu(x+z) - \mu^*| \leq \sigma|\mu^*|,$$

i.e., the lemma yields a nearly optimal solution to (P). Alternatively, we get the interval

$$\bar{h}(\lambda) \leq \mu^* \leq \mu(x+z) \leq \bar{h}(\lambda) + \sigma|\bar{h}(\lambda)|. \tag{6.6}$$

(Note that the error $\sigma|\mu^*| \leq \sigma|\bar{h}(\lambda)|$ if $\mu^* \leq 0$, which is the case if, e.g., $y = 0$ is feasible as in the standard trust region subproblem. This is reversed if $\bar{h}(\lambda) > 0$.)

The lemma is used in the case that the current iterate yields a strictly feasible estimate, i.e., $\beta < x_\lambda C x_\lambda < \alpha$. Then a vector z with $||z|| = 1$ and $||Rz||$ small, is computed using a Linpack routine for estimating the smallest singular value. From (6.5), we see that if we can find τ such that $x+\tau z$ satisfies the constraint with equality, then we should get a good improvement in our estimate of the optimum. In addition, note that x_λ is optimal for a subproblem, e.g., if $\lambda < 0$, then x_λ is optimal for (P) with α replaced by $x_\lambda C x_\lambda$. We can therefore continue with a new modified problem with β replaced by $x_\lambda C x_\lambda$. In addition, if we know that the optimal Lagrange multiplier is negative, then we can actually replace β by α.

The lemma provides a stopping criterion since we can conclude that the duality gap is bounded by $\sigma|\mu^*|$. However, a smaller gap is obtained from $|\mu(x+z) - \bar{h}(\lambda)|$.

Safeguarding must be done in order to maintain positive definiteness of the pencil during the iterations. The safeguarding procedure needs parameters $\lambda_L, \lambda_U, \lambda_S$, and λ, such that $[\lambda_L, \lambda_U]$ is an interval of uncertainty that contains the optimal Lagrange multiplier λ^*, while $-\infty \leq \lambda_S \leq \lambda_T \leq \infty$ with the interval $[\lambda_S, \lambda_T]$ containing the interval of positive definiteness. For example, given $B - \lambda C \succ 0$, updating λ_L, λ_U follows from the concavity of the dual function.

ALGORITHM 6.2.

Safeguarding λ:
If $\bar{h}'_T < 0$,
    $\lambda_U = \min\{\lambda_U, \lambda\}$
Else
    $\lambda_L = \max\{\lambda_L, \lambda\}$
End

Note that we do not have to consider $\lambda = 0$ as a special case unless it is the optimal multiplier, in which case the algorithm stops. However, updating λ_S and λ_T does not follow as easily. It is not immediately clear how to use the information from the Cholesky factorization to improve the estimates for the interval of positive definiteness. Note that only one of these needs to be updated since we can immediately determine which side of the current λ the optimal λ^* is on. Initial estimates can be calculated from

$$\lambda_S = \max_{c_{ii}<0} \frac{b_{ii}}{c_{ii}}, \qquad \lambda_T = \min_{c_{ii}>0} \frac{b_{ii}}{c_{ii}}.$$

The following outlines an iteration for an algorithm for (P). Convergence is guaranteed by the properties of the dual program. We have not included the instances where safeguarding and updating of the safeguarding parameters are done.

ALGORITHM 6.3.

Suppose λ and x are given with $B - \lambda C = R^t R$ positive definite and $R^t R x = \psi$.

1. If the convergence criteria is satisfied, then STOP.
2. Take a Newton step as described in Algorithm 6.1.
3. Backtrack if necessary until the dual functional is improved and the pencil is positive definite. (Find the Cholesky factorization $B - \lambda C = R^t R$.)
4. If $\beta < x^t C x < \alpha$, then $\mu(x)$ provides an upper bound on the optimal value; improve this upper bound using, e.g., $\tau, \hat{z}$ or use some other technique for the primal problem.

A MATLAB program has been written and tested on randomly generated problems that satisfy our assumptions. The test results showed an average of 3.4 iterations for convergence. This program can be obtained using anonymous ftp from princeton.edu in the directory pub/henry. See the readme file for the description of the contents of this directory. A detailed numerical study of this algorithm is currently being done. Moreover, the dual program is particularly well-suited for interior point methods. A primal-dual interior point method is presented in [28]. It is shown to be very robust and efficient. In particular, it does not need to treat the hard case in any special way.

Acknowledgments. We would like to thank Urs von Matt for his careful refereeing and many improvements and corrections to the paper. We also wish to mention that since completing this work, we have become aware of related work on duality for the standard trust region subproblem in [10] and work on generalizations for indefinite equality constrained trust region subproblems in [25]. Thanks also to the SOR group at Princeton University as well as DIMACS for their support during the second author's sabbatical research leave.

REFERENCES

[1] J. L. BARLOW, *Error analysis of update methods for the symmetric eigenvalue problem*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 598–618.

[2] E. R. BARNES, *A variation on Karmarkar's algorithm for solving linear programming problems*, Math. Programming, 36 (1986), pp. 174–182.

[3] R. G. CARTER, *Multi-model algorithms for optimization*, Tech. Report TR86-3, 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 Proc. SIAM Conference on Numerical Optimization, Boulder, CO, 1984. Also Tech. Report TR84-1, Department of Mathematical Sciences, Rice University, Houston, TX.

[5] A. R. CONN, *Nonlinear programming, exact penalty functions and projection techniques for non-smooth functions*, in Numerical Optimization, P.T. Boggs, R. Byrd, and R.B. Schnabel, eds., Society for Industrial and Applied Mathematics, 1984, pp. 3–25.

[6] 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.

[7] I. I. DIKIN, *Iterative solution of problems of linear and quadratic programming*, Dokl. Akad. Nauk SSSR, 174 (1967), pp. 747–748. Translation, Soviet Math. Dokl., 8 (1967), pp. 674–675.

[8] H. B. EFIMOV, *Higher Geometry*, Mir, Moscow, 1980. Translated from Russian by P.C. Sinha.

[9] R. FLETCHER, *Practical Methods of Optimization*. John Wiley & Sons, New York, 1987.

[10] O. E. FLIPPO AND B. JANSEN, *Duality and sensitivity in nonconvex quadratic optimization over an ellipsoid*, Tech. Report 93-15, Technical University of Delft, Delft, The Netherlands, 1993.

[11] G. E. FORSYTHE AND G. H. GOLUB, *On the stationary values of a second-degree polynomial on the unit sphere*, SIAM J. Appl. Math., 13 (1965), pp. 1050–1068.

[12] W. GANDER, G. H. GOLUB, AND U. VON MATT, *A constrained eigenvalue problem*, Linear Algebra Appl., 114/115 (1989), pp. 815–839.

[13] D. M. GAY, *Computing optimal locally constrained steps*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 186–197.

[14] I. GOHBERG, P. LANCASTER, AND L. RODMAN, *Matrices and Indefinite Scalar Products*. Birkhauser-Verlag, Basel, 1983.

[15] G. GOLUB AND U. VON MATT, *Quadratically constrained least squares and quadratic problems*, Numer. Math., 59 (1991), pp. 561–580.

[16] C.-G. HAN, P. M. PARDALOS, AND Y. YE, *Computational aspects of an interior point algorithm for quadratic programming problems with box constraints*, in Large Scale Numerical Optimization, T.F. Coleman and Y. Li, eds., Society for Industrial and Applied Mathematics, 1990.

[17] M. HEINKENSCHLOS, *On the solution of a two ball trust region subproblem*, Tech. Report Nr.92-16, Universitat Trier, Mathematik/Informatik, 1992.
[18] D. HERSHKOWITZ AND H. SCHNEIDER, *On the inertia of intervals of matrices*, SIAM J. Matrix Anal. Appl., 11(1990), pp. 565–574.
[19] R. HORN AND C. JOHNSON, *Matrix Analysis*, Cambridge University Press, New York, 1985.
[20] D.G. LUENBERGER, *Linear and Nonlinear Programming*, Addison-Wesley, Reading, MA, 1984.
[21] A. A. MELKMAN AND C. A. MICCHELLI, *Optimal estimation of linear operators in Hilbert spaces from inaccurate data*, SIAM J. Numer. Anal., 16 (1979), pp. 87–105.
[22] B. MOHAR AND S. POLJAK, *Eigenvalues in combinatorial optimization*, Tech. Report 92752, Charles University, Praha, Czechoslovakia, 1992.
[23] J. J. MORÉ, *The Levenberg–Marquardt algorithm: implementation and theory*, in Lecture Notes in Mathematics #630, Numerical Analysis, G.A. Watson, ed., Springer-Verlag, Berlin, Heidelberg, New York, 1977, pp. 105–116.
[24] J. J. MORÉ, *Recent developments in algorithms and software for trust region methods*, Mathematical Programming, the State of the Art, Springer-Verlag, Berlin, New York, 1983, pp. 258–287.
[25] J. J. MORÉ, *Generalizations of the trust region problem*, Tech. Report MCS-P349-0193, Argonne National Labs, Argonne, IL, 1993.
[26] J. J. MORÉ AND D. C. SORENSEN, *Computing a trust region step*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 553–572.
[27] M. J. D. POWELL, *Overview of constrained optimization*, Introductory address, SIAM Conference on Optimization, Chicago, IL, 1992.
[28] F. RENDL, R. J. VANDERBEI, AND H. WOLKOWICZ, *Max-min eigenvalue problems, primal-dual interior point algorithms, and trust region subproblems*, Tech. Report CORR 93-30, Department of Combinatorics and Optimization, Waterloo, Ontario. Also, Proc. of NATO Conf. on Nonlinear Programming, Il-Ciocco, Italy, 1993, to appear.
[29] F. RENDL AND H. WOLKOWICZ, *A framework for trust region subproblems with applications to large scale minimization*, Tech. report, University of Waterloo, Waterloo, Ontario, manuscript.
[30] D. C. SORENSEN, *Newton's method with a model trust region modification*, SIAM J. Numer. Anal., 19 (1982), pp. 409–426.
[31] ———, *Trust region methods for unconstrained minimization*, in Nonlinear Optimization 1981, M.J.D. Powell, ed., Academic Press, London, 1982.
[32] R. STERN AND H. WOLKOWICZ, *Trust regions and nonsymmetric eigenvalue perturbations*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 755–778.
[33] R. J. STERN AND J. J. YE, *Variational analysis of an extended eigenvalue problem*, Linear Algebra Appl., to appear.
[34] N. TSING AND F. UHLIG, *Inertia, numerical range, and zeros of quadratic forms for matrix pencils*, SIAM J. Matrix Anal. Appl., 12(1991), pp.146–159.
[35] R. J. VANDERBEI, M. S. MEKETON, AND B. A. FREEDMAN, *A modification of Karmarkar's linear programming algorithm*, Algorithmica, 1 (1986), pp. 395–407.
[36] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Oxford University Press, London, 1965.
[37] K. A. WILLIAMSON, *Parameter identification in systems of ordinary differential equations*, Ph.D. thesis, Department of Mathematical Sciences, Rice University, Houston, TX, 1990, manuscript.
[38] Y. YUAN, *On a subproblem of trust region algorithms for constrained optimization*, Math. Programming, 47 (1990), pp. 53–63.
[39] Y. YUAN, *A dual algorithm for minimizing a quadratic function with two quadratic constraints*, J. Comput. Math., 9 (1991), pp. 348–359.
[40] Y. ZHANG, *Computing a Celis-Dennis-Tapia trust-region step for equality constrained optimization*, Math. Programming, 55 (1992), pp. 109–124.

SIAM J. OPTIMIZATION
Vol. 5, No. 2, pp. 314–347, May 1995

006

A REDUCED HESSIAN METHOD FOR LARGE-SCALE CONSTRAINED OPTIMIZATION*

LORENZ T. BIEGLER†, JORGE NOCEDAL‡, AND CLAUDIA SCHMID†

Abstract. We propose a quasi-Newton algorithm for solving large optimization problems with nonlinear equality constraints. It is designed for problems with few degrees of freedom and is motivated by the need to use sparse matrix factorizations. The algorithm incorporates a correction vector that approximates the cross term Z^TWYp_Y in order to estimate the curvature in both the range and null spaces of the constraints. The algorithm can be considered to be, in some sense, a practical implementation of an algorithm of Coleman and Conn. We give conditions under which local and superlinear convergence is obtained.

Key words. successive quadratic programming, reduced Hessian methods, constrained optimization, quasi-Newton method, large-scale optimization

AMS subject classifications. 65, 49

1. Introduction.

We consider the nonlinear optimization problem

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

$$\text{subject to } c(x) = 0, \tag{2}$$

where $f : \mathbf{R}^n \to \mathbf{R}$ and $c : \mathbf{R}^n \to \mathbf{R}^m$ are smooth functions. We are particularly interested in the case when the number of variables n is large, and the algorithm we propose, which is a variation of the successive quadratic programming (SQP) method, is designed to be efficient in this case. We assume that the first derivatives of f and c are available, but our algorithm does not require second derivatives.

The SQP method for solving (1)–(2) generates, at an iterate x_k, a search direction d_k by solving

$$\min_{d\in\mathbf{R}^n} g(x_k)^T d + \frac{1}{2} d^T W(x_k) d \tag{3}$$

$$\text{subject to } c(x_k) + A(x_k)^T d = 0, \tag{4}$$

where g denotes the gradient of f, W denotes the Hessian of the Lagrangian function $L(x,\lambda) = f(x) + \lambda^T c(x)$, and A denotes the $n \times m$ matrix of constraint gradients

$$A(x) = [\nabla c_1(x), \ldots, \nabla c_m(x)]. \tag{5}$$

* Received by the editors May 5, 1993; accepted for publication (in revised form) December 1, 1993.

† Chemical Engineering Department, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213. This research was partially support by the Engineering Design Research Center, a National Science Foundation-sponsored Engineering Research Center at Carnegie Mellon University.

‡ Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, Illinois 60208 (nocedal@eecs.nwu.edu) and Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Illinois 060439. This research was supported by National Science Foundation grants CCR-9101359 and ASC-9213149, U.S. Department of Energy grant DE-FG02-87ER25047-A004, and Office of Scientific Computing, U.S. Department of Energy contract W-31-109-Eng-38.

A new iterate is then computed as

$$x_{k+1} = x_k + \alpha_k d_k, \tag{6}$$

where α_k is a steplength parameter chosen so as to reduce the value of the merit function. In this study we use the ℓ_1 merit function

$$\phi_\mu(x) = f(x) + \mu \|c(x)\|_1, \tag{7}$$

where μ is a penalty parameter; see, for example, Conn [13], Han [24], or Fletcher [17]. We could have used other merit functions, but the essential points we wish to convey in this article are not dependent upon the particular choice of the merit function.

The solution of the quadratic program (3)–(4) can be written in a simple form if we choose a suitable basis of $\mathbf{R}^n$ to represent the search direction d_k. For this purpose, we introduce a nonsingular matrix of dimension n, which we write as

$$[Y_k \; Z_k], \tag{8}$$

where $Y_k \in \mathbf{R}^{n\times m}$ and $Z_k \in \mathbf{R}^{n\times(n-m)}$, and we assume that

$$A_k^T Z_k = 0. \tag{9}$$

(From now on we abbreviate $A(x_k)$ as A_k, $g(x_k)$ as g_k, etc.) Thus Z_k is a basis for the tangent space of the constraints. We can now express d_k, the solution to (3)–(4), as

$$d_k = Y_k p_Y + Z_k p_Z, \tag{10}$$

for some vectors $p_Y \in \mathbf{R}^m$ and $p_Z \in \mathbf{R}^{n-m}$. Due to (9) the linear constraints (4) become

$$c_k + A_k^T Y_k p_Y = 0. \tag{11}$$

If we assume that A_k has full column rank, then the nonsingularity of $[Y_k \; Z_k]$ and (9) imply that the matrix $A_k^T Y_k$ is nonsingular, so that p_Y is determined by (11)

$$p_Y = -[A_k^T Y_k]^{-1} c_k. \tag{12}$$

Substituting this in (10), we have

$$d_k = -Y_k [A_k^T Y_k]^{-1} c_k + Z_k p_Z. \tag{13}$$

Note that

$$Y_k [A_k^T Y_k]^{-1} \tag{14}$$

is a right inverse of A_k^T and that the first term in (13) represents a particular solution of the linear equations (4).

We have thus reduced the size of the SQP subproblem, which can now be expressed exclusively in terms of the variables p_Z. Indeed, substituting (10) into (3), considering $Y_k p_Y$ as constant, and ignoring constant terms, we obtain the unconstrained quadratic problem

$$\min_{p_Z \in \mathbf{R}^{n-m}} (Z_k^T g_k + Z_k^T W_k Y_k p_Y)^T p_Z + \frac{1}{2} {p_Z}^T (Z_k^T W_k Z_k) p_Z. \tag{15}$$

If we assume that $Z_k^T W_k Z_k$ is positive definite, the solution of (15) is

$$p_Z = -(Z_k^T W_k Z_k)^{-1}[Z_k^T g_k + Z_k^T W_k Y_k p_Y]. \tag{16}$$

This determines the search direction of the SQP method.

We are particularly interested in the class of problems in which the number of variables n is large, but $n - m$ is small. In this case it is practical to approximate $Z_k^T W_k Z_k$ using a variable metric formula such as Broyden-Fletcher, Goldfarb, Shannon (BFGS). On the other hand, the matrix $Z_k^T W_k Y_k$, of dimension $(n-m) \times m$ may be too expensive to compute directly when m is large. For this reason several authors simply ignore the "cross term" $Z_k^T W_k Y_k p_Y$ in (16) and compute only an approximation to the reduced Hessian $Z_k^T W_k Z_k$; see Coleman and Conn [11], Nocedal and Overton [26], and Xie [29]. This approach is quite adequate when the basis matrices Y_k and Z_k in (8) are chosen to be orthonormal (Gurwitz and Overton [23]).

For large problems, however, computing orthogonal bases can be expensive, and it is more efficient to obtain Y_k and Z_k by simple elimination of variables (cf. Fletcher [17]). Unfortunately, in this case ignoring the cross term $Z_k^T W_k Y_k p_Y$ can make the algorithm inefficient, as is illustrated by an example given in a companion paper (Biegler, Nocedal, and Schmid [1]). The central point is that the range space component $Y_k p_Y$ may be very large, and ignoring the contribution from the cross term in (16) can result in a poor step.

Therefore, here we suggest ways of approximating the cross term $Z_k^T W_k Y_k p_Y$ by a vector w_k,

$$[Z_k^T W_k Y_k] p_Y \approx w_k, \tag{17}$$

without computing the matrix $Z_k^T W_k Y_k$. We consider two approaches for calculating w_k; the first involves an approximation to the matrix $[Z_k^T W_k Y_k]$ using Broyden's update, and the second generates w_k using finite differences. We will show that the rate of convergence of the new algorithm is 1-step Q-superlinear, as opposed to the 2-step superlinear rate for methods that ignore the cross term (Byrd [3] and Yuan [30]). The null space step (16) of our algorithm is given by

$$p_Z = -(Z_k^T W_k Z_k)^{-1}[Z_k^T g_k + \zeta_k w_k], \tag{18}$$

where $0 < \zeta_k \leq 1$ is a damping factor to be discussed later.

To describe our first strategy for computing the vector w_k, we consider a quasi-Newton method in which the rectangular matrix $Z_k^T W_k$ is approximated by a matrix S_k, using Broyden's method. We then obtain w_k by multiplying this matrix by $Y_k p_Y$, that is,

$$w_k = S_k Y_k p_Y.$$

How should S_k be updated? Since $W_{k+1} = \nabla^2_{xx} L(x_{k+1}, \lambda_{k+1})$, we have that

$$Z_k^T W_{k+1}(x_{k+1} - x_k) \approx Z_k^T[\nabla_x L(x_{k+1}, \lambda_{k+1}) - \nabla_x L(x_k, \lambda_{k+1})], \tag{19}$$

when x_{k+1} is close to x_k. We use this relation to establish the following secant equation: we demand that S_{k+1} satisfy

$$S_{k+1}(x_{k+1} - x_k) = Z_k^T[\nabla_x L(x_{k+1}, \lambda_{k+1}) - \nabla_x L(x_k, \lambda_{k+1})]. \tag{20}$$

One point in this derivation requires clarification. In the left-hand side of (19) we have $Z_k^T W_{k+1}$, and not $Z_{k+1}^T W_{k+1}$. We could have used Z_{k+1} in (19), avoiding an inconsistency of indices, but this is not necessary since we will show that using Z_k instead of Z_{k+1} in (20) results in algorithms with all the desirable properties. This fact will not be surprising to readers familiar with the analysis of SQP methods; see, for example, Coleman and Conn [11] or Nocedal and Overton [26]. In addition, using Z_k allows updating of S_{k+1} and B_{k+1} prior to creating Z_{k+1} at the new point.

Let us now consider how to approximate the reduced Hessian matrix $Z_k^T W_k Z_k$. Using (6) and (10) in (20), we obtain

$$[S_{k+1} Z_k] \alpha_k p_Z = -\alpha_k S_{k+1}(Y_k p_Y) + Z_k^T [\nabla_x L(x_{k+1}, \lambda_{k+1}) - \nabla_x L(x_k, \lambda_{k+1})].$$

Since S_{k+1} approximates $Z_k^T W_k$, this suggests the following secant equation for B_{k+1}, the quasi-Newton approximation to the reduced Hessian $Z_k^T W_k Z_k$:

$$B_{k+1} s_k = y_k, \tag{21}$$

where s_k is defined by

$$s_k = \alpha_k p_Z,$$

and y_k by

$$y_k = Z_k^T [\nabla_x L(x_{k+1}, \lambda_{k+1}) - \nabla_x L(x_k, \lambda_{k+1})] - \overline{w}_k, \tag{22}$$

with

$$\overline{w}_k = \alpha_k S_{k+1}(Y_k p_Y). \tag{23}$$

We will update B_k by the BFGS formula (cf. Fletcher [17])

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}, \tag{24}$$

provided $s_k^T y_k$ is sufficiently positive.

We highlight a subtle but important point. We have defined two correction terms, w_k and $\overline{w}_k$. Both are approximations to the cross term $(Z^T W Y) p_Y$. The first term, w_k, which is needed to define the null-space step (18) — and thus the new iterate x_{k+1} — makes use of the matrix S_k. The second term, $\overline{w}_k$, which is used in (22) to define the BFGS update of B_k, is computed by using the new Broyden matrix S_{k+1} and takes into account the steplength α_k. We see below that it is useful to incorporate the most recent information in $\overline{w}_k$. Note that this requires the Broyden update to be applied before the vector y_k for the BFGS update can be calculated from (22).

The Lagrange multiplier estimates λ_k needed in the definition (22) of y_k are defined by

$$\lambda_k = -[Y_k^T A_k]^{-1} Y_k^T g_k. \tag{25}$$

This formula is motivated by the fact that, at a solution x_* of (1)–(2), we have $-g_* = A_* \lambda_*$, and since $Y_* [A_*^T Y_*]^{-1}$ is a right inverse of A_*^T,

$$\lambda_* = -[Y_*^T A_*]^{-1} Y_*^T g_*.$$

Using the same right inverse (14) in the definitions of p_Y and λ_k allows us a convenient simplification in the formulae presented in the following sections. We stress, however, that other Lagrange multiplier estimates can be used and that the best choice in practice might be the one that involves the least computation or storage.

We can now outline the sequential quadratic programming method analyzed in this paper.

ALGORITHM I

1. Choose constants $\eta \in (0, 1/2)$ and τ, τ' with $0 < \tau < \tau' < 1$. Set $k := 1$, and choose a starting point x_1 and an $(n-m) \times (n-m)$ symmetric and positive definite starting matrix B_1.
2. Evaluate f_k, g_k, c_k, and A_k, and compute Y_k and Z_k.
3. Compute p_Y by solving the system

$$(A_k^T Y_k) p_Y = -c_k. \qquad \text{(range space step)} \tag{26}$$

4. Compute an approximation w_k to $(Z_k^T W_k Y_k) p_Y$.
5. Choose the damping parameter $\zeta_k \in (0, 1]$ and compute p_Z from

$$B_k p_Z = -[Z_k^T g_k + \zeta_k w_k]. \qquad \text{(null space step)} \tag{27}$$

Define the search direction by

$$d_k = Y_k p_Y + Z_k p_Z. \tag{28}$$

6. Set $\alpha_k = 1$, and choose the weight μ_k of the merit function (7).
7. Test the line search condition

$$\phi_{\mu_k}(x_k + \alpha_k d_k) \leq \phi_{\mu_k}(x_k) + \eta \alpha_k D\phi_{\mu_k}(x_k; d_k), \tag{29}$$

where $D\phi_{\mu_k}(x_k; d_k)$ is the directional derivative of the merit function ϕ in the direction d_k.
8. If (29) is not satisfied, choose a new $\alpha_k \in [\tau \alpha_k, \tau' \alpha_k]$ and go to (7); otherwise set

$$x_{k+1} = x_k + \alpha_k d_k. \tag{30}$$

9. Evaluate $f_{k+1}, g_{k+1}, c_{k+1}$, and A_{k+1}, and compute Y_{k+1} and Z_{k+1}.
10. Compute the Lagrange multiplier estimate

$$\lambda_{k+1} = -[Y_{k+1}^T A_{k+1}]^{-1} Y_{k+1}^T g_{k+1}. \tag{31}$$

Define $\overline{w}_k$ (as discussed in §3), and compute

$$s_k = \alpha_k p_Z \tag{32}$$

and

$$y_k = Z_k^T [\nabla_x L(x_{k+1}, \lambda_{k+1}) - \nabla_x L(x_k, \lambda_{k+1})] - \overline{w}_k. \tag{33}$$

If the update criterion (discussed in §3.3) is satisfied, compute B_{k+1} by the BFGS formula (24); else set $B_{k+1} = B_k$.
11. Set $k := k + 1$, and go to (3).

The algorithm has been left in a very general form, but in the next sections we discuss all its aspects in detail. In §2 we consider the choice of the basis matrices Y_k and Z_k. In §3 we describe the calculation of the correction terms w_k and $\overline{w}_k$, the conditions under which BFGS updating takes place, the choice of the damping parameter ζ_k, and the procedure for updating the weight μ_k in the merit function. In §§4 and 5 we analyze of the local behavior of the algorithm and show that the rate of convergence is at least R-linear. In §6 we present a superlinear convergence result, and some final remarks in §7 conclude the paper.

We now make a few comments about our notation. Throughout the paper, the vectors p_Y and p_Z are computed at x_k and could be denoted by $p_Y{}^{(k)}$ and $p_Z{}^{(k)}$, but we normally omit the superscript for simplicity. The symbol $\|\cdot\|$ denotes the l_2 vector norm or the corresponding induced matrix norm. When using the l_1 or l_∞ norms we indicate it explicitly by writing $\|\cdot\|_1$ or $\|\cdot\|_\infty$. A solution of problem (1) is denoted by x_*, and we define

$$e_k = x_k - x_* \quad \text{and} \quad \sigma_k = \max\{\|e_k\|, \|e_{k+1}\|\}. \tag{34}$$

Here, and for the rest of the paper, $\nabla L(x, \lambda)$ indicates the gradient of the Lagrangian with respect to x only.

2. The basis matrices. As long as Z_k spans the null space of A_k^T, and $[Y_k\ Z_k]$ is nonsingular, the choice of Y_k and Z_k is arbitrary. However, from the viewpoint of numerical stability and robustness of the algorithm, it is desirable to define Y_k and Z_k to be orthonormal, that is,

$$\begin{aligned} Z(x)^T Z(x) &= I_{n-m}, \\ Y(x)^T Y(x) &= I_m, \\ Y(x)^T Z(x) &= 0. \end{aligned}$$

One way of obtaining these matrices is by forming the QR factorization of A. For large problems, however, computing this QR factorization is often too expensive. Therefore many researchers, including Gabay [18], Gilbert [20], Fletcher [17], Murray and Prieto [25], and Xie [30], consider other, nonorthogonal choices of Y and Z. For example, if we partition x into m basic or dependent variables (which without loss of generality are assumed to be the first m variables) and $n - m$ nonbasic or control variables, we induce the partition

$$A(x)^T = [C(x)\ N(x)], \tag{35}$$

where the $m \times m$ basis matrix $C(x)$ is assumed to be nonsingular. We now define $Z(x)$ and $Y(x)$ to be

$$Z(x) = \begin{bmatrix} -C(x)^{-1}N(x) \\ I \end{bmatrix} \quad Y(x) = \begin{bmatrix} I \\ 0 \end{bmatrix}. \tag{36}$$

When $A(x)$ is large and sparse, a sparse LU decomposition of $C(x)$ can often be computed efficiently, and this approach will be considerably less expensive than the QR factorization of A. Note that from the assumed nonsingularity of $C(x)$ both $Y(x)$ and $Z(x)$ vary smoothly with x, provided the same partition of the variables is maintained. In our implementation of the new algorithm (Biegler, Nocedal, and Schmid [1]) we choose Y_k and Z_k by (36).

There is a price to pay for using nonorthogonal bases. If the matrix C is ill conditioned (and this can be difficult to detect), the step computation may be inaccurate. Moreover, even if the basis is well conditioned, the range space step $Y_k p_Y$ can be large, and ignoring the cross term can cause serious difficulties. This phenomenon is illustrated in a two-dimensional example given by Biegler, Nocedal, and Schmid [1]. It is shown in that example that if the cross term $Z_k^T W_k Y_k p_Y$ is ignored, the ratio $\|x_k + d_k\|/\|x_k\|$ can be arbitrarily large, even close to the solution. It is also shown that these inefficiencies disappear if the cross term is approximated as suggested in the following sections.

In the rest of the paper we allow much freedom in the choice of the basis matrices. They can be given by (36), can be orthonormal, or can be chosen in other ways. The only restrictions we impose are that $A_k^T Z_k = 0$ is satisfied, that the $n \times n$ matrix $[Y_k \; Z_k]$ is nonsingular and well conditioned, and that this matrix varies smoothly in a neighborhood of the solution.

3. Further details of the algorithm. In this section we consider how to calculate approximations w_k and $\overline{w}_k$ to $(Z_k^T W_k Y_k)p_Y$ to be used in the determination of the search direction p_Z and in updating B_k, respectively. We also discuss when to skip the BFGS update of the reduced Hessian approximation, as well as the selection of the damping factor ζ_k and the penalty parameter μ_k.

To calculate approximations to $(Z^T W Y)p_Y$, we propose two approaches. First, we consider a finite difference approximation to $Z_k^T W_k$ along the direction $Y_k p_Y$. While this approach requires additional evaluations of reduced gradients at each iteration, it gives rise to a very good step. The second, more economical approach defines w_k and $\overline{w}_k$ in terms of a Broyden approximation to $Z_k^T W_k$, as discussed in §1, and requires no additional function or gradient evaluations. Our algorithm will normally use this second approach, but as we later see, it is sometimes necessary to use finite differences.

3.1. Calculating w_k and $\overline{w}_k$ through finite differences. We first calculate the range space step p_Y at x_k through (26). Next we compute the reduced gradient of the Lagrangian at $x_k + Y_k p_Y$ and define

$$w_k = Z_k^T[\nabla L(x_k + Y_k p_Y, \lambda_k) - \nabla L(x_k, \lambda_k)]. \tag{37}$$

After the step to the new iterate x_{k+1} has been taken, we define

$$\overline{w}_k = Z_k^T[\nabla L(x_k + \alpha_k Y_k p_Y, \lambda_{k+1}) - \nabla L(x_k, \lambda_{k+1})], \tag{38}$$

which requires a new evaluation of gradients if $\alpha_k \neq 1$. Thus, up to three evaluations of the objective function gradient may be required at each iteration.

We note that this finite-difference approach is very similar to the algorithm of Coleman and Conn [10], [11]. Starting at a point z_k, the Coleman–Conn algorithm (with steplength $\alpha_k = 1$) is given by

$$Z_k p_Z = -Z(z_k) B_k^{-1} Z(z_k)^T g(z_k), \tag{39}$$

$$Y_k p_Y = -Y(z_k)[A(z_k)^T Y(z_k)]^{-1} c(z_k + Z_k p_Z), \tag{40}$$

$$z_{k+1} = z_k + Z_k p_Z + Y_k p_Y. \tag{41}$$

Let us now consider Algorithm I, and to better illustrate its similarity with the Coleman and Conn method, let us assume that instead of (37), w_k is defined by

$$w_k = Z(x_k + Y_k p_Y)^T g(x_k + Y_k p_Y) - Z(x_k)^T g(x_k),$$

which differs from (37) by terms of order $O(\|p_Y\|)$. Then Algorithm I with $\alpha_k = 1$ and $\zeta_k = 1$ is given by

$$Y_k p_Y = -Y(x_k)[A(x_k)^T Y(x_k)]^{-1} c(x_k), \tag{42}$$

$$\begin{aligned} Z_k p_Z &= -Z(x_k) B_k^{-1} [Z(x_k)^T g(x_k) + w_k] \\ &= -Z(x_k) B_k^{-1} [Z(x_k + Y_k p_Y)^T g(x_k + Y_k p_Y)], \end{aligned} \tag{43}$$

$$x_{k+1} = x_k + Y_k p_Y + Z_k p_Z. \tag{44}$$

The similarity between the two approaches is apparent in Fig. 1, especially if we consider the intermediate points in the Coleman–Conn iteration to be the starting and final points, respectively.

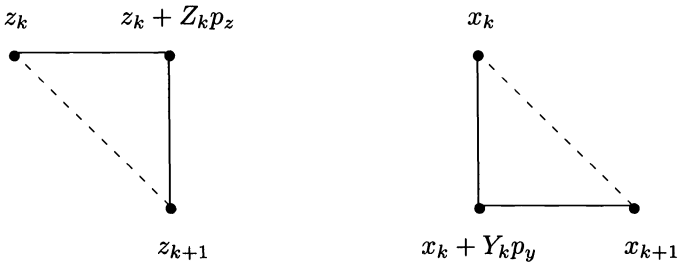


FIG. 1. *Comparison of Coleman–Conn method and Algorithm* I.

In the Coleman–Conn algorithm, the approximation B_k to reduced Hessian $Z_k^T W_k Z_k$ is obtained by moving along the null space direction $Z_k p_Z$, and making a new evaluation of the function and constraint gradients. To be more precise, Coleman and Conn define

$$y_k = Z_k^T [\nabla L(x_k + Z_k p_Z, \lambda_k) - \nabla L(x_k, \lambda_k)]$$

and $s_k = Z_k^T [x_{k+1} - x_k]$ and apply a quasi-Newton formula to update B_k. Algorithm I, using finite differences, amounts essentially to the same thing. To see this, note that if Formula (38) is used in (33), then

$$y_k = Z_k^T [\nabla L(x_{k+1}, \lambda_{k+1}) - \nabla L(x_k + \alpha_k Y_k p_Y, \lambda_{k+1})],$$

which represents a difference in reduced gradients of the Lagrangian along the null space direction $Z_k p_Z$.

Byrd [4] and Gilbert [19] showed that the sequence $\{z_k + Z_k p_Z\}$ (but not the sequence $\{z_k\}$) generated by the Coleman–Conn method converges 1-step Q-superlinearly.

If Algorithm I always computed the correction terms w_k and $\overline{w}_k$ by finite differences, its cost and convergence behavior would be similar to those of the Coleman–Conn method (except when $\alpha_k \neq 1$, which requires one extra gradient evaluation for Algorithm I). However, we are often able to avoid the use of finite differences and instead use the more economical approach discussed next.

3.2. Using Broyden's method to compute w_k and $\overline{w}_k$. We can approximate the rectangular matrix $Z_k^T W_k$ by a matrix S_k updated by Broyden's method, and then compute w_k and $\overline{w}_k$ by post-multiplying this matrix by $Y_k p_Y$ or by a multiple of this vector. As discussed in §1, it is reasonable to impose the secant equation (20) on this Broyden approximation, which can therefore be updated by the formula (cf. Fletcher [17])

$$S_{k+1} = S_k + \frac{(\bar{y}_k - S_k \bar{s}_k)\bar{s}_k^T}{\bar{s}_k^T \bar{s}_k}, \tag{45}$$

where

$$\bar{y}_k = Z_k^T [\nabla L(x_{k+1}, \lambda_{k+1}) - \nabla L(x_k, \lambda_{k+1})] \tag{46}$$

and

$$\bar{s}_k = x_{k+1} - x_k. \tag{47}$$

We now define

$$w_k = S_k Y_k p_Y \quad \text{and} \quad \overline{w}_k = \alpha_k S_{k+1} Y_k p_Y. \tag{48}$$

It should be noted that this approach requires the storage of the $(n-m) \times n$ matrix S_k, in addition to the reduced Hessian approximation, B_k. For problems where $n-m$ is small, this expense is far less than the storage of a full Hessian approximation to W_k. On the other hand, if $n-m$ is not very small, it may be preferable to use a limited-memory implementation of Broyden's method. Here the matrices S_k are represented implicitly, using, for example, the compact representation described in Byrd, Nocedal, and Schnabel [7]. The advantage of the limited memory implementation is that it requires the storage of only a few n–vectors to represent S.

Since there is no guarantee that the Broyden approximations S_k will remain bounded, we need to safeguard them. At the beginning of the algorithm we choose a positive constant Γ and define

$$w_k := \begin{cases} w_k & \text{if } \|w_k\| \leq \frac{\Gamma}{\|p_Y\|^{1/2}} \|p_Y\|, \\ w_k \frac{\Gamma \|p_Y\|^{1/2}}{\|w_k\|} & \text{otherwise.} \end{cases} \tag{49}$$

The correction $\overline{w}_k$ will be safeguarded in a different way. We choose a sequence of positive numbers $\{\gamma_k\}$ such that $\Sigma_{k=1}^{\infty} \gamma_k < \infty$, and we set

$$\overline{w}_k := \begin{cases} \overline{w}_k & \text{if } \|\overline{w}_k\| \leq \alpha_k \|p_Y\| / \gamma_k, \\ \overline{w}_k \frac{\alpha_k \|p_Y\|}{\gamma_k \|\overline{w}_k\|} & \text{otherwise.} \end{cases} \tag{50}$$

As the iterates converge to the solution, $p_Y \to 0$, so that from (48) and from the boundedness of Y_k we see that these safeguards allow the Broyden updates S_k to

become unbounded, but in a controlled manner. We show in §§4 and 5 that with the safeguards (49) and (50) Algorithm I is locally and R-linearly convergent and that this implies that the Broyden updates S_k do, in fact, remain bounded, so that the safeguards become inactive asymptotically.

Our Broyden approximation to the correction terms w_k and $\overline{w}_k$ was motivated by recent work of Gurwitz [22]. She approximates $Z_k^T W_k Z_k$ by the BFGS formula with

$$s_k = Z_k^T[x_{k+1} - x_k]$$

and

$$y_k = Z_k^T[\nabla L(x_{k+1}, \lambda_{k+1}) - \nabla L(x_k, \lambda_{k+1})]$$

and approximates $Z_k^T W_k Y_k$ by a matrix D_k using Broyden's formula (45) with

$$\bar{s}_k = Y_k^T[x_{k+1} - x_k],$$

$$\bar{y}_k = Z_k^T[\nabla L(x_{k+1}, \lambda_{k+1}) - \nabla L(x_k, \lambda_{k+1})] - B_k p_Z.$$

Since the updates may not always be defined, Gurwitz proposes to sometimes skip the update of B_k or D_k. She shows 1-step Q-superlinear convergence *if and only if* one of the updates is taken at each iteration, but this cannot be guaranteed. The analysis of this paper shows that it is preferable to update an approximation to $Z_k^T W_k$, as in Algorithm I, instead of an approximation to $Z_k^T W_k Y_k$, as proposed by Gurwitz, since our approach leads to 1-step superlinear convergence in all cases.

A related method was derived by Coleman and Fenyes [12]. Their lower partition BFGS formula (LPB) simultaneously updates approximations to $Z_k^T W_k Z_k$ and $Z_k^T W_k Y_k$, by means of a new variational problem. The resulting updating formula requires the solution of a cubic equation, and its roots can correspond to cases where updates should be avoided (e.g., $s_k^T y_k \leq 0$). The drawback of this approach is that choosing the correct root is not always easy.

An earlier proposal by Tagliaferro [28] consists of approximating the matrices $Z_k^T W_k Z_k$ and $Z_k^T W_k Y_k$ using the Powell-symmetric-Broyden (PSB) update formula and Broyden's method, respectively. One disadvantage of this approach is that the matrices generated by this updating procedure may become very ill conditioned.

3.3. Update criterion. It is well known that the BFGS update (24) is well defined only if the curvature condition $s_k^T y_k > 0$ is satisfied. This condition can always be enforced in the unconstrained case by performing an appropriate line search; see, for example, Fletcher [17]. When constraints are present, however, the curvature condition $s_k^T y_k > 0$ can be difficult to obtain, even near the solution.

To show this, we first note from (33), (28), and (32) and from the mean value theorem that

$$\begin{aligned} y_k &= Z_k^T \left[\int_0^1 \nabla_{xx}^2 L(x_k + \tau \alpha_k d_k, \lambda_{k+1}) d\tau \right] \alpha_k d_k - \overline{w}_k \\ &\equiv Z_k^T \tilde{W}_k \alpha_k d_k - \overline{w}_k \\ &= Z_k^T \tilde{W}_k Z_k s_k + \alpha_k Z_k^T \tilde{W}_k Y_k p_Y - \overline{w}_k, \end{aligned} \tag{51}$$

where we have defined

$$\tilde{W}_k = \int_0^1 \nabla_{xx}^2 L(x_k + \tau \alpha_k d_k, \lambda_{k+1}) d\tau. \tag{52}$$

Thus

$$s_k^T y_k = s_k^T \left(Z_k^T \tilde{W}_k Z_k \right) s_k + \alpha_k s_k^T \left(Z_k^T \tilde{W}_k Y_k \right) p_Y - s_k^T \overline{w}_k. \tag{53}$$

Near the solution, the first term on the right-hand side will be positive, since $Z_k^T \tilde{W}_k Z_k$ can be assumed positive definite. Nevertheless, the last two terms are of uncertain sign and can make $s_k^T y_k$ negative. Several reduced Hessian methods in the literature set $\overline{w}_k$ equal to zero for all k, and update B_k only if p_Y is small enough compared with s_k that the first term in the right-hand side of (53) dominates the second term (see Nocedal and Overton [26], Gurwitz and Overton [23], and Xie [29]).

Skipping the BFGS update may appear to be a crude heuristic, but we argue that it gives rise to a sound algorithm. First of all, the last two terms in (53) normally converge to zero faster than the first term, so that the right-hand side of (53) will often be positive near the solution and BFGS updating will take place frequently. Furthermore, if the right-hand side of (53) is negative, the range space step $Y_k p_Y$ is relatively large, resulting in sufficient progress towards the solution. These arguments will be made more precise in §5.

We therefore opt for skipping the BFGS update, when necessary, and we now present a strategy for deciding when to do so. Recall that σ_k, defined by (34), converges to zero if the iterates converge to x_*.

Update Criterion I. Choose a constant $\gamma_{\text{fd}} > 0$ and a sequence of positive numbers $\{\gamma_k\}$ such that $\Sigma_{k=1}^{\infty} \gamma_k < \infty$ (this is the same sequence $\{\gamma_k\}$ that was used in (50)).

If $\overline{w}_k$ is computed by Broyden's method, and if both $s_k^T y_k > 0$ and

$$\|p_Y\| \leq \gamma_k^2 \|p_Z\| \tag{54}$$

hold at iteration k, then update the matrix B_k by means of the BFGS formula (24) with s_k and y_k given by (32) and (33). Otherwise, set $B_{k+1} = B_k$.

If $\overline{w}_k$ is computed by finite differences, and if both $s_k^T y_k > 0$ and

$$\|p_Y\| \leq \gamma_{\text{fd}} \|p_Z\| / \sigma_k^{1/2} \tag{55}$$

hold at iteration k, then update the matrix B_k by means of the BFGS formula (24) with s_k and y_k given by (32) and (33). Otherwise, set $B_{k+1} = B_k$.

Note that σ_k requires knowledge of the solution vector x_* and is therefore not computable. However, later we see that σ_k can be replaced by any quantity that is of the same order as the error e_k, for example, the optimality conditions $(\|Z_k^T g_k\| + \|c_k\|)$. Nevertheless, for convenience we will leave σ_k in (55).

We now closely consider the properties of the BFGS matrices B_k when Update Criterion I is used. Let us define

$$\cos \theta_k = \frac{s_k^T B_k s_k}{\|s_k\| \, \|B_k s_k\|}, \tag{56}$$

which, as we will see, is a measure of the goodness of the null space step $Z_k p_Z$. We begin by restating a theorem from Byrd and Nocedal [5] regarding the behavior of $\cos \theta_k$ when the matrix B_k is updated by the BFGS formula.

THEOREM 3.1. *Let $\{B_k\}$ be generated by the* BFGS *formula* (24) *where, for all $k \geq 1$, $s_k \neq 0$ and*

$$\frac{y_k^T s_k}{s_k^T s_k} \geq m > 0, \tag{57}$$

$$\frac{\|y_k\|^2}{y_k^T s_k} \leq M. \tag{58}$$

Then, there exist constants $\beta_1, \beta_2, \beta_3 > 0$ *such that, for any* $k \geq 1$, *the relations*

$$\cos\theta_j \geq \beta_1, \tag{59}$$

$$\beta_2 \leq \frac{\|B_j s_j\|}{\|s_j\|} \leq \beta_3 \tag{60}$$

hold for at least $\lceil \frac{1}{2}k \rceil$ *values of* $j \in [1, k]$.

This theorem refers to the iterates for which BFGS updating takes place; but since, for the other iterates, $B_{k+1} = B_k$, the theorem characterizes the whole sequence of matrices $\{B_k\}$. Theorem 3.1 states that, if $s_k^T y_k$ is always sufficiently positive, in the sense that conditions (57) and (58) are satisfied, then at least half of the iterates at which updating takes place are such that $\cos\theta_j$ is bounded away from zero and $B_j s_j = O(\|s_j\|)$. Since it will be useful to refer easily to these iterates, we make the following definition.

DEFINITION 3.2. *We define* J *to be the set of iterates for which* (59) *and* (60) *hold. We call* J *the set of "good iterates" and define* $J_k = J \cap \{1, 2, \ldots, k\}$.

Note that if the matrices B_k are updated only a finite number of times, their condition number is bounded, and (59)–(60) are satisfied for all k. Thus in this case all iterates are good iterates.

We now study the case when BFGS updating takes place an infinite number of times. Let us assume that all functions under consideration are smooth and bounded. If at a solution point x_* the reduced Hessian $Z_*^T W_* Z_*$ is positive definite, then for all x_k in a neighborhood of x_* the smallest eigenvalue of $Z_k^T \tilde{W}_k Z_k$ is bounded away from zero ($\tilde{W}_k$ is defined in (52)). We now show that in such a neighborhood Update Criterion I implies (57)–(58).

Let us first consider the case when $\overline{w}_k$ is computed by Broyden's method. Using (53), (54), and (50), and since γ_k converges to zero, we have

$$\begin{aligned} s_k^T y_k &\geq C\|s_k\|^2 - O(\gamma_k^2\|s_k\|^2) - O(\gamma_k\|s_k\|^2) \\ &\geq m\|s_k\|^2, \end{aligned} \tag{61}$$

for some positive constants C, m. Also, from (51), (54), and (50) we have that

$$\begin{aligned} \|y_k\| &\leq O(\|s_k\|) + O(\gamma_k^2\|s_k\|) + O(\gamma_k\|s_k\|) \\ &\leq O(\|s_k\|). \end{aligned} \tag{62}$$

We thus see from (61)–(62) that there is a constant M such that for all k for which updating takes place,

$$\frac{\|y_k\|^2}{y_k^T s_k} \leq M,$$

which together with (61) shows that (57)–(58) hold when Broyden's method is used.

If $\overline{w}_k$ is computed by the finite-difference formula (38), we see from (33) and the mean value theorem that there is a matrix $\hat{W}_k$ such that

$$\begin{aligned} y_k &= Z_k^T[\nabla L(x_{k+1}, \lambda_{k+1}) - \nabla L(x_k + \alpha_k Y_k p_Y, \lambda_{k+1})] \\ &\equiv Z_k^T \hat{W}_k Z_k s_k. \end{aligned}$$

Reasoning as before we see that (61) and (62) also hold in this case, and that (57)–(58) are satisfied in the case when finite differences are used. We have therefore established the following result.

LEMMA 3.3. *In a neighborhood of a solution point x_*, and whenever* BFGS *updating takes place as stipulated by Update Criterion* I, $s_k^T y_k$ *is sufficiently positive in the sense that* (57)–(58) *hold.*

3.4. Choosing μ_k and ζ_k. We will now see that by appropriately choosing the penalty parameter μ_k and the damping parameter ζ_k for w_k, the search direction generated by Algorithm I is always a descent direction for the merit function. Moreover, for the good iterates J, it is a direction of strong descent.

Since d_k satisfies the linearized constraint (11), it is easy to show (see Eq. (2.24) of Byrd and Nocedal [6]) that the directional derivative of the ℓ_1 merit function in the direction d_k is given by

$$D\phi_{\mu_k}(x_k; d_k) = g_k^T d_k - \mu_k \|c_k\|_1. \tag{63}$$

The fact that the same right inverse of A_k^T is used in (26) and (31) implies that

$$g_k^T Y_k p_Y = \lambda_k^T c_k. \tag{64}$$

Recalling the decomposition (28) and using (64), we obtain

$$\begin{aligned} D\phi_{\mu_k}(x_k; d_k) &= g_k^T Z_k p_Z - \mu_k \|c_k\|_1 + \lambda_k^T c_k \\ &= (Z_k^T g_k + \zeta_k w_k)^T p_Z - \zeta_k w_k^T p_Z - \mu_k \|c_k\|_1 + \lambda_k^T c_k. \end{aligned} \tag{65}$$

Now from (32) and (27) we have that

$$B_k s_k = -\alpha_k (Z_k^T g_k + \zeta_k w_k). \tag{66}$$

Substituting this in (56), we obtain

$$\cos\theta_k = \frac{-(Z_k^T g_k + \zeta_k w_k)^T p_Z}{\|Z_k^T g_k + \zeta_k w_k\| \, \|p_Z\|}. \tag{67}$$

Recalling the inequality $\lambda_k^T c_k \le \|\lambda_k\|_\infty \|c_k\|_1$, and using (67) in (65), we obtain, for all k,

$$D\phi_{\mu_k}(x_k; d_k) \le -\|Z_k^T g_k + \zeta_k w_k\| \, \|p_Z\| \cos\theta_k - \zeta_k w_k^T p_Z - (\mu_k - \|\lambda_k\|_\infty)\|c_k\|_1. \tag{68}$$

Note also from (66) and (32) that

$$\frac{\|s_k\|}{\|B_k s_k\|} = \frac{\|p_Z\|}{\|Z_k^T g_k + \zeta_k w_k\|}. \tag{69}$$

We now concentrate on the good iterates J, as given in Definition 3.2. If $j \in J$, we have from (69) and (60) that

$$\frac{1}{\beta_3}\|Z_j^T g_j + \zeta_j w_j\| \le \|p_z^{(j)}\| \le \frac{1}{\beta_2}\|Z_j^T g_j + \zeta_j w_j\|. \tag{70}$$

Using this and (59) in (68), we obtain, for $j \in J$,

$$\begin{aligned} D\phi_{\mu_j}(x_j; d_j) &\le -\frac{1}{\beta_3}\|Z_j^T g_j + \zeta_j w_j\|^2 \cos\theta_j - \zeta_k w_j^T p_z^{(j)} - (\mu_j - \|\lambda_j\|_\infty)\|c_j\|_1 \\ &\le -\frac{\beta_1}{\beta_3}\|Z_j^T g_j\|^2 - \frac{2\zeta_j \cos\theta_j}{\beta_3}(g_j^T Z_j w_j) - \zeta_j w_j^T p_z^{(j)} - (\mu_j - \|\lambda_j\|_\infty)\|c_j\|_1, \end{aligned}$$

where we have dropped the nonpositive term $-\zeta_j^2 \cos\theta_j \|w_j\|^2/\beta_3$. Since we can assume that $\beta_3 > 1$ (it is defined as an upper bound in (60)), we have

$$D\phi_{\mu_j}(x_j; d_j) \leq -\frac{\beta_1}{\beta_3}\|Z_j^T g_j\|^2 + \left[2\zeta_j \cos\theta_j |g_j^T Z_j w_j| - \zeta_j w_j^T p_z^{(j)}\right] - (\mu_j - \|\lambda_j\|_\infty)\|c_j\|_1.$$

It is now clear that if

$$2\zeta_j \cos\theta_j |g_j^T Z_j w_j| - \zeta_j w_j^T p_z^{(j)} \leq \rho \|c_j\|_1, \tag{71}$$

for some constant ρ, and if

$$\mu_j \geq \|\lambda_j\|_\infty + 2\rho, \tag{72}$$

then for all $j \in J$,

$$D\phi_{\mu_j}(x_j; d_j) \leq -\frac{\beta_1}{\beta_3}\|Z_j^T g_j\|^2 - \rho\|c_j\|_1. \tag{73}$$

This means that if (71) and (72) hold, then for the good iterates $j \in J$, the search direction d_j is a strong direction of descent for the ℓ_1 merit function in the sense that the first-order reduction is proportional to the Karush–Kuhn–Tucker (KKT) error.

We will choose ζ_k so that (71) holds for *all* iterations. To show how to do this, we note from (27) that

$$p_z = -B_k^{-1} Z_k^T g_k - \zeta_k B_k^{-1} w_k,$$

so that, for $j = k$, (71) can be written as

$$\zeta_k [2\cos\theta_k |g_k^T Z_k w_k| + w_k^T B_k^{-1} Z_k^T g_k + \zeta_k w_k^T B_k^{-1} w_k] \leq \rho \|c_k\|_1. \tag{74}$$

Clearly this condition is satisfied for a sufficiently small and positive value of ζ_k. Specifically, at the beginning of the algorithm we choose a constant $\rho > 0$ and, at every iteration k, define

$$\zeta_k = \min\{1, \hat{\zeta}_k\}, \tag{75}$$

where $\hat{\zeta}_k$ is the largest value that satisfies (74) as an equality.

The penalty parameter μ_k must satisfy (72), so we define it at every iteration of the algorithm by

$$\mu_k = \begin{cases} \mu_{k-1} & \text{if } \mu_{k-1} \geq \|\lambda_k\|_\infty + 2\rho, \\ \|\lambda_k\|_\infty + 3\rho & \text{otherwise.} \end{cases} \tag{76}$$

The damping factor ζ_k and the updating formula for the penalty parameter μ_k have been defined so as to give strong descent for the good iterates J. We now show that they ensure that the search direction is also a direction of descent (but not necessarily of strong descent) for the other iterates, $k \notin J$. Since (71) holds for all iterations by our choice of ζ_k, we have in particular

$$-\zeta_k w_k^T p_z \leq \rho \|c_k\|_1.$$

Using this and (76) in (68), we have

$$D\phi_{\mu_k}(x_k; d_k) \leq -\|Z_k^T g_k + \zeta_k w_k\| \, \|p_z\| \cos\theta_k - \rho_k \|c_k\|_1. \tag{77}$$

The directional derivative is thus nonpositive. Furthermore, since $w_k = 0$ whenever $c_k = 0$ (regardless of whether w_k is obtained by finite differences or through Broyden's method), it is easy to show that this directional derivative can be zero only at a stationary point of problem (1)–(2).

3.5. The algorithm. We can now give a complete description of the algorithm that incorporates all the ideas discussed so far and that specifies the only remaining question, namely, when to apply finite differences and when to use Broyden's method to approximate the cross term. The idea is to consider the relative sizes of p_Y and p_Z. Update Criterion I generates the three regions R_1, R_2, and R_3 illustrated in Fig. 2. The algorithm starts by computing w_k through Broyden's method and by calculating p_Y and p_Z. If the search direction is in R_1 or R_3, we proceed. Otherwise we recompute w_k by finite differences, use this value to recompute p_Z, and proceed. The reason for applying finite differences in this fashion is that in the middle region R_2 Broyden's method is not good enough, nor is the convergence sufficiently tangential, to give a superlinear step. Therefore we must resort to finite differences to obtain a good estimate of w_k. The motivation behind this strategy will become clearer when we study the rate of convergence of the algorithm in §6.

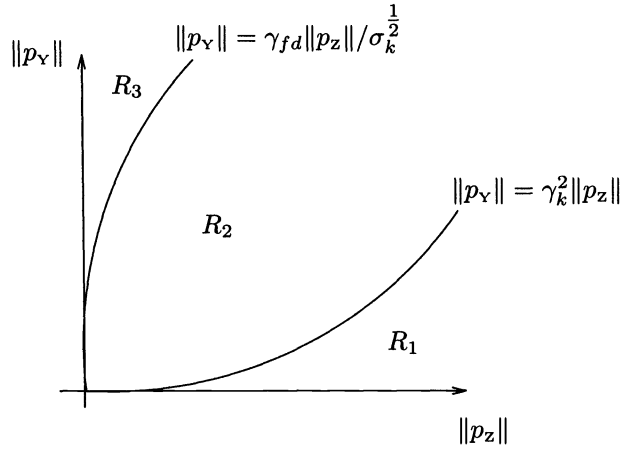


FIG. 2. *Three regions generated by Update Criterion* I.

Note from Updating Criterion I that the BFGS update of B_k is skipped if the search direction is in R_3. A precise description of the algorithm follows.

ALGORITHM II

1. Choose constants $\eta \in (0, 1/2)$, $\rho > 0$ and τ, τ' with $0 < \tau < \tau' < 1$, and positive constants Γ and γ_{fd} for conditions (49) and (55), respectively. For conditions (50) and (54), select a summable sequence of positive numbers $\{\gamma_k\}$. Set $k := 1$, and choose a starting point x_1, an initial value μ_1 for the penalty parameter, an $(n-m) \times (n-m)$ symmetric and positive definite starting matrix B_1, and an $(n-m) \times n$ starting matrix S_1.
2. Evaluate f_k, g_k, c_k, and A_k, and compute Y_k and Z_k.
3. Set findiff = false and compute p_Y by solving the system

$$(A_k^T Y_k) p_Y = -c_k. \qquad \text{(range space step)} \tag{78}$$

4. Calculate w_k using Broyden's method, from (48) and (49).
5. Choose the damping parameter ζ_k from (74) and (75), and compute p_Z from

$$B_k p_Z = -[Z_k^T g_k + \zeta_k w_k]. \qquad \text{(null space step)} \tag{79}$$

6. If (55) is satisfied and (54) is *not* satisfied, set findiff = true and recompute w_k from (37). (In practice we replace σ_k by $\|Z_k^T g_k\| + \|c_k\|$ in (55).)
7. If findiff = true, use this new value of w_k to choose the damping parameter ζ_k from (74) and (75), and recompute p_z from (79).
8. Define the search direction by

$$d_k = Y_k p_Y + Z_k p_Z, \tag{80}$$

and set $\alpha_k = 1$.
9. Test the line search condition

$$\phi_{\mu_k}(x_k + \alpha_k d_k) \leq \phi_{\mu_k}(x_k) + \eta \alpha_k D\phi_{\mu_k}(x_k; d_k). \tag{81}$$

10. If (81) is not satisfied, choose a new $\alpha_k \in [\tau\alpha_k, \tau'\alpha_k]$ and go to step 9; otherwise set

$$x_{k+1} = x_k + \alpha_k d_k. \tag{82}$$

11. Evaluate $f_{k+1}, g_{k+1}, c_{k+1}, A_{k+1}$, and compute Y_{k+1} and Z_{k+1}.
12. Compute the Lagrange multiplier estimate

$$\lambda_{k+1} = -[Y_{k+1}^T A_{k+1}]^{-1} Y_{k+1}^T g_{k+1}, \tag{83}$$

and update μ_k so as to satisfy (76).
13. Update S_{k+1} using (45) to (47). If findiff = false, calculate $\overline{w}_k$ by Broyden's method through (48) and (50); otherwise calculate $\overline{w}_k$ by (38).
14. If ($s_k^T y_k \leq 0$) or if (findiff=true and (55) is not satisfied) or if (findiff=false and (54) is not satisfied), set $B_{k+1} = B_k$. Else, compute

$$s_k = \alpha_k p_Z, \tag{84}$$

$$y_k = Z_k^T [\nabla L(x_{k+1}, \lambda_{k+1}) - \nabla L(x_k, \lambda_{k+1})] - \overline{w}_k, \tag{85}$$

and compute B_{k+1} by the BFGS formula (24).
15. Set $k := k + 1$, and go to 3.

We mentioned in §3.1 that, when using finite differences, there are various ways of defining w_k and $\overline{w}_k$, but for concreteness we now assume in steps 6 and 13 that they are computed by (37) and (38), respectively. We should also point out that the curves in Fig. 2 may intersect, creating a fourth region, and in practice we should stipulate a new set of conditions in this region. We discuss these conditions in another paper that considers the implementation of the algorithm (Biegler, Nocedal, and Schmid [1]).

In the next sections we present several convergence results for Algorithm II. The analysis, which does not assume that the BFGS matrices B_k or the Broyden matrices S_k are bounded, is based on the results of Byrd and Nocedal [6], who have studied the convergence of the Coleman–Conn updating algorithm. We also make use of some results of Xie [29], who has analyzed the algorithm proposed by Nocedal and Overton [26] using nonorthogonal bases Y and Z. The main difference between this paper and that of Xie stems from our use of the correction terms w_k and $\overline{w}_k$, which are not employed in his method.

4. Semilocal behavior of the algorithm. We first show that the merit function ϕ decreases significantly at the good iterates J and that this gives the algorithm a weak convergence property. To establish the results of this section, we make the following assumptions.

Assumption 4.1. The sequence $\{x_k\}$ generated by Algorithm II is contained in a convex set D with the following properties:

(I) The functions $f : \mathbf{R}^n \to \mathbf{R}$ and $c : \mathbf{R}^n \to \mathbf{R}^m$ and their first and second derivatives are uniformly bounded in norm over D.

(II) The matrix $A(x)$ has full column rank for all $x \in D$, and there exist constants γ_0 and β_0 such that

$$\|Y(x)[A(x)^T Y(x)]^{-1}\| \leq \gamma_0, \quad \|Z(x)\| \leq \beta_0, \tag{86}$$

for all $x \in D$.

(III) For all $k \geq 1$ for which B_k is updated, (57) and (58) hold.

(IV) The correction term w_k is chosen so that there is a constant $\kappa > 0$ such that for all k,

$$\|w_k\| \leq \kappa \|c_k\|^{1/2}. \tag{87}$$

Note that Condition (I) is rather strong, since it would often be satisfied only if D is bounded, and it is far from certain that the iterates will remain in a bounded set. Nevertheless, the convergence result of this section can be combined with the local analysis of §5 to give a satisfactory semiglobal result. Condition (II) requires that the basis matrices Y and Z be chosen carefully, and is important to obtain good behavior in practice. Note that (86) and (78) imply that

$$\|Y_k p_Y\| \leq \gamma_0 \|c_k\|. \tag{88}$$

Condition (III) is justified by Lemma 3.1. Condition (III) and Theorem 3.1 ensure that at least half of the iterates at which BFGS updating takes place are good iterates.

We have left some freedom in the choice of w_k since (87) suffices for the analysis of this section. Relation (87) holds for the finite-difference approach, since (37) implies that $w_k = O(Y_k p_Y)$ and since Condition (I) ensures that $\{\|c_k\|\}$ is uniformly bounded (see (121)). Furthermore, the safeguard (49) and (88) immediately imply that (87) is satisfied when the Broyden approximation is used.

The following result concerns the good iterates J, as given in Definition 3.2.

LEMMA 4.1. *If Assumptions* 4.1 *hold and if $\mu_j = \mu$ is constant for all sufficiently large j, then there is a positive constant γ_μ such that for all large $j \in J$,*

$$\phi_\mu(x_j) - \phi_\mu(x_{j+1}) \geq \gamma_\mu \left[\|Z_j^T g_j\|^2 + \|c_j\|_1\right]. \tag{89}$$

Proof. Using (73), we have for all $j \in J$

$$D\phi_{\mu_j}(x_j; d_j) \leq -b_2 \left[\|Z_j^T g_j\|^2 + \|c_j\|_1\right], \tag{90}$$

where $b_2 = \min(\beta_1/\beta_3, \rho)$. Note that the line search enforces the Armijo condition (81),

$$\phi_{\mu_j}(x_j) - \phi_{\mu_j}(x_{j+1}) \geq -\eta \alpha_j D\phi_{\mu_j}(x_j; d_j). \tag{91}$$

It is then clear from (90) that (89) holds, provided the α_j, $j \in J$, can be bounded from below. Suppose that $\alpha_j < 1$, which means that (91) failed for a steplength $\tilde{\alpha}$:

$$\phi_{\mu_j}(x_j + \tilde{\alpha} d_j) - \phi_{\mu_j}(x_j) > \eta \tilde{\alpha} D\phi_{\mu_j}(x_j; d_j), \tag{92}$$

where

$$\tau \tilde{\alpha} \leq \alpha_j \tag{93}$$

(see step 10 of Algorithm II). On the other hand, expanding to second order, we have

$$\phi_{\mu_j}(x_j + \tilde{\alpha} d_j) - \phi_{\mu_j}(x_j) \leq \tilde{\alpha} D\phi_{\mu_j}(x_j; d_j) + \tilde{\alpha}^2 b_1 \|d_j\|^2, \tag{94}$$

where b_1 depends on μ_j. Combining (92) and (94), we have

$$(\eta - 1)\tilde{\alpha} D\phi_{\mu_j}(x_j; d_j) < \tilde{\alpha}^2 b_1 \|d_j\|^2. \tag{95}$$

Next we show that, for $j \in J$,

$$\|d_j\|^2 \leq b_3[\|Z_j^T g_j\|^2 + \|c_j\|_1], \tag{96}$$

for some constant b_3. To do this, we make repeated use of the following elementary result:

$$a, b \geq 0 \quad \Rightarrow \quad a^2 + 2ab + b^2 \leq 3a^2 + 3b^2. \tag{97}$$

Using (80), (97), (86), and (88), we have

$$\begin{aligned} \|d_j\|^2 &\leq \|Z_j p_Z^{(j)}\|^2 + 2\|Z_j p_Z^{(j)}\| \, \|Y_j p_Y^{(j)}\| + \|Y_j p_Y^{(j)}\|^2 \\ &\leq 3\left[\|Z_j p_Z^{(j)}\|^2 + \|Y_j p_Y^{(j)}\|^2\right] \\ &\leq 3\left[\beta_0^2 \|p_Z^{(j)}\|^2 + \gamma_0^2 \|c_j\|^2\right]. \end{aligned} \tag{98}$$

Also by (70), (97), and (87) and noting that $\|\cdot\| \leq \|\cdot\|_1$, we have that for $j \in J$

$$\begin{aligned} \|p_Z^{(j)}\|^2 &\leq \frac{1}{\beta_2^2}\left[\|Z_j^T g_j\|^2 + 2\zeta_j \|Z_j^T g_j\| \, \|w_j\| + \zeta_j^2 \|w_j\|^2\right] \\ &\leq \frac{3}{\beta_2^2}\left[\|Z_j^T g_j\|^2 + \zeta_j^2 \|w_j\|^2\right] \\ &\leq \frac{3}{\beta_2^2}\left[\|Z_j^T g_j\|^2 + \kappa^2 \|c_j\|_1\right], \end{aligned}$$

since $\zeta_j \leq 1$. Since $\|c_j\|_1$ is uniformly bounded on D, we see from this relation and (98) that (96) holds, where

$$b_3 = \max\{9\beta_0^2/\beta_2^2, 3(3\kappa^2\beta_0^2/\beta_2^2 + \gamma_0^2 \sup_{x \in D} \|c(x)\|)\}.$$

Combining (95), (90), and (96), and recalling that $\eta < 1$, we obtain

$$\tilde{\alpha} > \frac{(1 - \eta) b_2}{b_1 b_3}. \tag{99}$$

This relation and (93) imply that the steplengths α_j are bounded away from zero for all $j \in J$. Since by assumption $\mu_j = \mu$ for all large j, we conclude that (89) holds with $\gamma_\mu = \eta b_2 \min\{1, (1-\eta)\tau b_2/(b_1 b_3)\}$. □

It is now easy to show that the penalty parameter settles down and that the set of iterates is not bounded away from stationary points of the problem.

THEOREM 4.2. *If Assumptions* 4.1 *hold, then the weights* $\{\mu_k\}$ *are constant for all sufficiently large* k *and*

$$\liminf_{k\to\infty}(\|Z_k^T g_k\| + \|c_k\|) = 0.$$

Proof. First note that by Assumptions 4.1 (I)–(II) and (83) that $\{\|\lambda_k\|\}$ is bounded. Therefore, since the procedure (76) increases μ_k by at least ρ whenever it changes the penalty parameter, it follows that there are an index k_0 and a value μ such that for all $k > k_0$, $\mu_k = \mu \geq \|\lambda_k\| + 2\rho$.

If BFGS updating is performed an infinite number of times, by Assumption 4.1 (III) and Theorem 3.1 there is an infinite set J of good iterates, and by Lemma 4.1 and the fact that the Armijo condition (81) forces $\phi_\mu(x_k)$ to decrease at each iterate, we have that for $k > k_0$,

$$\begin{aligned}\phi_\mu(x_{k_0}) - \phi_\mu(x_{k+1}) &= \sum_{j=k_0}^{k} (\phi_\mu(x_j) - \phi_\mu(x_{j+1})) \\ &\geq \sum_{j\in J\cap[k_0,k]} (\phi_\mu(x_j) - \phi_\mu(x_{j+1})) \\ &\geq \gamma_\mu \sum_{j\in J\cap[k_0,k]} [\|Z_j^T g_j\|^2 + \|c_j\|_1].\end{aligned}$$

By Assumption 4.1(I) $\phi_\mu(x)$ is bounded below for all $x \in D$, so the last sum is finite, and thus the term inside the square brackets converges to zero. Therefore

$$\lim_{\substack{j\in J\\ j\to\infty}} (\|Z_j^T g_j\| + \|c_j\|_1) = 0. \tag{100}$$

If BFGS updating is performed a finite number of times, then, as discussed after Definition 3.1, all iterates are good iterates, and in this case we obtain the stronger result

$$\lim_{k\to\infty} (\|Z_k^T g_k\| + \|c_k\|_1) = 0. \qquad \square$$

5. Local convergence. In this section we show that if x_* is a local minimizer that satisfies the second-order optimality conditions, and if the penalty parameter μ_k is chosen large enough, then x_* is a point of attraction for the sequence of iterates $\{x_k\}$ generated by Algorithm II. To prove this result, we make the following assumptions. In what follows, G denotes the reduced Hessian of the Lagrangian function, namely,

$$G_k = Z_k^T \nabla_{xx}^2 L(x_k, \lambda_k) Z_k. \tag{101}$$

Assumptions 5.1. The point x_* is a local minimizer for problem (1)–(2), at which the following conditions hold.

1. The functions $f : \mathbf{R}^n \to \mathbf{R}$ and $c : \mathbf{R}^n \to \mathbf{R}^m$ are twice continuously differentiable in a neighborhood of x_*, and their Hessians are Lipschitz continuous in a neighborhood of x_*.

2. The matrix $A(x_*)$ has full column rank. This implies that there exists a vector $\lambda_* \in \mathbf{R}^m$ such that

$$\nabla L(x_*, \lambda_*) = g(x_*) + A(x_*)\lambda_* = 0.$$

3. For all $q \in \mathbf{R}^{n-m}$, $q \neq 0$, we have $q^T G_* q > 0$.

4. There exist constants γ_0, β_0, and γ_c such that, for all x in a neighborhood of x_*,

$$\|Y(x)[A(x)^T Y(x)]^{-1}\| \leq \gamma_0, \quad \|Z(x)\| \leq \beta_0, \tag{102}$$

and

$$\|[Y(x)\ Z(x)]^{-1}\| \leq \gamma_c. \tag{103}$$

5. $Z(x)$ and $\lambda(x)$ are Lipschitz continuous in a neighborhood of x_*. That is, there exist constants γ_z and γ_λ such that

$$\|\lambda(x) - \lambda(z)\| \leq \gamma_\lambda \|x - z\|, \tag{104}$$

$$\|Z(x) - Z(z)\| \leq \gamma_z \|x - z\|, \tag{105}$$

for all x, z near x_*.

Note that conditions 1, 3, and 5 imply that for all (x, λ) sufficiently near (x_*, λ_*), and for all $q \in \mathbf{R}^{n-m}$,

$$m\|q\|^2 \leq q^T G(x, \lambda) q \leq M\|q\|^2, \tag{106}$$

for some positive constants m, M. We also note that Assumptions 5.1 ensure that the conditions (57)–(58) required by Theorem 3.1 hold whenever BFGS updating takes place in a neighborhood of x_*, as shown in Lemma 5.1. Therefore Theorem 3.1 can be applied in the convergence analysis.

The following two lemmas are proved by Xie [29] for very general choices of Y and Z. His result generalizes Lemmas 4.1 and 4.2 of Byrd and Nocedal [6]; see also Powell [27].

LEMMA 5.1. *If Assumptions* 5.1 *hold, then for all x sufficiently near x_**

$$\gamma_1 \|x - x_*\| \leq \|c(x)\| + \|Z(x)^T g(x)\| \leq \gamma_2 \|x - x_*\|, \tag{107}$$

for some positive constants γ_1, γ_2.

This result states that, near x_*, the quantities $c(x)$ and $Z(x)^T g(x)$ may be regarded as a measure of the error at x. The next lemma states that, for a large enough weight, the merit function may also be regarded as a measure of the error.

LEMMA 5.2. *Suppose that Assumptions* 5.1 *hold at x_*. Then for any $\mu > \|\lambda_*\|_\infty$ there exist constants $\gamma_3 > 0$ and $\gamma_4 > 0$, such that for all x sufficiently near x_**

$$\gamma_3 \|x - x_*\|^2 \leq \phi_\mu(x) - \phi_\mu(x_*) \leq \gamma_4 \left[\|Z(x)^T g(x)\|^2 + \|c(x)\|_1\right]. \tag{108}$$

Note that the left inequality in (108) implies that, for a sufficiently large value of the penalty parameter, the merit function will have a strong local minimizer at x_*. We

now use the descent property of Algorithm II to show convergence of the algorithm. However, because of the nonconvexity of the problem, the line search could generate a step that decreases the merit function but that takes us away from the neighborhood of x_*. To rule this out, we make the following assumption.

Assumption 5.2. The line search has the property that, for all large k, $\phi_\mu((1-\theta)x_k + \theta x_{k+1}) \leq \phi_\mu(x_k)$ for all $\theta \in [0,1]$. In other words, x_{k+1} is in the connected component of the level set $\{x : \phi_\mu(x) \leq \phi_\mu(x_k)\}$ that contains x_k.

There is no practical line search algorithm that can guarantee this condition, but it is likely to hold close to x_*. Assumption 5.2 is made by Byrd, Nocedal, and Yuan [8] when analyzing the convergence of variable metric methods for unconstrained problems, as well as by Byrd and Nocedal [6] in the analysis of Coleman–Conn updates for equality constrained optimization.

LEMMA 5.3. *Suppose that the iterates generated by Algorithm* II (*with a line search satisfying Assumption* 5.2) *are contained in a convex region D satisfying Assumptions* 4.1. *If an iterate x_{k_0} is sufficiently close to a solution point x_* that satisfies Assumptions* 5.1, *and if the weight μ_{k_0} is large enough, then the sequence of iterates converges to x_*.*

Proof. By Assumptions 4.1 (I)–(II) and (83) we know that $\{\|\lambda_k\|\}$ is bounded. Therefore the procedure (76) ensures that the weights μ_k are constant, say $\mu_k = \mu$ for all large k. Moreover, if an iterate gets sufficiently close to x_*, we know by (76) and by the continuity of λ that $\mu > \|\lambda_*\|$. For such value of μ, Lemma 5.2 implies that the merit function has a strict local minimizer at x_*. Now suppose that once the penalty parameter has settled, and for a given $\epsilon > 0$, there is an iterate x_{k_0} such that

$$||x_{k_0} - x_*|| \leq \frac{\gamma_3}{\gamma_2\gamma_4\hat{\gamma}_0}\epsilon^2,$$

where $\hat{\gamma}_0$ is such that $\|\cdot\|_1 \leq \hat{\gamma}_0\|\cdot\|$. Assumption 5.2 shows that for any $k \geq k_0$, x_k is in the connected component of the level set of x_{k_0} that contains x_{k_0}, and we can assume that ϵ is small enough that Lemmas 5.1 and 5.2 hold in this level set. Thus since $\phi_\mu(x_k) \leq \phi_\mu(x_{k_0})$ for $k \geq k_0$, and since we can assume that $\|Z_{k_0}^T g_{k_0}\| \leq 1$, we have from Lemmas 5.1 and 5.2, for any $k \geq k_0$

$$\begin{aligned}
||x_k - x_*|| &\leq \gamma_3^{-\frac{1}{2}}(\phi_\mu(x_k) - \phi_\mu(x_*))^{\frac{1}{2}} \\
&\leq \gamma_3^{-\frac{1}{2}}(\phi_\mu(x_{k_0}) - \phi_\mu(x_*))^{\frac{1}{2}} \\
&\leq \left(\frac{\gamma_4}{\gamma_3}\right)^{\frac{1}{2}} \left[\|Z_{k_0}^T g_{k_0}\|^2 + \|c_{k_0}\|_1\right]^{\frac{1}{2}} \\
&\leq \left(\frac{\gamma_4}{\gamma_3}\right)^{\frac{1}{2}} \left[\|Z_{k_0}^T g_{k_0}\|^2 + \hat{\gamma}_0\|c_{k_0}\|\right]^{\frac{1}{2}} \\
&\leq \left(\frac{\gamma_2\gamma_4\hat{\gamma}_0}{\gamma_3}\|x_{k_0} - x_*\|\right)^{\frac{1}{2}} \\
&\leq \epsilon.
\end{aligned}$$

This implies that the whole sequence of iterates remains in a neighborhood of radius ϵ of x_*. If ϵ is small enough, we conclude by (108), by the monotonicity of $\{\phi_\mu(x_k)\}$, and by Theorem 4.2 that the iterates converge to x_*. □

The assumptions of this lemma, which is modeled after a result in Xie [29], are restrictive — especially the assumption on the penalty parameter. One can relax these assumptions and obtain a stronger result, such as Theorem 4.3 in Byrd and Nocedal [6], but the proof would be more complex and is not particularly relevant to Algorithm II since it is based only on the properties of the merit function. Therefore, instead of further analyzing the local convergence properties of the new algorithm, we will study its rate of convergence.

5.1. R-linear convergence. For the rest of the paper we assume that the line search strategy satisfies Assumption 5.2. We also assume that the iterates generated by Algorithm II converge to a point x_* at which Assumptions 5.1 hold, which implies that for all large k, $\mu_k = \mu > \|\lambda_*\|$. The analysis that follows depends on how often BFGS updating is applied. To make this concept precise, we define U to be the set of iterates at which BFGS updating takes place,

$$U = \{k : B_{k+1} = \mathrm{BFGS}(B_k, s_k, y_k)\}, \tag{109}$$

and let

$$U_k = U \cap \{1, 2, \ldots, k\}. \tag{110}$$

The number of elements in U_k will be denoted by $|U_k|$.

THEOREM 5.4. *Suppose that the iterates $\{x_k\}$ generated by Algorithm* II *converge to a point x_* that satisfies Assumptions* 5.1. *Then for any $k \in U$ and any $j \geq k$*

$$\|x_j - x_*\| \leq C r^{|U_k|}, \tag{111}$$

for some constants $C > 0$ and $0 \leq r < 1$.

Proof. Using (89) and (108), we have for $i \in J$,

$$\phi_\mu(x_i) - \phi_\mu(x_{i+1}) \geq \frac{\gamma_\mu}{\gamma_4} \left[\phi_\mu(x_i) - \phi_\mu(x_*)\right]. \tag{112}$$

Let us define $r = (1 - \gamma_\mu/\gamma_4)^{1/4}$. Then for $i \in J$

$$\phi_\mu(x_{i+1}) - \phi_\mu(x_*) \leq r^4 \left[\phi_\mu(x_i) - \phi_\mu(x_*)\right]. \tag{113}$$

We know that the merit function decreases at each step, and by (108) we have, for $j \geq k$ and $k \in U$,

$$\begin{aligned} \|x_j - x_*\| &\leq \gamma_3^{-\frac{1}{2}} (\phi_\mu(x_j) - \phi_\mu(x_*))^{\frac{1}{2}} \\ &\leq \gamma_3^{-\frac{1}{2}} (\phi_\mu(x_k) - \phi_\mu(x_*))^{\frac{1}{2}}. \end{aligned}$$

We continue in this fashion, bounding the right-hand side by terms involving earlier iterates, but using now (113) for all good iterates. Since by Theorem 3.1 at least half of the iterates at which updating takes place are good iterates (i.e., $|J_k| \geq \frac{1}{2}|U_k|$), we have

$$\begin{aligned} \|x_j - x_*\| &\leq \gamma_3^{-\frac{1}{2}} \left[r^{4|J_k|}(\phi_\mu(x_1) - \phi_\mu(x_*))\right]^{\frac{1}{2}} \\ &\leq \gamma_3^{-\frac{1}{2}} \left[r^{2|U_k|}(\phi_\mu(x_1) - \phi_\mu(x_*))\right]^{\frac{1}{2}} \\ &\leq [\gamma_3^{-\frac{1}{2}} (\phi_\mu(x_1) - \phi_\mu(x_*))^{\frac{1}{2}}] r^{|U_k|} \\ &\equiv C r^{|U_k|}. \qquad \square \end{aligned}$$

This result implies that if $\{|U_k|/k\}$ is bounded away from zero, then Algorithm II is R-linearly convergent. However, BFGS updating could take place only a finite number of times, in which case this ratio would converge to zero. It is also possible for BFGS updating to take place an infinite number of times, but every time less often, in such a way that $|U_k|/k \to 0$. We therefore need to examine the iteration more closely.

We make use of the matrix function ψ defined by

$$\psi(B) = \operatorname{tr}(B) - \ln(\det(B)), \tag{114}$$

where tr denotes the trace, and det the determinant. It can be shown that

$$\ln \operatorname{cond}(B) < \psi(B), \tag{115}$$

for any positive definite matrix B (Byrd and Nocedal [5]). We also make use of the weighted quantities

$$\tilde{y}_k = G_*^{-1/2} y_k, \qquad \tilde{s}_k = G_*^{1/2} s_k, \tag{116}$$

$$\tilde{B}_k = G_*^{-1/2} B_k G_*^{-1/2}, \tag{117}$$

$$\cos \tilde{\theta}_k = \frac{\tilde{s}_k^T \tilde{B}_k \tilde{s}_k}{\|\tilde{B}_k \tilde{s}_k\| \|\tilde{s}_k\|}, \tag{118}$$

and

$$\tilde{q}_k = \frac{\tilde{s}_k^T \tilde{B}_k \tilde{s}_k}{\tilde{s}_k^T \tilde{s}_k}. \tag{119}$$

One can show (see Eq. (3.22) of Byrd and Nocedal [5]) that if B_k is updated by the BFGS formula, then

$$\begin{aligned} \psi(\tilde{B}_{k+1}) = \psi(\tilde{B}_k) + \frac{\|\tilde{y}_k\|^2}{\tilde{y}_k^T \tilde{s}_k} - 1 - \ln \frac{\tilde{y}_k^T \tilde{s}_k}{\tilde{s}_k^T \tilde{s}_k} + \ln \cos^2 \tilde{\theta}_k \\ + \left[1 - \frac{\tilde{q}_k}{\cos^2 \tilde{\theta}_k} + \ln \frac{\tilde{q}_k}{\cos^2 \tilde{\theta}_k} \right]. \end{aligned} \tag{120}$$

This expression characterizes the behavior of the BFGS matrices B_k and is crucial to the analysis of this section. Before we can make use of this relation, however, we need to consider the accuracy of the correction terms. We begin by showing that when finite differences are used to estimate w_k and $\overline{w}_k$, these are accurate to second order.

LEMMA 5.5. *If at the iterate x_k, the corrections w_k and $\overline{w}_k$ are computed by the finite-difference formulae* (37)–(38), *and if x_k is sufficiently close to a solution point x_* that satisfies Assumptions* 5.1, *then*

$$w_k = O(\|p_Y\|), \tag{121}$$

$$\|w_k - Z_*^T W_* Y_k p_Y\| = O(\sigma_k \|p_Y\|), \tag{122}$$

and

$$\|\overline{w}_k - \alpha_k Z_*^T W_* Y_k p_Y\| = O(\sigma_k \|p_Y\|). \tag{123}$$

Proof. Recalling that $\nabla L(x,\lambda) = g(x) + A(x)\lambda$, we have from (37) that

$$\begin{aligned} w_k &= Z_k^T[\nabla L(x_k + Y_k p_Y, \lambda_k) - \nabla L(x_k, \lambda_k)] \\ &= Z_k^T[\nabla L(x_k + Y_k p_Y, \lambda_*) - \nabla L(x_k, \lambda_*)] \\ &\quad + Z_k^T[(A(x_k + Y_k p_Y) - A_k)(\lambda_k - \lambda_*)] \\ &= Z_k^T \left[\int_0^1 \nabla_{xx}^2 L(x_k + \tau Y_k p_Y, \lambda_*) d\tau\right] Y_k p_Y \\ &\quad + Z_k^T[(A(x_k + Y_k p_Y) - A_k)(\lambda_k - \lambda_*)] \\ &\equiv Z_k^T \overline{W}_k Y_k p_Y + Z_k^T[(A(x_k + Y_k p_Y) - A_k)(\lambda_k - \lambda_*)]. \end{aligned} \tag{124}$$

Let us assume that x_k is in the neighborhood of x_* where (102)–(105) hold. Then $\|\lambda_k - \lambda_*\| = O(\|e_k\|) = O(\sigma_k)$, where σ_k is defined by (34). Therefore the last term in (124) is $O(\|p_Y\|\sigma_k)$, which proves (121). Also, a simple computation shows that

$$[Z_k^T \overline{W}_k - Z_*^T W_*] Y_k p_Y = O(\sigma_k \|p_Y\|). \tag{125}$$

Using these facts in (124) yields the desired result (122). To prove (123), we note only that $\alpha_k \le 1$ and reason in the same manner. □

Next we show that the condition number of the matrices B_k is bounded and that, in the limit, at the iterates U at which BFGS updating takes place, the matrices B_k are accurate approximations of the reduced Hessian of the Lagrangian.

THEOREM 5.6. *Suppose that the iterates $\{x_k\}$ generated by Algorithm* II *converge to a solution point x_* that satisfies Assumptions* 5.1. *Then $\{\|B_k\|\}$ and $\{\|B_k^{-1}\|\}$ are bounded, and for all $k \in U$*

$$\|(B_k - Z_*^T W_* Z_*) p_Z\| = o(\|d_k\|). \tag{126}$$

Proof. We only consider iterates k for which BFGS updating of B_k takes place. We have from (85), (82), (80), (52), and (84)

$$\begin{aligned} y_k &= Z_k^T[\nabla L(x_{k+1}, \lambda_{k+1}) - \nabla L(x_k, \lambda_{k+1})] - \overline{w}_k \\ &= Z_k^T \left[\int_0^1 \nabla_{xx}^2 L(x_k + \tau\alpha_k d_k, \lambda_{k+1}) d\tau\right] \alpha_k d_k - \overline{w}_k \\ &= \alpha_k Z_k^T \tilde{W}_k (Z_k p_Z + Y_k p_Y) - \overline{w}_k \\ &= Z_k^T \tilde{W}_k Z_k s_k + \alpha_k (Z_k^T \tilde{W}_k - Z_*^T W_*) Y_k p_Y + (\alpha_k Z_*^T W_* Y_k p_Y - \overline{w}_k). \end{aligned} \tag{127}$$

Since $\overline{w}_k$ can be computed by Broyden's method or by finite differences, we need to consider these two cases separately.

Part I. Let us first assume that $\overline{w}_k$ is determined by Broyden's method. A simple computation shows that $\|Z_k^T \tilde{W}_k - Z_*^T W_*\| = O(\sigma_k)$, and from (50) we have that $\overline{w}_k = O(\|p_Y\|/\gamma_k)$. Using this and Assumptions 5.1 in (127), we have

$$\begin{aligned} y_k &= Z_k^T \tilde{W}_k Z_k s_k + (\sigma_k + 1 + 1/\gamma_k) O(\alpha_k \|p_Y\|) \\ &= (Z_k^T \tilde{W}_k Z_k - G_*) s_k + G_* s_k + (\sigma_k + 1 + 1/\gamma_k) O(\alpha_k \|p_Y\|). \end{aligned} \tag{128}$$

Recalling (116) and noting that $\tilde{y}_k^T \tilde{s}_k = y_k^T s_k$, we have

$$\tilde{y}_k^T \tilde{s}_k = s_k^T (Z_k^T \tilde{W}_k Z_k - G_*) s_k + \|\tilde{s}_k\|^2 + (\sigma_k + 1 + 1/\gamma_k) O(\alpha_k \|p_Y\|) \|\tilde{s}_k\|,$$

since $\|\tilde{s}_k\|$ and $\|s_k\|$ are of the same order. Therefore

$$\begin{aligned} \frac{\tilde{y}_k^T \tilde{s}_k}{\|\tilde{s}_k\|^2} &= 1 + \frac{s_k^T (Z_k^T \tilde{W}_k Z_k - G_*) s_k}{\|\tilde{s}_k\|^2} + (\sigma_k + 1 + 1/\gamma_k) O\left(\frac{\|\alpha_k p_Y\|}{\|\tilde{s}_k\|}\right) \\ &= 1 + O(\sigma_k) + (\sigma_k + 1 + 1/\gamma_k) O\left(\frac{\|\alpha_k p_Y\|}{\|\tilde{s}_k\|}\right). \end{aligned} \tag{129}$$

Similarly from (128) and (116) we have

$$\begin{aligned} \tilde{y}_k^T \tilde{y}_k \le{}& \|(Z_k^T \tilde{W}_k Z_k - G_*) s_k\|^2 \|G_*^{-1}\| + 2\|(Z_k^T \tilde{W}_k Z_k - G_*) s_k\| \, \|G_*^{-1/2}\| \, \|\tilde{s}_k\| + \|\tilde{s}_k\|^2 \\ &+ 2(\sigma_k + 1 + 1/\gamma_k) O(\|\alpha_k p_Y\|) \|G_*^{-\frac{1}{2}}\| \left(\|\tilde{s}_k\| + \|(Z_k^T \tilde{W}_k Z_k - G_*) s_k\| \|G_*^{-1/2}\| \right) \\ &+ (\sigma_k + 1 + 1/\gamma_k)^2 O(\|\alpha_k p_Y\|)^2, \end{aligned}$$

and thus

$$\begin{aligned} \frac{\|\tilde{y}_k\|^2}{\|\tilde{s}_k\|^2} \le{}& 1 + O(\sigma_k) + (\sigma_k + 1 + 1/\gamma_k)(1 + \sigma_k) O\left(\frac{\|\alpha_k p_Y\|}{\|\tilde{s}_k\|}\right) \\ &+ (\sigma_k + 1 + 1/\gamma_k)^2 O\left(\frac{\|\alpha_k p_Y\|^2}{\|\tilde{s}_k\|^2}\right). \end{aligned} \tag{130}$$

At this point we invoke the update criterion and note from (54) that, if BFGS updating of B_k takes place at iteration k, then $\|\alpha_k p_Y\| \le \gamma_k^2 \|s_k\|$, where $\{\gamma_k\}$ is summable. Using this, the assumption that σ_k converges to zero, and (129), we see that for large k

$$\frac{\tilde{y}_k^T \tilde{s}_k}{\|\tilde{s}_k\|^2} = 1 + O(\sigma_k + \gamma_k), \tag{131}$$

and using (130)

$$\frac{\|\tilde{y}_k\|^2}{\|\tilde{s}_k\|^2} = 1 + O(\sigma_k + \gamma_k).$$

Therefore

$$\frac{\|\tilde{y}_k\|^2}{\tilde{y}_k^T \tilde{s}_k} = \frac{\|\tilde{y}_k\|^2}{\|\tilde{s}_k\|^2} \frac{\|\tilde{s}_k\|^2}{\tilde{y}_k^T \tilde{s}_k} = 1 + O(\sigma_k + \gamma_k). \tag{132}$$

We now consider $\psi(\tilde{B}_{k+1})$ given by (120). A simple expansion shows that for large k, $\ln(1 + O(\sigma_k + \gamma_k)) = O(\sigma_k + \gamma_k)$. Using this, (131), and (132), we have

$$\psi(\tilde{B}_{k+1}) = \psi(\tilde{B}_k) + O(\sigma_k + \gamma_k) + \ln \cos^2 \tilde{\theta}_k + \left[1 - \frac{\tilde{q}_k}{\cos^2 \tilde{\theta}_k} + \ln \frac{\tilde{q}_k}{\cos^2 \tilde{\theta}_k} \right]. \tag{133}$$

Note that for $x \ge 0$ the function $1 - x + \ln x$ is nonpositive, implying that the term in square brackets is nonpositive and that $\ln \cos^2 \tilde{\theta}_k$ is also nonpositive. We can therefore delete these terms to obtain

$$\psi(\tilde{B}_{k+1}) \le \psi(\tilde{B}_k) + O(\sigma_k + \gamma_k). \tag{134}$$

Before proceeding further we show that a similar expression holds when finite differences are used.

Part II. Let us now consider the iterates k for which updating takes place and for which $\overline{w}_k$ is computed by finite differences. In this case (55) holds. Again we begin by considering (127),

$$y_k = Z_k^T \tilde{W}_k Z_k s_k + \alpha_k (Z_k^T \tilde{W}_k - Z_*^T W_*) Y_k p_Y + (\alpha_k Z_*^T W_* Y_k p_Y - \overline{w}_k).$$

Using (123) the last term is of order $\sigma_k(\alpha_k \|p_Y\|)$, and so is the second term. Thus

$$\begin{aligned} y_k &= Z_k^T \tilde{W}_k Z_k s_k + O(\sigma_k \alpha_k \|p_Y\|) \\ &= (Z_k^T \tilde{W}_k Z_k - G_*) s_k + G_* s_k + O(\sigma_k \alpha_k \|p_Y\|). \end{aligned} \tag{135}$$

Noting once more that $\tilde{y}_k^T \tilde{s}_k = y_k^T s_k$ and recalling the definition (116), we have

$$\tilde{y}_k^T \tilde{s}_k = s_k^T (Z_k^T \tilde{W}_k Z_k - G_*) s_k + \|\tilde{s}_k\|^2 + O(\sigma_k \alpha_k \|p_Y\| \|\tilde{s}_k\|),$$

since $\|\tilde{s}_k\|$ and $\|s_k\|$ are of the same order. Therefore

$$\begin{aligned} \frac{\tilde{y}_k^T \tilde{s}_k}{\|\tilde{s}_k\|^2} &= 1 + \frac{s_k^T (Z_k^T \tilde{W}_k Z_k - G_*) s_k}{\|\tilde{s}_k\|^2} + O\left(\sigma_k \frac{\|\alpha_k p_Y\|}{\|\tilde{s}_k\|}\right) \\ &= 1 + O(\sigma_k) + O\left(\sigma_k \frac{\|\alpha_k p_Y\|}{\|\tilde{s}_k\|}\right). \end{aligned} \tag{136}$$

Similarly from (135) and (116) we have

$$\begin{aligned} \tilde{y}_k^T \tilde{y}_k \le{}& \|(Z_k^T \tilde{W}_k Z_k - G_*) s_k\|^2 \|G_*^{-1}\| + 2\|(Z_k^T \tilde{W}_k Z_k - G_*) s_k\| \, \|G_*^{-1/2}\| \, \|\tilde{s}_k\| + \|\tilde{s}_k\|^2 \\ &+ \sigma_k O\left(\|\alpha_k p_Y\| \|G_*^{-\frac{1}{2}}\| \left[\|\tilde{s}_k\| + \|(Z_k^T \tilde{W}_k Z_k - G_*) s_k\| \|G_*^{-1/2}\|\right]\right) \\ &+ \sigma_k^2 O(\|\alpha_k p_Y\|)^2, \end{aligned}$$

and thus

$$\frac{\|\tilde{y}_k\|^2}{\|\tilde{s}_k\|^2} \le 1 + O(\sigma_k) + \sigma_k O\left(\frac{\|\alpha_k p_Y\|}{\|\tilde{s}_k\|}\right) + \sigma_k^2 O\left(\frac{\|\alpha_k p_Y\|^2}{\|\tilde{s}_k\|^2}\right). \tag{137}$$

We now invoke Update Criterion I and note from (55) that, if BFGS updating of B_k takes place at iteration k, then $\|p_Y\| \le \gamma_{\text{fd}} \|p_Z\| / \sigma_k^{1/2}$. Using this, (136), and the fact that σ_k converges to zero, we see that for large k

$$\frac{\tilde{y}_k^T \tilde{s}_k}{\|\tilde{s}_k\|^2} = 1 + O(\sigma_k^{1/2}),$$

and using (137)

$$\frac{\|\tilde{y}_k\|^2}{\|\tilde{s}_k\|^2} = 1 + O(\sigma_k^{1/2}).$$

Therefore

$$\frac{\|\tilde{y}_k\|^2}{\tilde{y}_k^T \tilde{s}_k} = \frac{\|\tilde{y}_k\|^2}{\|\tilde{s}_k\|^2} \frac{\|\tilde{s}_k\|^2}{\tilde{y}_k^T \tilde{s}_k} = 1 + O(\sigma_k^{1/2}). \tag{138}$$

We now consider $\psi(\tilde{B}_{k+1})$ given by (120). Noting that $\ln(1+O(\sigma_k^{1/2})) = O(\sigma_k^{1/2})$ for all large k, we see that if updating takes place at iteration k

$$\psi(\tilde{B}_{k+1}) = \psi(\tilde{B}_k) + O(\sigma_k^{1/2}) + \ln\cos^2\tilde{\theta}_k + \left[1 - \frac{\tilde{q}_k}{\cos^2\tilde{\theta}_k} + \ln\frac{\tilde{q}_k}{\cos^2\tilde{\theta}_k}\right]. \tag{139}$$

Since both $\ln\cos^2\tilde{\theta}_k$ and the term inside the square brackets are nonpositive, we can delete them to obtain

$$\psi(\tilde{B}_{k+1}) \leq \psi(\tilde{B}_k) + O(\sigma_k^{1/2}). \tag{140}$$

We now combine the results of Parts I and II of this proof. Let us subdivide the set of iterates U for which BFGS updating takes place into two subsets: U' corresponds to the iterates in which $\overline{w}_k$ is computed by Broyden's method, and U'' to the iterates in which finite differences are used. We also define $U'_k = U' \cap \{1,2,\ldots,k\}$ and $U''_k = U'' \cap \{1,2,\ldots,k\}$.

Summing over the set of iterates in U_k, using (134) and (140), and noting that $B_{j+1} = B_j$ for $j \notin U_k$, we have

$$\psi(\tilde{B}_{k+1}) \leq \psi(\tilde{B}_1) + C_1 \sum_{j\in U''_k} \sigma_j^{1/2} + C_2 \sum_{j\in U'_k} \sigma_j + C_3 \sum_{j\in U'_k} \gamma_j, \tag{141}$$

for some constants C_1, C_2, C_3. Since $0 \leq r \leq 1$ and $|U''_j| \leq |U_j|$ we have, from (111)

$$\begin{aligned}\sum_{j\in U''} \sigma_j^{1/2} &\leq \sum_{j\in U''} C^{1/2} r^{|U_j|/2} \\ &\leq \sum_{j\in U''} C^{1/2} r^{|U''_j|/2} \\ &= \sum_{i=1}^{|U''|} C^{1/2} r^{i/2} \\ &< \infty.\end{aligned}$$

Similarly,

$$\sum_{j\in U'} \sigma_j < \infty,$$

and since $\{\gamma_k\}$ is summable, we conclude from (141) that $\{\psi(\tilde{B}_k)\}$ is bounded above. By (114) $\psi(\tilde{B}_k) = \sum_{i=1}^n (l_i - \ln l_i)$, where l_i are the eigenvalues of $\tilde{B}_k$, and it is easy to see that this implies that both $\|B_k\|$ and $\|B_k^{-1}\|$ are bounded.

To prove (126), we sum relations (133) and (139), recalling that σ_k, γ_k and $\sigma_k^{1/2}$ are summable, to obtain

$$\psi(\tilde{B}_{k+1}) \leq C + \sum_{j\in U_k} \left(\ln\cos^2\tilde{\theta}_k + \left[1 - \frac{\tilde{q}_k}{\cos^2\tilde{\theta}_k} + \ln\frac{\tilde{q}_k}{\cos^2\tilde{\theta}_k}\right]\right),$$

for some constant C. Since $\psi(\tilde{B}_{k+1}) > 0$, and since both $\ln\cos^2\theta_k$ and the term inside the square brackets are nonpositive, we see that

$$\lim_{\substack{k\to\infty \\ k\in U}} \ln\cos^2\tilde{\theta}_k = 0$$

and

$$\lim_{\substack{k\to\infty \\ k\in U}} \left[1 - \frac{\tilde{q}_k}{\cos^2\tilde{\theta}_k} + \ln\frac{\tilde{q}_k}{\cos^2\tilde{\theta}_k}\right] \to 0.$$

Now, for $x \geq 0$ the function $1 - x + \ln x$ is concave and has its unique maximizer at $x = 1$. Therefore the relations above imply that

$$\lim_{\substack{k\to\infty \\ k\in U}} \cos\tilde{\theta}_k = \lim_{\substack{k\to\infty \\ k\in U}} \tilde{q}_k = 1. \tag{142}$$

Now from (118)–(119)

$$\begin{aligned}\frac{\|G_*^{-1/2}(B_k - G_*)p_z\|^2}{\|G_*^{1/2}p_z\|^2} &= \frac{\|(\tilde{B}_k - I)\tilde{s}_k\|^2}{\|\tilde{s}_k\|^2} \\ &= \frac{\|\tilde{B}_k\tilde{s}_k\|^2 - 2\tilde{s}_k^T\tilde{B}_k\tilde{s}_k + \tilde{s}_k^T\tilde{s}_k}{\tilde{s}_k^T\tilde{s}_k} \\ &= \frac{\tilde{q}_k^2}{\cos\tilde{\theta}_k^2} - 2\tilde{q}_k + 1.\end{aligned}$$

It is clear from (142) that the last term converges to 0 for $k \in U$, which implies that (126) holds. □

This result immediately implies that the iterates are R-linearly convergent, regardless of how often updating takes place.

THEOREM 5.7. *Suppose that the iterates $\{x_k\}$ generated by Algorithm* II *converge to a solution point x_* that satisfies Assumptions* 5.1 *and the fact that $|U| \to \infty$. Then the rate of convergence is at least* R-*linear.*

Proof. Theorem 5.6 implies that the condition number of the matrices $\{B_k\}$ is bounded. Therefore, *all* the iterates are good iterates. Reasoning as in the proof of Theorem 5.4, we conclude that for all j

$$\|x_j - x_*\| \leq Cr^j,$$

for some constants $C > 0$ and $0 \leq r < 1$. □

Prior to considering the convergence rate, we show that the Broyden matrices S_k are bounded.

LEMMA 5.8. *Suppose that the iterates $\{x_k\}$ generated by Algorithm* II *converge* R-*linearly to a solution point x_* that satisfies Assumptions* 5.1. *Then the Broyden matrices S_k are bounded and the safeguards* (49) *and* (50) *become inactive for all large k.*

Proof. We make use of the well-known bounded deterioration property for Broyden's method (cf. Lemma 8.2.1 in Dennis and Schnabel [15]), which states that under Assumptions 5.1

$$\|S_{k+1} - Z_*^T W_*\| \leq \|S_k - Z_*^T W_*\| + C\sigma_k,$$

for some constant $C > 0$. As a result of the R-linear convergence of $\{x_k\}$, we obtain

$$\begin{aligned}\|S_{k+1} - Z_*^T W_*\| &\leq \|S_1 - Z_*^T W_*\| + C\sum_{i=1}^{k}\sigma_k \\ &< \infty,\end{aligned}$$

which shows that the matrices S_k remain bounded. We then see from (48) that the Broyden corrections w_k and $\overline{w}_k$ satisfy

$$w_k = O(\|p_Y\|) \quad \overline{w}_k = O(\|p_Y\|), \tag{143}$$

and it is clear that safeguards (49) and (50) become inactive for all large k. □

Therefore, the algorithm will not modify the information supplied by Broyden's method, asymptotically. This is an important point in establishing superlinear convergence.

6. Superlinear convergence. Without the correction terms w_k and $\overline{w}_k$, and with appropriate update criteria, Algorithm II is 2-step Q-superlinearly convergent. This was proved by Nocedal and Overton [26] assuming that Y_k and Z_k are orthogonal bases and assuming that a good starting matrix B_1 is used. This result has been extended by Xie [29] for more general bases and for any starting matrix $B_1 > 0$. In this section we show that if the correction terms are used in Algorithm II, the rate of convergence is 1-step Q-superlinear. This result is possible by Update Criterion I and by the selected application of finite-difference approximations, which allow BFGS updating to occur more frequently.

To establish superlinear convergence, we need to ensure that the steplengths α_k have the value 1 for all large k. When a smooth merit function, such as Fletcher's differentiable function (Fletcher [17]) is used, it is not difficult to show that, near the solution, unit steplengths give a sufficient reduction in the merit function and will be accepted. However, the nondifferentiable ℓ_1 merit function (7) used in this paper may reject steplengths of one, even very close to the solution. This so-called Maratos effect requires that the algorithm be modified to allow unit steplengths and to achieve a fast rate of convergence. We do not consider this modification here, so as not to complicate our already lengthy analysis and since it does not affect the main structure of the algorithm or its essential properties. In the companion paper (Biegler, Nocedal, and Schmid [1]), which is devoted to a numerical investigation of Algorithm II, we describe how to incorporate the nonmonotone line search (or watchdog technique) of Chamberlain et al. [9] that allows unit steplengths to be accepted for all large k. The analysis of the modified algorithm would be similar to that presented in §5.5 of Byrd and Nocedal [6].

In the remainder of this section we assume that the iterates generated by Algorithm II converge R-linearly to a solution and that unit steplengths are taken for all large k. In the presentation of the results that follow we do not restate the assumptions under which R-linear convergence was proved in §5, but simply assume that R-linear convergence occurs. We begin by showing that the damping parameter ζ_k, used in (79) to ensure that descent directions are always generated, has the value of 1 for all large k.

We have shown in Theorem 5.6 that $\|B_k^{-1}\|$ is bounded above. Also, (121), (102), and (78) show that, when finite differences are used, $w_k = O(\|p_Y\|) = O(\|c_k\|)$, and by (143) we see that this is also the case when Broyden's method is used. Using these facts, and noting that $\|\cdot\| \leq \|\cdot\|_1$, we see that there is a constant C such that the left-hand side of (74) can be bounded by

$$\zeta_k[2\cos\theta_k|g_k^T Z_k w_k| + w_k^T B_k^{-1} Z_k^T g_k + \zeta_k w_k^T B_k^{-1} w_k] \leq [\zeta_k C(\|e_k\| + \zeta_k\|c_k\|)]\|c_k\|_1,$$

since $g_k^T Z_k = O(\|e_k\|)$. As the iterates converge to the solution, and since $\zeta_k \leq 1$, the term inside the square brackets is less than the constant ρ given in (74), showing that

$\zeta_k = 1$ for all large k. This and the remarks made at the end of §5 show that all the safeguards included in Algorithm II become inactive asymptotically.

We can now show that the Broyden matrices satisfy the condition of Dennis and Moré [14] for superlinear convergence. Note from Algorithm II that a Broyden update of S_k is always performed, regardless of whether a BFGS update of B_k takes place or not. The following result is a straightforward modification of a well-known property for Broyden's method.

LEMMA 6.1. *Suppose that the iterates generated by Algorithm* II *converge* R-*linearly to a point* x_* *that satisfies Assumptions* 5.1. *Then*

$$\lim_{k\to\infty} \frac{\|(S_k - Z_*^T W_*)d_k\|}{\|d_k\|} = 0. \tag{144}$$

Proof. The proof is essentially given in Griewank [21] and is also very similar to the analysis in Dennis and Schnabel [15, pp. 183–184], but we give it here for the sake of completeness. Using the Broyden formula (45), we have

$$\begin{aligned} S_{k+1} - Z_*^T W_* &= S_k - Z_*^T W_* + \frac{(\bar{y}_k - S_k\bar{s}_k)\bar{s}_k^T}{\bar{s}_k^T\bar{s}_k} \\ &= S_k - Z_*^T W_* + \frac{(\bar{y}_k - Z_*^T W_*\bar{s}_k)\bar{s}_k^T}{\bar{s}_k^T\bar{s}_k} + \frac{(Z_*^T W_* - S_k)\bar{s}_k\bar{s}_k^T}{\bar{s}_k^T\bar{s}_k} \\ &= (S_k - Z_*^T W_*)(I - \bar{s}_k\bar{s}_k^T/\bar{s}_k^T\bar{s}_k) + (\bar{y}_k - Z_*^T W_*\bar{s}_k)\bar{s}_k^T/\bar{s}_k^T\bar{s}_k. \end{aligned}$$

Defining $E_k = S_k - Z_*^T W_*$, applying Lemma 8.2.5 of Dennis and Schnabel [15], recalling (46)–(47), and using the mean value theorem, we obtain

$$\begin{aligned} \|E_{k+1}\|_F &\le \|E_k(I - \bar{s}_k\bar{s}_k^T/\bar{s}_k^T\bar{s}_k)\|_F + O(\sigma_k) \\ &\le \|E_k\|_F - \frac{\|E_k\bar{s}_k\|^2}{2\|E_k\|_F\|\bar{s}_k\|^2} + O(\sigma_k). \end{aligned}$$

Rearranging this expression yields

$$\frac{\|E_k\bar{s}_k\|^2}{\|\bar{s}_k\|^2} \le 2\|E_k\|_F\left[\|E_k\|_F - \|E_{k+1}\|_F + O(\sigma_k)\right]. \tag{145}$$

By Lemma 5.8, we know that the matrices S_k remain bounded, therefore there exists some Δ such that for all $k \ge \bar{k}$, $\|E_k\| \le \Delta/2$ and

$$\sum_{k=\bar{k}}^{\infty} \frac{\|E_k\bar{s}_k\|^2}{\|\bar{s}_k\|^2} \le \Delta[\|E_{\bar{k}}\|_F + \sum_{k=\bar{k}}^{\infty} O(\sigma_k)].$$

Since $\{\sigma_k\}$ converges R-linearly, the last term is summable, which implies that

$$\lim_{k\to\infty} \frac{\|E_k\bar{s}_k\|^2}{\|\bar{s}_k\|^2} = 0.$$

Noting that $\bar{s}_k = \alpha_k d_k$ gives the desired result. □

This lemma shows that in the limit S_k is an accurate approximation to $Z_*^T W_*$ along d_k, and Theorem 5.6 shows that, when updating takes place, B_k is an accurate

approximation to $Z_*^T W_* Z_*$ along p_Y. We use these two facts and the following lemma, which is an application of the well-known result of Boggs, Tolle, and Wang [2].

LEMMA 6.2. *Suppose that the iterates generated by Algorithm* II *converge* R-*linearly to a point* x_* *that satisfies Assumptions* 5.1, *and suppose that* $\alpha_k = 1$ *for all large* k. *If, in addition,*

$$\lim_{k\to\infty} \frac{\|B_k p_Z + w_k - Z_*^T W_* d_k\|}{\|d_k\|} = 0, \tag{146}$$

then the rate of convergence is 1-*step* Q-*superlinear.*

Proof. Nocedal and Overton [26, Thm. 3.2] show that if an algorithm of the form

$$\begin{bmatrix} \tilde{S}_k \\ A_k^T \end{bmatrix} d_k = -\begin{bmatrix} Z_k^T g_k \\ c_k \end{bmatrix}, \tag{147}$$

$$x_{k+1} = x_k + d_k,$$

converges to a point x_* that satisfies Assumptions 5.1, and if

$$\lim_{k\to\infty} \frac{\|(\tilde{S}_k - Z_*^T W_*) d_k\|}{\|d_k\|} = 0, \tag{148}$$

then the rate of convergence is superlinear. Algorithm II clearly satisfies the second equation in (147), $A_k^T d_k = -c_k$. Now, since $d_k = Y_k p_Y + Z_k p_Z$, we have

$$[Y_k \; Z_k]^{-1} d_k = \begin{bmatrix} p_Y \\ p_Z \end{bmatrix}. \tag{149}$$

Let us write $w_k = T_k p_Y$ for some matrix T_k. Then, recalling that $\zeta_k = 1$ for all large k, we have from (79) that

$$[T_k \; B_k][Y_k \; Z_k]^{-1} d_k = -Z_k^T g_k.$$

Thus we can define $\tilde{S}_k = [T_k \; B_k][Y_k \; Z_k]^{-1}$, and the condition (148) for superlinear convergence is

$$\lim_{k\to\infty} \frac{\|([T_k \; B_k][Y_k \; Z_k]^{-1} - Z_*^T W_*) d_k\|}{\|d_k\|} = 0.$$

However, using (149) and $w_k = T_k p_Y$, we have that $[T_k \; B_k][Y_k \; Z_k]^{-1} d_k = T_k p_Y + B_k p_Z = w_k + B_k p_Z$, giving the desired result. □

We can now prove the final result of this section. The analysis is complicated by the fact that BFGS updating may not always take place and by the fact that the correction terms are sometimes computed by finite differences and sometimes by Broyden's method. We therefore consider the following three sets of iterates, based on Update Criterion I and illustrated in Fig. 2.

- $R_1 = \{j \mid \|p_Y^{(j)}\| \le \gamma_j^2 \|p_Z^{(j)}\|\}$,
- $R_2 = \{j \notin R_1 \mid \|p_Y^{(j)}\| \le \|p_Z^{(j)}\|/\sigma_j^{1/2}\}$,
- $R_3 = \{j \mid \|p_Y^{(j)}\| > \|p_Z^{(j)}\|/\sigma_j^{1/2}\}$,

and note that both γ_k and σ_k are summable.

THEOREM 6.3. *Suppose that the iterates generated by Algorithm* II *converge* R-*linearly to a point* x_* *that satisfies Assumptions* 5.1, *and suppose that* $\alpha_k = 1$ *for all large* k. *Then the rate of convergence is* 1-*step* Q-*superlinear.*

Proof. Since $d_k = Y_k p_Y + Z_k p_Z$, we have

$$\begin{bmatrix} p_Y \\ p_Z \end{bmatrix} = [Y_k \; Z_k]^{-1} d_k.$$

Therefore, assumption (103) implies that

$$\|p_Y\| = O(\|d_k\|), \qquad \|p_Z\| = O(\|d_k\|). \tag{150}$$

Now

$$\begin{aligned} \|B_k p_Z + w_k - Z_*^T W_* d_k\| &\le \|B_k p_Z - Z_*^T W_* Z_k p_Z\| + \|w_k - Z_*^T W_* Y_k p_Y\| \\ &\le \|B_k p_Z - Z_*^T W_* Z_* p_Z\| + \|w_k - Z_*^T W_* Y_k p_Y\| \\ &\quad + O(\|e_k\| \|p_Z\|). \end{aligned}$$

Since by (150) the last term is of order $o(\|p_Z\|) = o(\|d_k\|)$, the objective of the proof is to show that

$$\|B_k p_Z - Z_*^T W_* Z_* p_Z\| + \|w_k - Z_*^T W_* Y_k p_Y\| = o(\|d_k\|), \tag{151}$$

for this together with (146) will give the desired result. We consider the three regions R_1, R_2, and R_3 separately. Algorithm II is designed so that, in R_2, w_k must be computed by finite differences. On the other hand, since p_Z is recomputed in step 7, after which we can be in any of the three regions, we see that in R_1 and R_3 w_k may be computed by finite differences or by Broyden.

If $k \in R_1$, we have that $\|p_Y\| = o(\|p_Z\|) = o(\|d_k\|)$. We also know from (143) that $w_k = O(\|p_Y\|)$ when the correction is computed by Broyden's method, and by (121) this relation also holds when w_k is computed by finite differences. Therefore, for $k \in R_1$,

$$\|w_k - Z_*^T W_* Y_k p_Y\| = o(\|d_k\|). \tag{152}$$

Furthermore, since updating always takes place in R_1, (126) holds:

$$\|B_k p_Z - Z_*^T W_* Z_* p_Z\| = o(\|d_k\|). \tag{153}$$

We have thus established (151) for all $k \in R_1$.

Let us now suppose that $k \in R_2$, in which case w_k is computed by finite differences. Using (122), we have that

$$\|w_k - Z_*^T W_* Y_k p_Y\| = o(\|p_Y\|) = o(\|d_k\|), \tag{154}$$

where the last step follows from (150). Since updating always takes place in R_2, (153) also holds in this case, and we conclude that (151) holds for all $k \in R_2$.

Finally we consider the case when $k \in R_3$. Now p_Z satisfies

$$p_Z = o(\|p_Y\|) = o(\|d_k\|). \tag{155}$$

If $k \in R_3$ and the correction term w_k is computed by Broyden's method as $w_k = S_k Y_k p_Y$ (see (48)), we have

$$\begin{aligned}\|w_k - Z_*^T W_* Y_k p_Y\| &= \|(S_k - Z_*^T W_*) Y_k p_Y\| \\ &\leq \|(S_k - Z_*^T W_*) d_k\| + \|(S_k - Z_*^T W_*) Z_k p_Z\|.\end{aligned}$$

Using (144), (155), and the boundedness of S_k, we see that the right-hand side is of order $o(\|d_k\|)$, so that (154) holds. On the other hand, if w_k is computed by finite differences, we have directly from (122) that (154) holds. In addition, (155) and the boundedness of B_k show that (153) holds for all $k \in R_3$, regardless of whether finite differences or Broyden's method are used. □

7. Final remarks. We have presented a new reduced Hessian algorithm for large-scale equality-constrained optimization. The motivation for this work has been practical: our earlier reduced Hessian code, designed for large problems, was often subject to instabilities, and we have aimed to develop a more robust algorithm that resembles the full-space SQP method but is less expensive to implement. In a forthcoming paper (Biegler, Nocedal, and Schmid [1]), we discuss our computational experience with the new method. That paper describes how to handle inequality constraints and discusses numerous important details of implementation not considered here. These include the choices of all constants and tolerances, the strategy for coping with the case when the basis matrix C in (35) changes, and the procedure for computing the damping parameter ζ_k, which was only outlined in (75). We also discuss in that paper how to apply the updating criterion away from the solution. We believe that the new algorithm can be very useful for solving large problems, especially those with few degrees of freedom.

We have focused only on convergence results that helped us in the design of the algorithm and that revealed its main properties. The analysis was complicated by two factors. We did not assume that the BFGS matrices B_k or the Broyden matrices S_k were bounded, which required careful consideration of their behavior. This analysis paid off by suggesting safeguards that are useful in practice and ensure a superlinear rate of convergence. The other complicating factor was the fact that the frequency of BFGS updating can vary drastically: it can take place at every iteration, never, or in various patterns. As was found earlier by Xie [29], it is necessary to develop the theory in sufficient generality to cover all of these cases, and this significantly increased the complexity of some of the results.

Acknowledgments. We thank R. Byrd for many interesting discussions on the subject of this paper. We are also thankful to a referee who made very useful suggestions on how to improve the presentation of the results.

REFERENCES

[1] L. T. BIEGLER, J. NOCEDAL, AND C. SCHMID, manuscript, 1993.

[2] P. T. BOGGS, J. W. TOLLE, AND P. WANG, *On the local convergence of a quasi-Newton method for constrained optimization*, SIAM J. Control Optim., 20 (1982), pp. 161–171.

[3] R. H. BYRD, *An example of irregular convergence in some constrained optimization methods that use the projected Hessian*, Math. Programming, 32 (1985), pp. 232–237.

[4] ———, *On the convergence of constrained optimization methods with accurate Hessian information on a subspace*, SIAM J. Numer. Anal., 27 (1990), pp. 141–153.

[5] R. H. BYRD AND J. NOCEDAL, *A tool for the analysis of quasi-Newton methods with application to unconstrained minimization*, SIAM J. Numer. Anal., 26 (1989), pp. 727–739.

[6] R. H. BYRD AND J. NOCEDAL, *An analysis of reduced Hessian methods for constrained optimization*, Math. Programming, 49 (1991), pp. 285–323.
[7] R. H. BYRD, J. NOCEDAL, AND R. B. SCHNABEL, *Representations of quasi-Newton matrices and their application to limited memory methods*, Math. Programming, 63(1994), pp. 129–156.
[8] R. H. BYRD, J. NOCEDAL, AND Y. YUAN, *Global convergence of a class of quasi-Newton methods on convex problems*, SIAM J. Numer. Anal., 24 (1987), pp. 1171–1190.
[9] R. M. CHAMBERLAIN, C. LEMARECHAL, H. C. PEDERSON, AND M. J. D. POWELL, *The watchdog technique for forcing convergence in algorithms for constrained optimization*, Math. Programming Studies, 16 (1982), pp. 1–17.
[10] T. F. COLEMAN AND A. R. CONN, *Nonlinear programming via an exact penalty function: global analysis*, Math. Programming, 24 (1982), pp. 137–161.
[11] ———, *On the local convergence of a quasi-Newton method for the nonlinear programming problem*, SIAM J. Numer. Anal., 21 (1984), pp. 755–769.
[12] T. F. COLEMAN AND P. A. FENYES, *Partitioned quasi-Newton methods for nonlinear equality constrained optimization*, Math. Programming, 53 (1992), pp. 17–44.
[13] A. R. CONN, *Constrained optimization using a nondifferentiable penalty function*, SIAM J. Numer. Anal., 13 (1973), pp. 145–154.
[14] 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.
[15] J. E. DENNIS, JR. AND R. B. SCHNABEL, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1983.
[16] R. FLETCHER, *An exact penalty for nonlinear programming with inequalities*, Math. Programming, 5 (1973), pp. 129–150.
[17] ———, *Practical Methods of Optimization*, 2nd ed., John Wiley and Sons, Chichester, 1987.
[18] D. GABAY, *Reduced quasi-Newton methods with feasibility improvement for nonlinearly constrained optimization*, Math. Programming Studies, 16 (1982), pp. 18–44.
[19] J. C. GILBERT, *On the local and global convergence of a reduced quasi-Newton method*, Optimization, 20 (1989), pp. 421–450.
[20] ———, *Maintaining the positive definiteness of the matrices in reduced Hessian methods for equality constrained optimization*, Math. Programming, 50 (1991), pp. 1–28.
[21] A. GRIEWANK, *The "global" convergence of Broyden-like methods with a suitable line search*, J. Austral. Math. Soc. Ser. B, 28 (1986), pp. 75–92.
[22] C. B. GURWITZ, *Local convergence of a two-piece update of a projected Hessian matrix*, SIAM J. Optim., 4 (1994), pp. 461–485.
[23] C. B. GURWITZ AND M. L. OVERTON, SQP *methods based on approximating a projected Hessian matrix*, SIAM. J. Sci. Statist. Comp., 10 (1989), pp. 631–653.
[24] S. P. HAN, *A globally convergent method for nonlinear programming*, J. Optim. Theory Appl., 22/23 (1977), pp. 297–309.
[25] W. MURRAY AND F. J. PRIETO, *A sequential quadratic programming algorithm using an incomplete solution of the subproblem*, Tech Report, Department of Operations Research, Stanford University, Stanford, CA, 1992.
[26] J. NOCEDAL AND M. L. OVERTON, *Projected Hessian updating algorithms for nonlinearly constrained optimization*, SIAM J. Numer. Anal., 22 (1985), pp. 821–850.
[27] M. J. D. POWELL, *The convergence of variable metric methods for nonlinearly constrained optimization calculations*, in Nonlinear Programming 3, O. Mangasarian, R. Meyer, and S. Robinson, eds., Academic Press, New York, 1978, pp. 27–63.
[28] F. TAGLIAFERRO, *On a quasi-Newton update and its application to equality constrained optimization*, Technical Report, Universita di Trieste, Italy, 1989.
[29] Y. XIE, *Reduced Hessian algorithms for solving large-scale equality constrained optimization problems*, Ph.D. thesis, Department of Computer Science, University of Colorado, Boulder, 1991.
[30] Y. YUAN, *An only* 2*-step* Q*-superlinear convergence example for some algorithms that use reduced Hessian approximations*, Math. Programming, 32 (1985), pp. 224–231.

SIAM J. OPTIMIZATION
Vol. 5, No. 2, pp. 348–378, May 1995

007

A ROBUST TRUST-REGION ALGORITHM WITH A NONMONOTONIC PENALTY PARAMETER SCHEME FOR CONSTRAINED OPTIMIZATION*

MAHMOUD EL-ALEM†

Abstract. An algorithm for solving the problem of minimizing a nonlinear function subject to equality constraints is introduced. This algorithm is a trust-region algorithm. In computing the trial step, a projected-Hessian technique is used that converts the trust-region subproblem to one similar to that for the unconstrained case. To force global convergence, the augmented Lagrangian is employed as a merit function.

One of the main advantages of this algorithm is the way that the penalty parameter is updated. We introduce an updating scheme that allows (for the first time, to the best of our knowledge) the penalty parameter to be decreased whenever it is warranted. The behavior of this penalty parameter is studied.

A convergence theory for this algorithm is presented. It is shown that this algorithm is globally convergent and that the globalization strategy will not disrupt fast local convergence. The local rate of convergence is also discussed. This theory is sufficiently general so that it holds for any algorithm that generates steps whose normal components give at least a fraction of Cauchy decrease in the quadratic model of the constraints and uses Fletcher's exact penalty function as a merit function.

Key words. constrained optimization, global convergence, projected Hessian, penalty parameter, local convergence, trust region, equality constrained

AMS subject classifications. 65K05, 49D37

1. Introduction.

In this paper, we study the following nonlinear equality constrained optimization problem

$$(\text{EQ}) \equiv \begin{cases} \text{minimize} & f(x) \\ \text{subject to} & h(x) = 0, \end{cases}$$

where $h(x) = [h_1(x), \ldots, h_m(x)]^T$. We assume that f and $h_i, i = 1, 2, \ldots, m$ are twice continuously differentiable and that ∇h has full column rank in the range of interest, where $\nabla h(x) = [\nabla h_1(x), \ldots, \nabla h_m(x)]$.

We can obtain first and second order conditions of optimality with reference to the Lagrangian function associated with problem (EQ), namely, $l(x, \lambda) = f(x) + \lambda^T h(x)$, where $\lambda \in \Re^m$ is the Lagrange multiplier vector. The first order necessary condition for a point $x_\star$ to be a stationary point of problem (EQ) is the existence of a Lagrange multiplier $\lambda_\star$ such that $(x_\star, \lambda_\star)$ is a zero of the following $(n+m) \times (n+m)$ nonlinear system of equations

$$\begin{bmatrix} \nabla_x l(x, \lambda) \\ h(x) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{1.1}$$

Consider an $n \times (n-m)$ matrix $Z(x)$, with orthonormal columns that has the property $Z(x)^T \nabla h(x) = 0$. The columns of $Z(x)$ form an orthonormal basis for the null space of $\nabla h(x)^T$. The matrix $Z(x)$ can be obtained from the QR factorization of $\nabla h(x)$ as follows:

$$\nabla h(x) = [\, Y(x) \quad Z(x) \,] \begin{bmatrix} R(x) \\ 0 \end{bmatrix}, \tag{1.2}$$

* Received by the editors October 13, 1992; accepted for publication (in revised form) April 27, 1994. This research was supported by Department of Energy grant DE-fg005-86ER25017, Center for Research on Parallel Computation grant CCR-9120008, and the REDI Foundation.

† Department of Mathematics, Faculty of Science, Alexandria University, Alexandria, Egypt.

where $Y(x) \in \Re^{n\times m}$. The orthonormal columns of $Y(x)$ form a basis for the column space of $\nabla h(x)$ and $R(x)$ is an $m\times m$ nonsingular upper triangular matrix. It is easy to see that $Y(x)^T Y(x) = I_m$, $Z(x)^T Z(x) = I_{n-m}$, and $Y(x)Y(x)^T + Z(x)Z(x)^T = I_n$.

Using this factorization, an equivalent first order necessary condition can be written in the form

$$\begin{bmatrix} Z(x_\star)^T \nabla f(x_\star) \\ h(x_\star) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{1.3}$$

The second order sufficiency condition for the point $x_\star$ to be a solution of problem (EQ) is the existence of a multiplier $\lambda_\star \in \Re^m$ such that the point $(x_\star, \lambda_\star)$ satisfies the first order necessary condition (1.1) and the matrix $Z(x_\star)^T \nabla_x{}^2 l(x_\star, \lambda_\star) Z(x_\star)$ is positive definite.

Throughout this paper, all the norms used are 2-norms and subscripted values of functions are used to denote evaluation at a particular point. For example, f_k means $f(x_k)$, l_k means $l(x_k, \lambda_k)$, and so on.

Some of the algorithms that solve problem (EQ) use Newton's method to find a zero of (1.1). This gives rise to the following $(n+m)\times(n+m)$ linear system:

$$\begin{bmatrix} \nabla_x{}^2 l_k & \nabla h_k \\ \nabla h_k^T & 0 \end{bmatrix} \begin{bmatrix} s_k \\ \Delta\lambda_k \end{bmatrix} = - \begin{bmatrix} \nabla_x l_k \\ h_k \end{bmatrix}. \tag{1.4}$$

If we premultiply the first block of (1.4) by Z_k^T, we obtain the $n\times n$ linear system

$$\begin{bmatrix} Z_k^T \nabla_x{}^2 l_k \\ \nabla h_k^T \end{bmatrix} s_k = - \begin{bmatrix} Z_k^T \nabla f_k \\ h_k \end{bmatrix}. \tag{1.5}$$

Letting $s_k = Y_k u_k + Z_k v_k$ and using the factorization (1.2), the above system becomes

$$\begin{bmatrix} Z_k^T \nabla_x{}^2 l_k Y_k & Z_k^T \nabla_x{}^2 l_k Z_k \\ R_k^T & 0 \end{bmatrix} \begin{bmatrix} u_k \\ v_k \end{bmatrix} = - \begin{bmatrix} Z_k^T \nabla f_k \\ h_k \end{bmatrix}. \tag{1.6}$$

By solving this system of equations for u_k and v_k, we can obtain s_k. More details can be found in Gill and Murray (1974)[9] and Goodman (1985)[10].

The Lagrange multiplier λ_{k+1} is obtained using the least-squares estimate

$$\lambda_{k+1} = \operatorname{argmin} \|\nabla h_{k+1}\lambda + \nabla f_{k+1}\|. \tag{1.7}$$

Using (1.2), this problem is equivalent to solving $R_{k+1}\lambda_{k+1} = -Y_{k+1}^T \nabla f_{k+1}$.

We can proceed by maintaining a quasi-Newton approximation B_k to the Hessian of the Lagrangian $\nabla_x^2 l_k$ in (1.6). More details can be found in Nocedal and Overton (1985)[14]. So, the algorithm for computing the trial step s_k and the multiplier λ_{k+1} is outlined as follows.

ALGORITHM 1.1

At each iteration k, do

Solve $R_k^T u_k = -h_k$, for u_k.

Solve $Z_k^T B_k Z_k v_k = -Z_k^T \nabla f_k - Z_k^T B_k Y_k u_k$, for v_k.

Set $s_k = Y_k u_k + Z_k v_k$ and $x_{k+1} = x_k + s_k$.

Find λ_{k+1} by solving $R_{k+1}\lambda_{k+1} = -Y_{k+1}^T \nabla f_{k+1}$.

End do

It is easy to see that for problem (EQ), if the exact second-order information is used, the above algorithm can be viewed as a Newton's method applied to the nonlinear system (1.1) (see Goodman (1985)[10]). Hence, it shares the advantages and the disadvantages of Newton's method. From the good side of Newton's method, it is locally q-quadratically convergent. However, from the bad side of Newton's method, it is not a globally convergent method. It is guaranteed to converge only if the starting point is close enough to the solution. This means that it may not converge at all if the starting point is far away from the solution. More details can be found in Tapia (1978)[21].

The next section deals with adding a trust-region modification to this method to force convergence to a solution from any starting point without sacrificing fast local convergence.

2. Trust-region globalization. The key idea of the trust-region method is to restrict the trial step to a region where you trust your model. This can be done by imposing the trust-region constraint $\|s_k\| \leq \Delta_k$, where the trust-region radius Δ_k is adjusted automatically from iteration to iteration. The intent is to reduce a merit function $\Phi(x)$ and the aim is to make the iterates $x_{k+1} = x_k + s_k;\ \ k = 1, 2, 3, \ldots$ acceptable points where s_k is obtained by solving some trust-region subproblems. More details about the trust-region method can be found in Dennis and Schnabel (1983)[4].

Byrd, Schnabel, and Shultz (1987)[2] suggested computing the trial steps using the following technique: Set $s_k = Y_k u_k + Z_k v_k$ where Y_k and Z_k are as in (1.2). The two components u_k and v_k are computed by solving two subproblems. For computing u_k, they suggested solving the following linear system:

$$R_k^T u_k = -\alpha_k h_k,$$

where α_k is a constant that satisfies some specified conditions. The tangential component v_k is obtained by solving the trust-region subproblem

$$\begin{aligned} &\underset{v \in \Re^{n-m}}{\text{minimize}}\ (Z_k^T \nabla f_k + \alpha_k Z_k^T \nabla_x^2 l_k Y_k u_k)^T v_k + \tfrac{1}{2} {v_k}^T {Z_k}^T \nabla_x^2 l_k Z_k v_k \\ &\text{subject to } \|v_k\|^2 \leq \Delta_k^2 - \alpha_k^2 \|u_k\|^2. \end{aligned}$$

This approach suffers from the disadvantage that the step depends on the unknown parameter α_k and there is no clear way for choosing this parameter.

An interesting way of using this approach to compute a trial step that does not depend on the parameter α_k was suggested by Omojokun (1989)[15]. He calculated s_k by solving two trust-region subproblems. For computing u_k, he suggested solving

$$\begin{aligned} &\underset{u \in \Re^m}{\text{minimize}}\ \|\nabla {h_k}^T Y_k u_k + h_k\|^2 \\ &\text{subject to } \|Y_k u_k\| \leq \tau \Delta_k, \end{aligned}$$

where $\tau \in (0, 1)$ is a constant. The tangential component is obtained by solving the trust-region subproblem

$$\begin{aligned} &\underset{v \in \Re^{n-m}}{\text{minimize}}\ (Z_k^T \nabla f_k + Z_k^T \nabla_x^2 l_k Y_k u_k)^T v_k + \tfrac{1}{2} {v_k}^T {Z_k}^T \nabla_x^2 l_k Z_k v_k \\ &\text{subject to } \|v_k\|^2 \leq \Delta_k^2 - \|Y_k u_k\|^2. \end{aligned}$$

To force global convergence, Byrd, Schnabel, and Shultz (1987)[2] and Omojokun (1989)[15] employed a nondifferentiable merit function. This type of merit function suffers from the Maratos effect, which may disrupt fast local convergence. See Maratos (1978)[12].

To avoid the Maratos effect, they suggested adding to the step what is called the second-order correction, and is a step of the form $w_k = -R_k^{-T} h_{k_+}$ where k_+ is an intermediate point. See also Coleman and Conn (1982)[3], Fletcher (1982)[7], and Mayne and Polak (1982)[13]. However, this approach adds extra expense to the step calculation since it requires an extra constraint evaluation to compute a trial step.

In this paper, we use an inexpensive way to compute the trial steps. We employ, as a merit function, a differentiable penalty function. We use Fletcher's exact penalty function

$$\Phi(x, \lambda; r) = f(x) + \lambda(x)^T h(x) + r\|h(x)\|^2, \tag{2.1}$$

where λ is the least-squares estimate of the multiplier and r is the penalty parameter. We introduce a new nonmonotonic penalty parameter scheme. This penalty parameter is very inexpensive to calculate.

We present a convergence theory for this algorithm. Our global convergence theory is so general that it covers the algorithm of Byrd, Schnabel, and Shultz (1987)[2] and the algorithm of Omojokun (1989)[15] provided that (2.1) is used as a merit function and Scheme 3.4 (see §3.3) is used for updating the penalty parameter.

The remainder of this paper is organized as follows. In §3, we describe in detail the trust-region subproblems that will be considered and the way of computing the trial steps. A scheme for updating the radius of the trust region is presented together with a discussion about the criteria for accepting or rejecting the trial steps. Our new scheme for updating the penalty parameter is presented in §3 as well as the algorithm. In §4, we state the global assumptions under which we prove global convergence. In §5, we present our global convergence theory. We start with presenting some needed intermediate results together with some lemmas that analyze the behavior of the penalty parameter. We end this section by presenting the main global convergence results of our algorithm. In §6, we present the local convergence analysis. Section 7 contains concluding remarks.

3. The trust-region algorithm. The algorithm has four main ingredients. The first one is computing the trial step. It is discussed in §3.1. The second one is testing the step and updating the trust-region radius and is discussed in §3.2. The third one is updating the penalty parameter and is discussed in §3.3. The fourth ingredient of our algorithm is how to update the matrix B_k. This is discussed at the end of §3.3.

3.1. Computing the trial steps. In our trust-region algorithm, at each iteration, two model subproblems are solved to obtain a trial step s_k. Our way of computing the trial step is similar to that of Byrd, Schnabel, and Schultz (1987)[2] with a simpler way of determining the parameter α_k (see §2). We start by solving for u_k the following linear system of equations:

$$R_k^T u_k = -h_k, \tag{3.1}$$

then we control the size of this step by solving for α_k the one-dimensional minimization problem

$$\begin{aligned} &\underset{\alpha_k \in \Re}{\text{minimize}} \quad \|h_k + \alpha_k \nabla h_k^T Y_k u_k\| \\ &\text{subject to} \quad \alpha_k \|u_k\| \leq \tau \Delta_k, \end{aligned}$$

where $\tau \in (0,1]$ is a fixed constant. This is equivalent to setting

$$\alpha_k = \begin{cases} 1 & \text{if } \|u_k\| \leq \tau\Delta_k \\ \frac{\tau\Delta_k}{\|u_k\|} & \text{if } \|u_k\| > \tau\Delta_k. \end{cases} \tag{3.2}$$

See Zhang and Zhu (1990)[24].

To get the tangential component, we solve for v_k the trust-region subproblem

$$\underset{v\in\Re^{n-m}}{\text{minimize}} \ (Z_k^T\nabla f_k + \alpha_k Z_k^T B_k Y_k u_k)^T v_k + \tfrac{1}{2} {v_k}^T Z_k^T B_k Z_k v_k \tag{3.3}$$

$$\text{subject to } \|v_k\| \leq \Delta_k, \tag{3.4}$$

where B_k is the Hessian of the Lagrangian $\nabla_x^2 l_k$ or an approximation of it.

The trial step then has the form $s_k = \alpha_k Y_k u_k + Z_k v_k$. This is outlined in the following scheme.

SCHEME 3.1. Computing the trial steps.
Given $0 < \tau \leq 1$.
At each iteration k, do
Solve (3.1) for u_k, then find α_k using (3.2).
Solve (3.3) and (3.4) for v_k.
Set $s_k = \alpha_k Y_k u_k + Z_k v_k$ and set $x_{k+1} = x_k + s_k$.
Find λ_{k+1} by solving $R_{k+1}\lambda_{k+1} = -Y_{k+1}^T \nabla f_{k+1}$.
End do.

The Omojokun way of computing the normal component $s_k^n = Y_k u_k$ is more expensive since, to compute u_k, it requires solving a trust-region subproblem at each trial step. Our way requires computing u_k only once per acceptable step, namely, when the algorithm moves to a new point after finding an acceptable step. To compute u_k, we solve (3.1), which is an upper triangular linear system. Y_k and R_k are obtained with no extra cost, since they are obtained from the QR factorization that was performed to compute the multiplier of the last acceptable step.

3.2. Testing the step and updating the trust-region radius. Let $x_{k+1} = x_k + s_k$ where s_k is the step computed by the algorithm, and let λ_{k+1} be the corresponding Lagrange multiplier; we test whether the point (x_{k+1}, λ_{k+1}) is making progress towards a solution $(x_\star, \lambda_\star)$. To do this we use, as a merit function, Fletcher's exact penalty function (2.1). We test (x_{k+1}, λ_{k+1}) to determine whether it makes an improvement in the merit function.

We define the actual reduction in the merit function in moving from (x_k, λ_k) to (x_{k+1}, λ_{k+1}) to be

$$Ared_k = \Phi(x_k, \lambda_k; r_k) - \Phi(x_{k+1}, \lambda_{k+1}; r_k),$$

which can be written as

$$Ared_k = l(x_k, \lambda_k) - l(x_{k+1}, \lambda_k) - (\lambda_{k+1} - \lambda_k)^T h_{k+1} + r_k[\|h_k\|^2 - \|h_{k+1}\|^2].$$

The calculation of the step s_k is based on a quadratic approximation of the Lagrangian function and a linear approximation to the constraints. Using these approximations in a straightforward manner, the predicted reduction has the form

$$\begin{aligned} Pred_k = &-\nabla_x l_k^T s_k - \frac{1}{2} {s_k}^T B_k s_k - (\lambda_{k+1} - \lambda_k - \nabla\lambda_k^T s_k)^T [h_k + \nabla h_k^T s_k] \\ &+ r_k[\|h_k\|^2 - \|h_k + \nabla h_k^T s_k\|^2]. \end{aligned}$$

This form of $Pred_k$ has been used by Maciel (1992)[11]. An undesirable property of using the above expression is that $Pred_k$ depends on $\nabla\lambda_k$, which requires the evaluation of the Hessians of the objective function and the constraints. To avoid these calculations, the following form of predicted reduction can be used:

$$Pred_k = -\nabla_x l_k^T s_k - \frac{1}{2} {s_k}^T B_k s_k - (\lambda_{k+1} - \lambda_k)^T [h_k + \nabla h_k^T s_k] + r_k[\|h_k\|^2 - \|h_k + \nabla h_k^T s_k\|^2].$$

This expression for $Pred_k$ has been used by El-Alem (1988)[5] and (1991)[6]. Our definition of the predicted reduction has the form

$$Pred_k = -\nabla_x l_k^T s_k - \frac{1}{2} {s_k}^T B_k Z_k v_k - (\lambda_{k+1} - \lambda_k)^T \left[h_k + \frac{1}{2}\nabla h_k^T s_k\right] + r_k[\|h_k\|^2 - \|h_k + \nabla h_k^T s_k\|^2].$$

The above expression for $Pred_k$ was also used by Powell and Yuan (1991)[19]. They pointed out that the presence of the terms $\frac{1}{2} s_k^T B_k Z_k v_k$ instead of $\frac{1}{2} s_k^T B_k s_k$ and the term $h_k + \frac{1}{2}\nabla h_k^T s_k$ instead of $h_k + \nabla h_k^T s_k$ allow for a Q-superlinear rate of convergence. See §6 for more details about these terms and how they allow for Q-superlinear rate of convergence.

The normal predicted decrease and the tangential predicted decrease are also considered. They are denoted by $Npred_k$ and $Tpred_k$, respectively. The $Npred_k$ is the decrease at the kth iteration in the linearized model of the constraints by the step $s_k^n = \alpha_k Y_k u_k$ and is defined by

$$Npred_k = \|h_k\|^2 - \|h_k + \alpha_k \nabla h_k^T Y_k u_k\|^2.$$

It predicts the actual reduction in the constraints obtained by the normal component s_k^n.

The $Tpred_k$ is the decrease at the kth iteration in the quadratic model of the Lagrangian by the step $s_k^t = Z_k v_k$. It predicts the actual reduction in the Lagrangian function obtained by the tangential component s_k^t. It is defined by

$$Tpred_k = -(Z_k^T \nabla f_k + Z_k^T B_k s_k^n)^T v_k - \frac{1}{2} v_k^T Z_k^T B_k Z_k v_k \ .$$

The trust-region algorithm should produce steps that result in a decrease in the merit function Φ. To guarantee this, the predicted reduction must be greater than zero and the actual reduction must be greater than some fraction of the predicted reduction. Therefore, at each iteration, the penalty parameter r_k is chosen such that $Pred_k > 0$ and the step is accepted if $\frac{Ared_k}{Pred_k} \geq \eta_1 > 0$, where $\eta_1 \in (0,1)$ is a small fixed constant. We reject the step if $\frac{Ared_k}{Pred_k} < \eta_1$. In this case, we decrease the radius of the trust region by picking $\Delta_k \in [a_1\|s_k\|, a_2\|s_k\|]$, where $0 < a_1 \leq a_2 < \frac{1}{\sqrt{1+\tau^2}}$ and then go back and compute another trial step with a new value of the trust-region radius.

If the step is accepted, then the trust-region radius is updated by comparing the value of $Ared_k$ with $Pred_k$. Namely, if $\eta_1 \leq \frac{Ared_k}{Pred_k} < \eta_2$ where $\eta_2 \in (\eta_1, 1)$, then the radius of the trust region is updated by the rule: $\Delta_{k+1} = \min[\Delta_k, a_3\|s_k\|]$ where $a_3 > \frac{1}{\sqrt{1+\tau^2}}$. However, if $\frac{Ared_k}{Pred_k} \geq \eta_2$, then we increase the radius of the trust region by setting $\Delta_{k+1} = \min[\Delta_\star, \max(\Delta_k, a_3\|s_k\|)]$, where $\Delta_\star$ is a positive constant. This

can be summarized in the following scheme.

SCHEME 3.2. Testing the step and updating the trust-region radius.
Given $0 < a_1 \leq a_2 < \frac{1}{\sqrt{1+\tau^2}} < a_3$, $0 < \eta_1 < \eta_2 < 1$ and $\Delta_\star \geq \Delta_1 > 0$.
At each iteration k, do
If $\frac{Ared_k}{Pred_k} < \eta_1$,
then set $\Delta_k \in [a_1\|s_k\|, a_2\|s_k\|]$.
goto Scheme 3.1 to find another trial step.
Else, if $\eta_1 \leq \frac{Ared_k}{Pred_k} < \eta_2$
then set $x_{k+1} = x_k + s_k$,
$\Delta_{k+1} = \min[\Delta_k, a_3\|s_k\|]$.
Else, set $x_{k+1} = x_k + s_k$,
$\Delta_{k+1} = \min[\Delta_\star, \max(\Delta_k, a_3\|s_k\|)]$.
End if
End if.
End do

The index k is increased only if the step is accepted. We use the notation k^j to denote the jth unacceptable trial step of iteration k.

It is worth noting that, under suitable assumptions, after a finite number of trial steps, an acceptable step will be found, i.e., the condition $\frac{Ared_{kj}}{Pred_{kj}} \geq \eta_1$ will be satisfied for some j. See Theorem 5.7.

3.3. Updating the penalty parameter. Now, we describe our strategy for updating the penalty parameter r. The author in (1988)[5] and (1991)[6] has suggested a scheme for updating the penalty parameter. The idea behind that scheme was to keep the penalty parameter as small as possible subject to satisfying conditions needed to prove global convergence. One of these conditions was that the sequence $\{r_k\}$ of penalty parameter must be nondecreasing. If that scheme were implemented in our problem, the scheme would be as follows.

SCHEME 3.3. El-Alem (1988)[5].
Given a constant $\rho > 0$ and $r_o = 1$.
At each iteration k, do
Set $r_k = r_{k-1}$.
If $Pred_k < \frac{r_{k-1}}{2}[\|h_k\|^2 - \|h_k + \nabla h_k^T s_k\|^2]$,
then set

$$r_k = 2\left\{\frac{\nabla_x l_k^T s_k + \frac{1}{2}s_k^T B_k Z_k v_k + (\lambda_{k+1} - \lambda_k)^T[h_k + \frac{1}{2}\nabla h_k^T s_k]}{\|h_k\|^2 - \|h_k + \nabla h_k^T s_k\|^2}\right\} + \rho\,.$$

End if
End do.

Even though when this scheme was implemented, good performance was reported, (see Williamson (1990)[23]), this way of updating the penalty parameter has the disadvantage of producing a nondecreasing sequence of penalty parameters. This means that if at one iteration the value of the penalty parameter is large, all the subsequent penalty parameters will remain at least as large as this one. Hence, the problem of obtaining feasibility has more weight than the problem of obtaining optimality. As a consequence we may progress too fast toward nonlinear feasibility at the expense of optimality. On the other hand, numerical experiments have suggested that efficient performance of the algorithm is linked to keeping the penalty parameter as small as possible (see Gill et al. (1986)[8]). We propose a scheme that allows (for the first

time, to the best of our knowledge) the penalty parameter to be decreased whenever it is warranted.

Our convergence theory requires that the predicted reduction in the merit function at each iteration be at least as much as a fraction of Cauchy decrease in the 2-norm of the residual of the linearized constraints. (For more detail about the fraction of Cauchy decrease condition see, for example, Powell (1975)[16].) Hence, we ask for this condition to be satisfied at each iteration.

Our convergence theory allows the sequence $\{r_k\}$ to be nonmonotonic, provided that it is controlled by a sequence $\{\underline{\rho}_k\}$, which we introduce below, in the sense that for all k, $\underline{\rho}_{k-1} \leq r_k$.

So, our strategy will be, at each iteration k, pick a number $r_k \geq \underline{\rho}_{k-1}$. Then test for inequality (3.7) (see below) to be satisfied or update the penalty parameter using (3.6) (see below) which enforces (3.7). This scheme is stated as follows.

SCHEME 3.4. Updating the penalty parameter.

Given a constant $\rho > 0$ and an integer $N > 0$;

Set $r_o = r_{-1} = \cdots = r_{-N+1} = 1$

At each iteration k, do

Find $\underline{\rho}_{k-1} = \min\{r_{k-1}, r_{k-2}, \ldots, r_{k-N}\}$,

$\overline{\rho}_{k-1} = \max\{r_{k-1}, r_{k-2}, \ldots, r_{k-N}\}$.

Set

$$\rho_{k-1} = \min\{\ \underline{\rho}_{k-1} + \rho\ ,\ \overline{\rho}_{k-1}\ \}. \tag{3.5}$$

Set $r_k = \rho_{k-1}$.

If

$$Pred_k < \frac{\rho_{k-1}}{2}[\|h_k\|^2 - \|h_k + \nabla h_k^T s_k\|^2],$$

then set

$$r_k = 2\left\{\frac{\nabla_x l_k^T s_k + \frac{1}{2} s_k^T B_k Z_k v_k + (\lambda_{k+1} - \lambda_k)^T [h_k + \frac{1}{2}\nabla h_k^T s_k]}{\|h_k\|^2 - \|h_k + \nabla h_k^T s_k\|^2}\right\} + \rho\ . \tag{3.6}$$

End if

End do.

The following are noteworthy.

1. The way of updating the penalty parameter ensures a predicted decrease in the merit function given by

$$Pred_k \geq \frac{r_k}{2}[\|h_k\|^2 - \|h_k + \nabla h_k^T s_k\|^2]. \tag{3.7}$$

That is, the predicted decrease is at least as much as the decrease in the linearized model of the constraints obtained by the normal component of s_k. So, at each iteration k, we have

$$Pred_k \geq \frac{r_k}{2} Npred_k\ . \tag{3.8}$$

2. If $N = 1$, then Scheme 3.4 will coincide with Scheme 3.3.

3. In the implementation, if we take N equal to the maximum number of iterations allowed, then we will have a scheme for updating the penalty parameter that has no requirement on r_k except that it satisfies inequality (3.7).

4. The sequence $\{\underline{\rho}_k\}$ is a monotonically nondecreasing sequence. (See §5.2 for a proof.) But the sequence $\{\overline{\rho}_k\}$ is a nonmonotonic sequence and only satisfies for all k, $\underline{\rho}_k \leq r_k \leq \overline{\rho}_k$. This inequality shows that even though the sequence $\{r_k\}$ is a nonmonotonic sequence, it is controlled by the two sequences $\{\underline{\rho}_k\}$, $\{\overline{\rho}_k\}$.

5. If at any iteration k we have $\frac{Ared_k}{Pred_k} < \eta_1$, then we reject the trial step and do not increase the iteration count k. As a consequence the set $\{r_{k-1}, \ldots, r_{k-N}\}$ remains unchanged. Thus, implicitly, the value of the penalty parameter is rejected and the only change in the problem by an unacceptable trial step is a decrease in the trust region radius.

Finally, we discuss our strategy for updating the matrix B_k. If the exact Hessian is used, then at each iteration k we compute $\nabla_x^2 l_k = \nabla^2 f_k + \nabla^2 h_k \lambda_k$. Otherwise, update B_k by some updating formula that satisfies global assumption 5 (see §4) if we are interested in obtaining only global convergence regardless of the rate of convergence, or that satisfies global assumption 5 and local assumption C (see §6.3) if we are interested in obtaining global convergence with a fast local rate of convergence.

3.4. Statement of the algorithm. The following is an outline of the algorithm.
Choose $x_0 \in \Re^n$, $\varepsilon > 0$, and $B_0 \in \Re^{n \times n}$.
Set $k = 0$.
At each iteration k, do
  If $\|Z_k^T \nabla f_k\| + \|h_k\| < \varepsilon$, stop.
  Compute s_k, λ_{k+1} according to Scheme 3.1.
  Update the penalty parameter according to Scheme 3.4.
  Test the step and update Δ_k according to Scheme 3.2.
  Update B_k (see §3.3).
  Set $k := k + 1$.
End do.

4. The global assumptions. In this section we state the assumptions under which we prove global convergence.

Let the sequence of iterates generated by the algorithm be $\{x_k\}$. For such a sequence we make the following assumptions.

1. For all k, x_k and $x_k + s_k \in \Omega$ where $\Omega \in \Re^n$ is a convex set.
2. f and $h_i \in C^2(\Omega)$ $i = 1, \ldots, m$.
3. $\nabla h(x)$ has full column rank for all $x \in \Omega$.
4. $f(x)$,$h(x)$,$\nabla h(x)$,$\nabla f(x)$,$\nabla^2 f(x)$, $R(x)^{-1}$ and each $\nabla^2 h_i(x)$, for $i = 1, \ldots, m$ are all uniformly bounded in norm in Ω.
5. The sequence of matrices $\{B_k\}, k = 1, 2, \ldots$ is bounded.

An immediate consequence of the global assumptions is that the matrices B_k, $Z_k^T B_k Z_k$, and $Z_k^T B_k Y_k$ have a uniform upper bound, i.e., there exists a constant $b > 0$, such that, for all k,

$$\|B_k\| \leq b, \quad \|Z_k^T B_k Z_k\| \leq b, \quad \text{and} \quad \|Z_k^T B_k Y_k\| \leq b. \tag{4.1}$$

Another immediate consequence of these assumptions is the existence of constants $b_0 > 0$, $b_1 > 0$, $b_2 > 0$, and $b_3 > 0$ such that, for all k,

$$\|u_k\| \leq b_0 \|h_k\|, \tag{4.2}$$

$$\|\lambda_{k+1} - \lambda_k\| \leq b_1 \|s_k\|, \tag{4.3}$$

$$\|h_{k+1} - h_k\| \leq b_2 \|s_k\|, \tag{4.4}$$

and

$$\|\nabla h_k\| \leq b_3. \tag{4.5}$$

If Ω were a compact set, assumption 4 would follow from the continuity assumption.

The same assumptions as our global assumptions are used by Byrd, Schnabel, and Shultz (1987)[2], El-Alem (1988)[5] and (1991)[6], and Powell and Yuan (1991)[19].

5. Global convergence analysis. In this section we present our global convergence theory. In §5.1, we prove some intermediate lemmas needed for proving global convergence. The behavior of the penalty parameter is discussed in §5.2. Section 5.3 is devoted to proving our main global convergence results.

We start by stating the main global convergence result in order to understand the motivation for the lemmas presented in the next two subsections.

The main global convergence result. Under the global assumptions, the algorithm produces iterates x_k satisfying

$$\liminf_{k\to\infty} [\|h_k\| + \|Z_k^T \nabla f_k\|] = 0.$$

The proof of this result is presented in §5.3.

5.1. Sufficient decrease in the model. All the results in this section deal with the decrease in the model obtained by the trial steps and their tangential and normal components.

The following lemma shows how accurate our definition of predicted reduction in the merit function is as an approximation to the actual reduction. It says that, if the penalty parameter is bounded, it is accurate to within the square of the length of the trial steps.

LEMMA 5.1. *Let the global assumptions hold. Then, for any x_k, $x_k + s_k \in \Omega$, we have*

$$|Ared_k - Pred_k| \leq b_4 r_k \|s_k\|^2, \tag{5.1}$$

where b_4 is a positive constant independent of k.

Proof. The proof is similar to the proof of Corollary 6.4 of El-Alem (1991)[6]. Note that in the proof, inequalities (4.1), (4.2), and the fact that $\|Z_k v_k\| \leq \|s_k\|$ are used. □

The following lemma shows that, at any iteration k, the normal predicted reduction $Npred_k$ is at least equal to the decrease in the 2-norm of the linearized constraints obtained by the Cauchy step, i.e., it satisfies the fraction of Cauchy decrease condition.

LEMMA 5.2. *At any iteration k, we have*

$$Npred_k \geq \|h_k\| \min\left[\|h_k\|, \frac{\tau \Delta_k}{b_0}\right], \tag{5.2}$$

where b_0 is as in (4.2).

Proof. From the definition of $Npred_k$, we need to show that

$$\|h_k\|^2 - \|h_k + \alpha_k \nabla h_k^T Y_k u_k\|^2 \geq \|h_k\| \min\left[\|h_k\|, \frac{\tau \Delta_k}{b_0}\right].$$

When $\|h_k\| = 0$ the above inequality is true a fortiori. Let $\|h_k\| > 0$ and consider

$$\|h_k\|^2 - \|h_k + \alpha_k \nabla h_k^T Y_k u_k\|^2 = \|h_k\|^2 - \|h_k + \alpha_k R_k^T u_k\|^2 = [1 - (1-\alpha_k)^2]\|h_k\|^2.$$

We consider two cases.

First, when $\|u_k\| \leq \tau \Delta_k$. In this case $\alpha_k = 1$ and we obtain

$$\|h_k\|^2 - \|h_k + \alpha_k \nabla h_k^T Y_k u_k\|^2 = \|h_k\|^2.$$

Second, when $\|u_k\| > \tau \Delta_k$, then $\alpha_k = \frac{\tau \Delta_k}{\|u_k\|}$ and using $0 < \alpha_k \leq 1$, we get

$$\|h_k\|^2 - \|h_k + \alpha_k \nabla h_k^T Y_k u_k\|^2 \geq \alpha_k \|h_k\|^2 = \frac{\tau \Delta_k}{\|u_k\|} \|h_k\|^2.$$

Using (4.2), we obtain

$$\|h_k\|^2 - \|h_k + \alpha_k \nabla h_k^T Y_k u_k\|^2 \geq \frac{\tau \Delta_k}{b_0} \|h_k\|.$$

Now, if we combine the two cases, we get the desired result. □

If we substitute (5.2) in (3.8), we obtain

$$Pred_k \geq \frac{r_k}{2} \|h_k\| \min \left[\|h_k\|, \frac{\tau \Delta_k}{b_0} \right]. \tag{5.3}$$

From the last lemma, using (1.2), we can write

$$\|h_k\|^2 - \|h_k + \nabla h_k^T s_k\|^2 \geq \|h_k\| \min \left[\|h_k\|, \frac{\tau \Delta_k}{b_0} \right]. \tag{5.4}$$

The following lemma shows that the tangential predicted decrease is at least equal to the decease in the quadratic model of the Lagrangian obtained by the Cauchy step, i.e., it satisfies the fraction of Cauchy decrease condition.

LEMMA 5.3. *For all k, the tangential predicted reduction satisfies*

$$Tpred_k \geq \frac{1}{4} \|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\| \min \left[\Delta_k, \frac{\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|}{2b} \right], \tag{5.5}$$

where s_k^n is the normal component of the step s_k and b is as in (4.1).

Proof. We first prove that

$$-(Z_k^T \nabla f_k + Z_k^T B_k s_k^n)^T v_k \geq \frac{1}{2} \|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\| \min \left[\Delta_k, \frac{\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|}{2\|Z_k^T B_k Z_k\|} \right].$$

When $\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\| = 0$ the above inequality is valid a fortiori.

Let $\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\| > 0$. If $\|v_k\| < \Delta_k$, then from the way of computing v_k, we have $Z_k^T B_k Z_k v_k + Z_k^T \nabla f_k + Z_k^T B_k s_k^n = 0$, and we can write

$$\begin{aligned}(Z_k^T \nabla f_k + Z_k^T B_k s_k^n)^T v_k &= -v_k^T Z_k^T B_k Z_k v_k \\ &= -(Z_k^T \nabla f_k + Z_k^T B_k s_k^n)^T (Z_k^T B_k Z_k)^+ (Z_k^T \nabla f_k + Z_k^T B_k s_k^n),\end{aligned}$$

where $(Z_k^T B_k Z_k)^+$ is the generalized inverse of $Z_k^T B_k Z_k$. We have

$$(Z_k^T \nabla f_k + Z_k^T B_k s_k^n)^T v_k \leq -\frac{1}{\|Z_k^T B_k Z_k\|} \|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|^2. \tag{5.6}$$

On the other hand, if $\|v_k\| = \Delta_k$, then from the way of computing v_k, there exists a constant $\mu_k \geq 0$ such that

$$(Z_k^T B_k Z_k + \mu_k I)v_k + Z_k^T \nabla f_k + Z_k^T B_k s_k^n = 0. \tag{5.7}$$

This equation implies that

$$\begin{aligned}(Z_k^T \nabla f_k + Z_k^T B_k s_k^n)^T v_k &= -v_k^T (Z_k^T B_k Z_k + \mu_k I)v_k \\ &= -(Z_k^T \nabla f_k + Z_k^T B_k s_k^n)^T (Z_k^T B_k Z_k + \mu_k I)^+ (Z_k^T \nabla f_k + Z_k^T B_k s_k^n),\end{aligned}$$

which implies that

$$(Z_k^T \nabla f_k + Z_k^T B_k s_k^n)^T v_k \leq -\frac{1}{\overline{\lambda}(Z_k^T B_k Z_k) + \mu_k} \|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|^2, \tag{5.8}$$

where $\overline{\lambda}(Z_k^T B_k Z_k)$ is the largest eigenvalue of $Z_k^T B_k Z_k$. On the other hand, from (5.7), we have

$$[\underline{\lambda}(Z_k^T B_k Z_k) + \mu_k]\|v_k\| \leq \|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|,$$

where $\underline{\lambda}(Z_k^T B_k Z_k)$ is the smallest eigenvalue of $Z_k^T B_k Z_k$. The above inequality implies that

$$\mu_k \leq \frac{\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|}{\Delta_k} - \underline{\lambda}(Z_k^T B_k Z_k). \tag{5.9}$$

By substituting (5.9) in (5.8), we obtain

$$(Z_k^T \nabla f_k + Z_k^T B_k s_k^n)^T v_k \leq -\frac{\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|^2 \Delta_k}{[\overline{\lambda}(Z_k^T B_k Z_k) - \underline{\lambda}(Z_k^T B_k Z_k)]\Delta_k + \|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|}.$$

Now, using the fact that $\overline{\lambda}(Z_k^T B_k Z_k) - \underline{\lambda}(Z_k^T B_k Z_k) \leq 2\|Z_k^T B_k Z_k\|$, the above inequality becomes

$$(Z_k^T \nabla f_k + Z_k^T B_k s_k^n)^T v_k \leq -\frac{\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|^2 \Delta_k}{2\|Z_k^T B_k Z_k\|\Delta_k + \|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|}. \tag{5.10}$$

So, from (5.6) and (5.10), we conclude that in both cases we can write

$$-(Z_k^T \nabla f_k + Z_k^T B_k s_k^n)^T v_k \geq \frac{1}{2}\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\| \min\left[\Delta_k, \frac{\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|}{2\|Z_k^T B_k Z_k\|}\right].$$

The rest of the proof follows directly from the definition of $Tpred_k$, the fact that $(Z_k^T \nabla f_k + Z_k^T B_k s_k^n)^T v_k + v_k^T Z_k^T B_k Z_k v_k \leq 0$, and (4.1). □

LEMMA 5.4. *Let s_k be the step generated by the algorithm at the kth iteration, then*

$$Pred_k \geq Tpred_k - b_5\|s_k\|\|h_k\| + \frac{r_k}{2} Npred_k,$$

where b_5 is a positive constant independent of k.

Proof. From the definition of $Pred_k$, we have

$$\begin{aligned}Pred_k = &-(Z_k^T \nabla f_k)^T v_k - \frac{1}{2} s_k^T B_k Z_k v_k - (\lambda_{k+1} - \lambda_k)^T \left[h_k + \frac{1}{2}\nabla h_k^T s_k\right] \\ &+ r_k[\|h_k\|^2 - \|h_k + \nabla h_k^T s_k\|^2].\end{aligned}$$

This can be written as

$$\begin{aligned} Pred_k &= -(Z_k^T\nabla f_k + Z_k^T B_k s_k^n)^T v_k - \frac{1}{2}v_k^T Z_k^T B_k Z_k v_k + \frac{1}{2}\alpha_k u_k^T Y_k^T B_k Z_k v_k \\ &\quad -(\lambda_{k+1}-\lambda_k)^T\left[h_k + \frac{1}{2}\nabla h_k^T s_k\right] + r_k[\|h_k\|^2 - \|h_k + \nabla h_k^T s_k\|^2] \\ &\geq Tpred_k - \|\lambda_{k+1}-\lambda_k\|\|h_k + \frac{1}{2}\nabla h_k^T s_k\| - \|Y_k^T B_k Z_k\|\|v_k\|\|u_k\| \\ &\quad + \frac{r_k}{2}Npred_k. \end{aligned}$$

Using (4.1)–(4.3), the fact that $\|h_k + \frac{1}{2}\nabla h^T s_k\| \leq \|h_k\|$, and $\|v_k\| \leq \|s_k\|$, the remainder of the proof follows immediately. □

The following lemma proves that if $\|h_k\|$ is small enough, then the penalty parameter will not be updated using (3.6), i.e., inequality (3.7) will hold for $r_k = \rho_{k-1}$. (See Scheme 3.4).

LEMMA 5.5. *Let k be the index of an iteration at which the algorithm does not terminate. If $\|h_k\| \leq c_1\Delta_k$ where c_1 is a small constant that satisfies*

$$c_1 \leq \min\left\{\frac{\varepsilon}{3\Delta_\star}, \frac{\varepsilon}{3bb_0\Delta_\star}, \frac{\varepsilon}{24\sqrt{2}b_5\Delta_\star}\min\left(1, \frac{\varepsilon}{6b\Delta_\star}\right)\right\}, \tag{5.11}$$

then

$$Pred_k \geq \frac{1}{2}Tpred_k + \frac{r_k}{2}Npred_k. \tag{5.12}$$

Proof. From Lemmas 5.4 and 5.3, we can write

$$\begin{aligned} Pred_k &\geq \frac{1}{2}Tpred_k + \frac{1}{8}\|Z_k^T\nabla f_k + Z_k^T B_k s_k^n\|\min\left[\Delta_k, \frac{\|Z_k^T\nabla f_k + Z_k^T B_k s_k^n\|}{2b}\right] \\ &\quad -b_5\|s_k\|\|h_k\| + \frac{r_k}{2}Npred_k. \end{aligned} \tag{5.13}$$

Since $c_1 \leq \frac{\varepsilon}{3\Delta_\star}$, then $\|h_k\| \leq \frac{\varepsilon}{3}$ and because the algorithm does not terminate, $\|Z_k^T\nabla f_k\| > \frac{2\varepsilon}{3}$, and we obtain

$$\begin{aligned} \|Z_k^T\nabla f_k + Z_k^T B_k s_k^n\| &\geq \|Z_k^T\nabla f_k\| - \|Z_k^T B_k Y_k\|\|u_k\|, \\ &\geq \frac{2\varepsilon}{3} - bb_0\|h_k\| \geq \frac{2\varepsilon}{3} - \frac{\varepsilon}{3} = \frac{\varepsilon}{3}. \end{aligned}$$

Hence, using $\|s_k\| \leq \sqrt{2}\Delta_k$, we have

$$\begin{aligned} Pred_k &\geq \frac{1}{2}Tpred_k + \frac{\varepsilon\Delta_k}{24}\min\left[1, \frac{\varepsilon}{6b\Delta_\star}\right] - \sqrt{2}c_1 b_5\Delta_k^2 + \frac{r_k}{2}Npred_k, \\ &\geq \frac{1}{2}Tpred_k + \left\{\frac{\varepsilon}{24}\min[1, \left[\frac{\varepsilon}{6b\Delta_\star}\right] - \sqrt{2}c1b_5\Delta_\star\right\}\Delta_k + \frac{r_k}{2}Npred_k. \end{aligned}$$

From (5.11), the quantity $\{\frac{\varepsilon}{24}\min[1, \frac{\varepsilon}{6b\Delta_\star}] - \sqrt{2}c_1 b_5\Delta_\star\}$ is positive. Hence,

$$Pred_k \geq \frac{1}{2}Tpred_k + \frac{r_k}{2}Npred_k,$$

which is the desired result. □

From the proof of the above lemma, we see that the fourth term in (5.13) did not enter into the calculation. This implies that if we set $r_k = \rho_{k-1}$ (see Scheme 3.4) inequality (5.12) remains valid. So, when $\|h_k\| \leq c_1\Delta_k$, the algorithm will not update the penalty parameter using (3.6). In other words, inequality (3.7) will always be satisfied.

LEMMA 5.6. *If the algorithm does not terminate, then any iteration at which* $\|h_k\| \leq c_1\Delta_k$, *satisfies*

$$Pred_k \geq c_2\Delta_k, \tag{5.14}$$

where c_1 *is given by* (5.11) *and* c_2 *is a positive constant independent of* k.

Proof. When $\|h_k\| \leq c_1\Delta_k$, where c_1 is given by (5.11), then from Lemmas 5.3 and 5.5, we have

$$Pred_k \geq \frac{1}{2}Tpred_k \geq \frac{1}{8}\|Z_k^T\nabla f_k + Z_k^T B_k s_k^n\|\min\left[\Delta_k, \frac{\|Z_k^T\nabla f_k + Z_k^T B_k s_k^n\|}{2b}\right].$$

But, since $c_1 < \frac{\varepsilon}{3\Delta_\star}$, then $\|h_k\| < \frac{\varepsilon}{3}$, and because the algorithm does not terminate, we have $\|Z_k^T\nabla f_k\| \geq \frac{2\varepsilon}{3}$. Thus, as in Lemma 5.5, we conclude that $\|Z_k^T\nabla f_k + Z_k^T B_k s_k^n\| \geq \frac{\varepsilon}{3}$. Hence,

$$Pred_k \geq \frac{\varepsilon}{24}\min\left[1, \frac{\varepsilon}{6b\Delta_\star}\right]\Delta_k.$$

The result now follows if we set

$$c_2 = \frac{\varepsilon}{24}\min[1, \frac{\varepsilon}{6b\Delta_\star}]. \qquad \square$$

The following theorem shows that the algorithm is well defined in the sense that it will never loop ad infinitum without finding an acceptable step.

THEOREM 5.7. *Let the global assumptions hold. At any iteration* k *at which the penalty parameter* r_k *is bounded, either the termination condition of the algorithm will be met or an acceptable step will be found.*

Proof. In the proof of this lemma we use the notation k^j to mean the jth unacceptable trial step of iteration k.

If the termination condition of the algorithm is satisfied, then there is nothing to prove. Assume that the point (x_k, λ_k) does not satisfy the termination condition of the algorithm.

Suppose that at iteration k the algorithm loops infinitely without finding an acceptable step. Hence all the trial steps are rejected and we obtain, for all j

$$(1 - \eta_1) < \left|\frac{Ared_{k^j}}{Pred_{k^j}} - 1\right|. \tag{5.15}$$

First, assume that $\|h_k\| = 0$. Therefore, for all j we have $\|h_{k^j}\| \leq c_1\Delta_{k^j}$, where c_1 is as in (5.11). In this case the penalty parameter remains the same. So, we have $r_{k^j} = r_k$ is bounded for all j.

On the other hand, from Lemmas 5.1 and 5.6, for any j such that $\Delta_{k^j} > 0$, we have

$$\left|\frac{Ared_{k^j}}{Pred_{k^j}} - 1\right| = \frac{|Ared_{k^j} - Pred_{k^j}|}{Pred_{k^j}} \leq \frac{b_4 r_k}{c_2}\Delta_{k^j}.$$

As j goes to infinity, Δ_{k^j} goes to zero and we get a contradiction with (5.15). So j cannot go to infinity. But this contradicts the supposition that the algorithm loops infinitely without finding an acceptable step and means that, after finitely many rejected trial steps, an acceptable one will be found.

Now assume that $\|h_k\| > 0$. From (3.7), (5.2), and Lemma 5.1, we can write

$$\left|\frac{Ared_{k^j}}{Pred_{k^j}} - 1\right| = \frac{|Ared_{k^j} - Pred_{k^j}|}{Pred_{k^j}} \leq \frac{2b_4\Delta_{k^j}^2}{\|h_{k^j}\| \min\{\|h_{k^j}\|, \frac{\tau\Delta_{k^j}}{b_0}\}}.$$

Here $\|h_{k^j}\| = \|h_k\| > 0$ is fixed. Therefore, for sufficiently large j, we have,

$$\min\left\{\|h_{k^j}\|, \frac{\tau\Delta_{k^j}}{b_0}\right\} = \frac{\tau\Delta_{k^j}}{b_0}.$$

Hence,

$$\left|\frac{Ared_{k^j}}{Pred_{k^j}} - 1\right| \leq \frac{2b_4b_0}{\tau\|h_k\|}\Delta_{k^j}.$$

As j goes to infinity, Δ_{k^j} goes to zero and we get a contradiction with (5.15). So j cannot go to infinity. Again this contradicts the supposition. Hence the supposition is wrong and the theorem is proved. □

Under the assumption that the algorithm does not terminate, the above theorem is true at any iteration k at which r_k is bounded. In the following section we prove that the penalty parameter is bounded for all k. This implies that Theorem 5.7 is true for all k.

5.2. The behavior of the penalty parameter. In our analysis of the penalty parameter, the sequences $\{\underline{\rho}_k\}$ and $\{\overline{\rho}_k\}$ are used. For their definitions see Scheme 3.4.

Our goal is to prove that there exists a constant $r_\star$ and an integer $\overline{k}$ such that $r_k = r_\star$ for all $k \geq \overline{k}$. To this end, we will prove the following. First we prove that $\{\overline{\rho}_k\}$ is bounded. This of course implies that $\{r_k\}$ and $\{\underline{\rho}_k\}$ are bounded. Second we show that $\{\underline{\rho}_k\}$ is a nondecreasing sequence. We also discuss the amount of increase in the sequence $\{\underline{\rho}_k\}$. Finally we show that the sequences $\{\underline{\rho}_k\}$, $\{r_k\}$, $\{\overline{\rho}_k\}$ attain the same value after finitely many iterations. We start with the following lemma which we use to conclude that r_k is bounded.

LEMMA 5.8. *Under the global assumptions, the sequence* $\{\overline{\rho}_k\}$ *is bounded.*

Proof. If the algorithm terminates, $\{\overline{\rho}_k\}$ is finite and trivially bounded. So, consider the case when the algorithm does not terminate. The proof is by contradiction. Suppose that the sequence $\{\overline{\rho}_k\}$ is not bounded. Then there exists an infinite sequence of indices $\{k_i\}$, such that

$$\overline{\rho}_k > \max\left\{\frac{\sqrt{2}b_o(2b_1 + b_0b + 2\rho b_3)}{\min(\tau, c_1b_0)}, \overline{\rho}_1\right\} \tag{5.16}$$

for all $k \in \{k_i\}$. Suppose that m is the first index such that (5.16) holds. It is clear, using inequality (5.16), that $m \geq 2$.

The only possibility that $\overline{\rho}_m > \overline{\rho}_{m-1}$ is when $r_m > \overline{\rho}_{m-1}$ and this can only happen when r_m is updated by (3.6). This implies that

$$r_m[\|h_m\|^2 - \|h_m + \nabla h_m^T s_m\|^2] = 2(Z_m^T\nabla f_m)^T v_m + s_m^T B_m Z_m v_m$$

$$+ 2(\lambda_{m+1} - \lambda_m)^T(h_m + \frac{1}{2}\nabla h_m^T s_m)$$
$$+ \rho\,[\|h_m\|^2 - \|h_m + \nabla h_m^T s_m\|^2].$$

Using (5.4) and the fact that $2(Z_m^T\nabla f_m + Z_m^T B_m s_m^n)^T v_m + v_m^T Z_m^T B_m Z_m v_m \le 0$, we can write

$$r_m\|h_m\|\min\left[\frac{\tau\Delta_m}{b_0}, \|h_m\|\right] \le 2\|\lambda_{m+1} - \lambda_m\|\left\|h_m + \frac{1}{2}\nabla h_m^T s_m\right\|$$
$$+ \|Y_m^T B_m Z_m\|\|u_m\|\|v_m\| - 2\rho\, h_m^T\nabla h_m^T s_m.$$

Using (4.1)–(4.3) and (4.5) and the fact that $\|h_k + \frac{1}{2}\nabla h_k^T s_k\| \le \|h_k\|$, we can write

$$r_m\|h_m\|\min\left[\frac{\tau\Delta_m}{b_0}, \|h_m\|\right] \le (2b_1 + bb_0 + 2\rho b_3)\|h_m\|\|s_m\|.$$

If we use the above inequality, together with the fact that r_k is updated by (3.6) only when $\|h_k\| > c_1\Delta_k$, we obtain

$$r_m \le \frac{\sqrt{2}b_0(2b_1 + bb_0 + 2\rho b_3)}{\min(\tau, c_1 b_0)}.$$

This result, together with the fact that $\overline{\rho}_{m-1}$ does not satisfy (5.16) implies that $\overline{\rho}_m$does not satisfy (5.16). This contradicts the supposition that m is the first index such that (5.16) is satisfied and means that there is no such m. Hence the sequence $\overline{\rho}_k$ is bounded. □

From the definition of $\{\overline{\rho}_k\}$ and the last lemma, it follows directly that the sequences $\{r_k\}$ and $\{\underline{\rho}_k\}$ are bounded.

LEMMA 5.9. *The sequence* $\{\underline{\rho}_k\}$ *is monotonically nondecreasing.*

Proof. From the way of updating the penalty parameter r_k we always have, for all k, $\underline{\rho}_{k-1} \le r_k$ and since $\underline{\rho}_k = \min\{r_k, r_{k-1}, \ldots, r_{k-N+1}\}$, then we must have $\underline{\rho}_{k-1} \le \underline{\rho}_k$, which means that the sequence $\{\underline{\rho}_k\}$ is monotonically nondecreasing. □

Now we argue that $\{\underline{\rho}_k\}$ will increase in a finite number of iterations until it reaches its upper bound. In other words, there exists an integer $\hat{k}$ such that $\underline{\rho}_k = \underline{\rho}_{\hat{k}}$ for all $k \ge \hat{k}$.

First of all, we study the possible increase in r_k over $\underline{\rho}_{k-1}$. In other words, if there is an increase in r_k over $\underline{\rho}_{k-1}$, how much is this increase? If r_k is increased over $\underline{\rho}_{k-1}$, it will increase through one of the following three possibilities:

1. It will be increased by at least ρ if it is increased according to (3.6) regardless of the result in (3.5) of Scheme 3.4.

2. It will be increased by at least ρ if $\underline{\rho}_{k-1} + \rho \le \overline{\rho}_{k-1}$ regardless of the result in the "if" statement of Scheme 3.4.

3. It will be increased by at least $(\overline{\rho}_{k-1} - \underline{\rho}_{k-1})$ if $\underline{\rho}_{k-1} < \overline{\rho}_{k-1}$, but $\underline{\rho}_{k-1} + \rho > \overline{\rho}_{k-1}$.

Notice that the amount $(\overline{\rho}_{k-1} - \underline{\rho}_{k-1})$ can be very small so that, if at each iteration the penalty parameter increases by this amount, it seems that the algorithm may take infinitely many iterations without $\{\underline{\rho}_k\}$ reaching its upper bound. Later on we show that this situation cannot happen.

Also, we notice that, for $\underline{\rho}_{k-1} < \overline{\rho}_{k-1}$ we always have $\underline{\rho}_{k-1} < r_k$, which means a possible increase in $\underline{\rho}_{k-1}$ to $\underline{\rho}_k$.

Finally, we notice that, the only possibility that $r_k = \underline{\rho}_{k-1}$ is when $\underline{\rho}_{k-1} = \overline{\rho}_{k-1}$ and $\underline{\rho}_{k-1}$ satisfies (3.7). In this case $\underline{\rho}_{k-1} = r_k = \overline{\rho}_{k-1}$ which implies that $\underline{\rho}_k = r_k = \overline{\rho}_k$.

Define the following three sets of indices:

$$I = \{\ k\ :\ \underline{\rho}_{k-1} + \rho \leq \overline{\rho}_{k-1}\ \}.$$
$$J = \{\ k\ :\ \underline{\rho}_{k-1} < \overline{\rho}_{k-1} \text{ but } \underline{\rho}_{k-1} + \rho > \overline{\rho}_{k-1}\ \}.$$
$$K = \{\ k\ :\ \underline{\rho}_{k-1} = \overline{\rho}_{k-1}\ \}.$$

The following propositions can be easily verified.

PROPOSITION 1. *If $k+1 \in I$ then $\underline{\rho}_{k+N} \geq \underline{\rho}_k + \rho$.*

PROPOSITION 2. *If $k \in K$ then either $k+1 \in K$, or $k+1, \ldots, k+N-1 \in I$.*

PROPOSITION 3. *If $k \in J$ then either $k+1 \in J$, or $k+1 \in K$, or $k+1, \ldots, k+N-1 \in I$.*

PROPOSITION 4. *If $k, k+1, \ldots, k+N-1 \in J$, then $k+N \in K$, or $k+N, \ldots, k+2N-2 \in I$.*

It is easy to see that (in the worst case) every $2N-1$ consecutive iteration at which the sequence $\{\underline{\rho}_k\}$ increases, will increase by at least ρ. Thus, because $\{\underline{\rho}_k\}$ is bounded, the sequence $\{\underline{\rho}_k\}$ will take only a finite number of iterations to attain its upper bound.

LEMMA 5.10. *If the algorithm does not terminate, then there exists a positive integer k_2 and a constant $r_\star > 0$ such that, for all $k \geq k_2$, $r_k = r_\star$.*

Proof. We notice that, because of Lemma 5.8, after finite number of iterations k_1 inequality (3.7) will be satisfied for all $k \geq k_1$. This implies that there exists an integer $k_2 > k_1$ such that $\underline{\rho}_k = \overline{\rho}_k$ for all $k \geq k_2$. However, from the way of updating r_k, this will imply that $\underline{\rho}_k = r_k = \overline{\rho}_k$ for all $k \geq k_2$. This implies $r_k = r_\star$ for all $k \geq k_2$. □

5.3. The main global results. We show that the algorithm always terminates. This is shown in two steps. First, it is shown that if the algorithm would not terminate, then $\lim_{k\to\infty} \|h_k\| = 0$. Second, it is shown that if the algorithm would not terminate, then $\liminf_{k\to\infty} \|Z_k^T \nabla f_k\| = 0$. Thus for every $\epsilon > 0$ there exists an integer k_0 such that $\|h_{k_0}\| + \|Z_{k_0}^T \nabla f_{k_0}\| < \epsilon$.

The following lemma is crucial in proving that the algorithm will converge to a feasible point. Intuitively speaking, it shows that the trust region will not collapse to a point as long as $\|h_k\|$ is bounded away from zero.

LEMMA 5.11. *Let the global assumptions hold. If the sequence of iterates generated by the algorithm is bounded away from the feasible region, i.e., $\|h_k\| > \varepsilon_0$, for some fixed positive constant ε_0 and all k, then there exists a constant $c_3 > 0$, such that, for all k*

$$\Delta_k \geq c_3. \tag{5.17}$$

Proof. The proof is by contradiction. Suppose that $\{\Delta_k\}$ is not bounded away from zero, then there exists a sequence of indices $\{k_j\}$ such that

$$\Delta_k < \frac{a_1 b_0 \sigma_1}{\tau}(1-\eta_2), \tag{5.18}$$

for all $k \in \{k_j\}$, where

$$\sigma_1 = \min\left\{\varepsilon_0, \frac{\tau\Delta_1}{a_1 b_0(1-\eta_2)}, \frac{\tau^2\varepsilon_0}{2\sqrt{2}b_4 b_0^2}\right\}.$$

Let m be the first integer such that (5.18) holds. It is clear from the definition of σ_1 that

$$\sigma_1 \le \frac{\tau\Delta_1}{a_1 b_0(1-\eta_2)}$$

which implies that $m \ge 2$.

Using (5.18) and the way of updating Δ_k, we can write

$$\frac{\tau\|s_{m^j-1}\|}{\sqrt{2}b_0} \le \frac{\tau\|s_{m^j-1}\|}{b_0} \le \frac{\tau\Delta_m}{a_1 b_0} \le \sigma_1(1-\eta_2) \le \sigma_1 \le \varepsilon_0, \tag{5.19}$$

where s_{m^j-1} is the last rejected step, just before finding an acceptable one and moving to the point (x_m, λ_m). Here $s_{m^j-1} = s_{m-1}$ if there are no rejected ones between s_{m-1} and s_m. We obtain from (5.3), that

$$Pred_{m^j-1} \ge \frac{r_{m^j-1}\varepsilon_0}{2}\min\left\{\varepsilon_0, \frac{\tau\Delta_{m^j-1}}{b_0}\right\} \ge \frac{r_{m^j-1}\varepsilon_0\tau\|s_{m^j-1}\|}{2\sqrt{2}b_0}. \tag{5.20}$$

On the other hand, from (5.1),

$$|Ared_{m^j-1} - Pred_{m^j-1}| \le r_{m^j-1}b_4\|s_{m^j-1}\|^2.$$

From (5.19), (5.20), and the above inequality, we have

$$\frac{|Ared_{m^j-1} - Pred_{m^j-1}|}{Pred_{m^j-1}} \le \frac{2\sqrt{2}b_4 b_0}{\tau\varepsilon_0}\|s_{m^j-1}\| \le \frac{2\sqrt{2}b_4 b_0^2\sigma_1(1-\eta_2)}{\tau^2\varepsilon_0} \le (1-\eta_2).$$

The above inequality implies that the step s_{m^j-1} was an acceptable step, i.e., $s_{m^j-1} = s_{m-1}$. It also implies that $\Delta_{m^j-1} \le \Delta_m$ and means that Δ_{m^j-1} satisfies (5.18). This contradicts the supposition that m is the first integer such that (5.18) holds. Therefore, there is no integer k such that (5.18) holds. The lemma is proved. □

The following theorem proves that under the global assumptions, either the algorithm satisfies its termination condition, or it converges to a feasible point.

THEOREM 5.12. *Let the global assumptions hold. If all members of the sequence of iterates generated by the algorithm fail to satisfy the termination condition, then*

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

Proof. We prove the theorem in two steps. First, we show that $\liminf_{k\to\infty}\|h_k\| = 0$, then we use this result to prove the theorem.

Assume there is an $\varepsilon_1 > 0$ such that $\|h_k\| \ge \varepsilon_1$, for all k. For any k, we have

$$\Phi_k - \Phi_{k+1} = Ared_k \ge \eta_1 Pred_k \ge \frac{\eta_1}{2}\|h_k\|\min\left[\frac{\tau\Delta_k}{b_0}, \|h_k\|\right]. \tag{5.21}$$

Since $\{\Phi_k\}$ is bounded below, $\Phi_{k+1} < \Phi_k$, for all $k \geq k_2$, where k_2 is as in Lemma 5.10 and $\|h_k\| \geq \varepsilon_1$ for all k, it follows that

$$\liminf_{k\to\infty} \Delta_k = 0.$$

On the other hand, because $\|h_k\| \geq \varepsilon_1$ for all k, Lemma 5.11 implies the existence of a constant $\bar{c}_3$, such that $\Delta_k > \bar{c}_3$ for all k, which is a contradiction with the above limit.

Therefore, the assumption $\|h_k\| \geq \varepsilon_1$ for all k has led to a contradiction. Hence

$$\liminf_{k\to\infty} \|h_k\| = 0. \tag{5.22}$$

This result shows that at least one subsequence of $\{x_k\}$ will converge to a feasible point.

Now we show that every subsequence will converge to a feasible point. Suppose that there exists a subsequence $\{k'_j\}$ of indices such that $\|h_{k'_j}\| > \varepsilon_1$. Because of this and (5.22) we may select two subsequences $\{k_j\}$ and $\{l_j\}$ as follows: Let $\{k_j\} \subset \{k'_j\}$ and for each j we select an l_j, such that

$$l_j = \max\left\{l \in [k_j, k_{j+1}) : \|h_i\| > \frac{\varepsilon_1}{2},\ k_j \leq i \leq l\right\}.$$

From (5.21), for all iterates l such that $k_j \leq l \leq l_j, j = 1, 2, \ldots$, we have

$$\Phi_l - \Phi_{l+1} \geq \frac{\eta_1 \varepsilon_1}{4} \min\left[\frac{\tau \Delta_k}{b_0}, \frac{\varepsilon_1}{2}\right].$$

From the above inequality, it follows that

$$\Phi_{k_j} - \Phi_{l_j+1} = \sum_{l=k_j}^{l_j} (\Phi_l - \Phi_{l+1}) \geq \frac{\eta_1 \varepsilon_1}{4} \sum_{l=k_j}^{l_j} \min\left[\frac{\tau \Delta_l}{b_0}, \frac{\varepsilon_1}{4}\right].$$

This implies $\sum_{l=k_j}^{l_j} \Delta_l \to 0$. But

$$\sum_{l=k_j}^{l_j} \Delta_l \geq \sum_{l=k_j}^{l_j} \frac{\|s_l\|}{\sqrt{2}} \geq \frac{1}{2}\|x_{k_j} - x_{l_j+1}\|.$$

So, as $j \to \infty$, $\|x_{k_j} - x_{l_j+1}\| \to 0$. This implies that there exists an integer k_3 sufficiently large such that $\|x_{k_j} - x_{l_j+1}\| \leq \frac{\varepsilon_1}{2\gamma}$, where $\gamma = \max(b_2, 1)$. Now, using (4.4), we have

$$\|h_{k_j}\| \leq \|h_{k_j} - h_{l_j+1}\| + \|h_{l_j+1}\| \leq \frac{b_2 \varepsilon_1}{2\gamma} + \frac{\varepsilon_1}{2} \leq \varepsilon_1$$

for all k_j sufficiently large, which is a contradiction.

So the supposition that $\|h_{k'_j}\| > \varepsilon_1$ has led to a contradiction. Hence, the supposition is wrong and the theorem is proved. □

The following lemma is needed in the proof of Theorem 5.14. It proves that under the assumption that the algorithm does not terminate, if $\{\|Z_k^T \nabla f_k\|\}$ is bounded away from zero, then the trust-region radius will be bounded away from zero.

LEMMA 5.13. *Let the global assumptions hold. If all members of the sequence of iterates generated by the algorithm fail to satisfy the termination condition and satisfy* $\|Z_k^T \nabla f_k\| > \varepsilon_2$, *for some fixed constant* $\varepsilon_2 > 0$, *then*

$$\Delta_k \geq c_4, \tag{5.23}$$

where c_4 *is a positive constant independent on* k.

Proof. Since the algorithm does not terminate, then from Theorem 5.12, $\|h_k\| \to 0$. Hence there exists an integer k_4 sufficiently large, such that, for all $k \geq k_4$, we have

$$\|h_k\| \leq \min\left\{\frac{\varepsilon_2}{2bb_0}, \frac{\varepsilon_2}{16\sqrt{2}b_5}\min\left[1, \frac{\varepsilon_2}{4b\Delta_\star}\right]\right\}. \tag{5.24}$$

Now, using (5.24), we can write

$$\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\| \geq \|Z_k^T \nabla f_k\| - bb_o\|h_k\| \geq \frac{\varepsilon_2}{2}.$$

From Lemmas 5.3, 5.4, and the above inequality, we can write

$$Pred_k \geq \frac{1}{2}Tpred_k + \left\{\frac{\varepsilon_2}{16}\min\left[1, \frac{\varepsilon_2}{4b\Delta_\star}\right] - \sqrt{2}b_5\|h_k\|\right\}\Delta_k.$$

Again, by using (5.24), we obtain

$$Pred_k \geq \frac{1}{2}Tpred_k \geq \frac{1}{8}\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|\min\left[\Delta_k, \frac{\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|}{2b}\right].$$

Hence, for all $k \geq k_4$, we have

$$Pred_k \geq \frac{\varepsilon_2}{16}\min\left[\Delta_k, \frac{\varepsilon_2}{4b}\right]. \tag{5.25}$$

The rest of the proof is by contradicting (5.23). Suppose that $\{\Delta_k\}$ is not bounded away from zero. Then there exists a sequence of indices $\{k_j\}$ such that

$$\Delta_k < a_1\sigma_2(1 - \eta_2) \tag{5.26}$$

for all $k \in \{k_j\}$, where

$$\sigma_2 = \min\left\{\frac{\varepsilon_2}{4b}, \frac{\varepsilon_2}{16\sqrt{2}r_\star b_4}, \frac{\Delta_{k_4}}{a_1(1 - \eta_2)}\right\}.$$

Let m be the first integer such that (5.26) holds. It is clear that $m \geq k_4 + 1$. Using (5.26), then from the way of updating Δ_k, we can write

$$\frac{\|s_{m^j-1}\|}{\sqrt{2}} \leq \|s_{m^j-1}\| \leq \frac{\Delta_m}{a_1} < \sigma_2(1 - \eta_2) \leq \sigma_2 \leq \frac{\varepsilon_2}{4b}, \tag{5.27}$$

where $s_m^j - 1$ is the last rejected step, just before finding an acceptable one and moving to the point (x_m, λ_m). Observe that $s_m^j - 1 = s_{m-1}$ if there are no rejected steps between s_{m-1} and s_m. We obtain from (5.25) and (5.27), that

$$Pred_{m^j-1} \geq \frac{\varepsilon_2}{16\sqrt{2}}\|s_{m^j-1}\|. \tag{5.28}$$

From (5.1), we have

$$|Ared_{m^j-1} - Pred_{m^j-1}| \leq r_\star b_4 \|s_{m^j-1}\|^2.$$

By using the above inequality and (5.28), we obtain

$$\frac{|Ared_{m^j-1} - Pred_{m^j-1}|}{Pred_{m^j-1}} \leq \frac{16\sqrt{2}r_\star b_4}{\varepsilon_2}\|s_{m^j-1}\| \leq \frac{16\sqrt{2}r_\star b_4\sigma_2(1-\eta_2)}{\varepsilon_2} \leq (1-\eta_2).$$

The above inequality implies that the step s_{m^j-1} was an acceptable one, i.e., $s_{m^j-1} = s_{m-1}$. It also implies that $\Delta_{m^j-1} \leq \Delta_m$ and means that $m-1$ satisfies (5.26). This contradicts the supposition that m is the first integer such that (5.26) holds. Therefore, there is no integer k such that (5.26) holds. The lemma is proved. □

The following theorem proves that under the global assumptions, if each member of the sequence of iterates generated by the algorithm does not satisfy the termination condition of the algorithm, then there exists a subsequence $\{x_{k_j}\}$ of these iterates for which $\{\|Z_{k_j}^T \nabla f_{k_j}\|\}$ converges to zero.

THEOREM 5.14. *Let the global assumptions hold. If all members of the sequence of iterates generated by the algorithm fail to satisfy the termination condition, then*

$$\liminf_{k\to\infty} \|Z_k^T \nabla f_k\| = 0.$$

Proof. The proof is by contradiction. Suppose that there exists an $\varepsilon_3 > 0$ such that $\|Z_k^T \nabla f_k\| \geq \varepsilon_3$ for all k. As in Lemma 5.13, there exists an integer k_4 sufficiently large such that for all $k \geq k_4$, we have

$$Pred_k \geq \frac{\varepsilon_3}{16}\min\left[\Delta_k, \frac{\varepsilon_3}{4b}\right].$$

On the other hand, for all $k \geq k_2$, $r_k = r_\star$. Hence, for $k \geq \max\{k_4, k_2\}$, we have

$$\Phi_k - \Phi_{k+1} = Ared_k \geq \eta_1 Pred_k \geq \frac{\eta_1\varepsilon_3}{16}\min\left[\Delta_k, \frac{\varepsilon_3}{4b}\right]. \tag{5.29}$$

Since Φ_k is bounded below and $\Phi_{k+1} < \Phi_k$, for all $k \geq \max\{k_4, k_2\}$, we have

$$\liminf_{k\to\infty} \Delta_k = 0.$$

On the other hand, because of the assumption that the algorithm does not terminate and that $\|Z_k^T \nabla f_k\| \geq \varepsilon_3$, for all k, Lemma 5.13 implies the existence of a constant $\bar{c}_4$, such that $\Delta_k > \bar{c}_4$ for all k. This contradicts the above limit. Therefore, the supposition $\|Z_k^T \nabla f_k\| \geq \varepsilon_3$, for all k has led to a contradiction. Hence the supposition is wrong and the lemma is proved. □

The above two theorems imply that under the global assumptions and the assumption that the algorithm does not terminate, the algorithm produces an infinite sequence of iterates $\{x_k\}$ that satisfies

$$\liminf_{k\to\infty} [\|h_k\| + \|Z_k^T \nabla f_k\|] = 0. \tag{5.30}$$

This result contradicts the assumption that the algorithm does not terminate and means that the termination condition of the algorithm will be met after finitely many iterations.

Satisfying the termination condition by itself means that the point at which the algorithm terminates lies in a ball of radius $O(\varepsilon)$ and center at a stationary point $(x_\star, \lambda_\star)$.

In practice there is no difference between $\liminf_{k\to\infty} [\ \|h_k\| + \|Z_k^T \nabla f_k\|\ ] = 0$ and $\lim_{k\to\infty} [\ \|h_k\| + \|Z_k^T \nabla f_k\|\ ] = 0$. Both mean that the algorithm will terminate after finitely many iterations.

If the point $(x_\star, \lambda_\star)$ is not an isolated local minimizer that satisfies the second order sufficiency condition, then our analysis is stopped here. On the other hand, if the algorithm avoids the neighborhoods of stationary points that do not satisfy the second order sufficiency condition, then we remove the termination condition from the algorithm and proceed, in the following section, with the local analysis.

6. The local analysis. In this section, in addition to the global assumptions, we add the following assumption.

LOCAL ASSUMPTION A. We assume that the problem has a finite number of isolated local minimizers and each one satisfies the second order sufficiency condition.

We remove the termination condition from the algorithm and proceed with the analysis. Because there is no termination condition, Lemma 5.10 and Theorems 5.12 and 5.14 are no longer valid. However, the global analysis still implies that given any $\varepsilon > 0$ there exists a ball $\mathcal{B}_\varepsilon(\bar{x}, \bar{\lambda})$ of radius ε and center $(\bar{x}, \bar{\lambda})$, where $(\bar{x}, \bar{\lambda})$ is a stationary point of the problem, such that the sequence of iterates generated by the algorithm is not bounded away from this ball, i.e., for some k sufficiently large, we have $(x_k, \lambda_k) \in \mathcal{B}_\varepsilon(\bar{x}, \bar{\lambda})$.

The local analysis of our algorithm is presented in three sections. In §6.1 we study the behavior of the penalty parameter after removing the termination condition from the algorithm. In §6.2, we prove that the sequence of iterates $\{(x_k, \lambda_k)\}$ converges to a local minimizer $(x_\star, \lambda_\star)$. Section 6.3 is devoted to studying the local rate of convergence of our algorithm. We show that our globalization strategy will not disrupt the fast local rate of convergence.

If the point $(x_\star, \lambda_\star)$ satisfies the second order sufficiency condition (see §1), then by the continuity assumption, there exists a neighborhood $\mathcal{N}(x_\star, \lambda_\star)$ of $(x_\star, \lambda_\star)$ such that $Z(x)^T \nabla_x^2 l(x, \lambda) Z(x) > 0$ for all $(x, \lambda) \in \mathcal{N}(x_\star, \lambda_\star)$.

6.1. The local behavior of the penalty parameter. In this section, we prove technical lemmas needed to study the local behavior of the penalty parameter. At the end of this section we prove that, under the global assumptions and Assumption A, the penalty parameter is bounded.

The point $(x_\star, \lambda_\star)$ is used in this section to mean a stationary point of the problem that satisfies the second order sufficiency condition and $\mathcal{N}(x_\star, \lambda_\star)$ is used to mean a neighborhood of $(x_\star, \lambda_\star)$ such that $Z(x)^T \nabla_x^2 l(x, \lambda) Z(x) > 0$, for all $x \in \mathcal{N}(x_\star, \lambda_\star)$.

LEMMA 6.1. *If $(x_k, \lambda_k) \in \mathcal{N}(x_\star, \lambda_\star)$, there exists a positive constant e_1, such that*

$$\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\| \geq e_1 \|v_k\|.$$

Proof. Since $(x_k, \lambda_k) \in \mathcal{N}(x_\star, \lambda_\star)$ then $Z_k^T B_k Z_k$ is positive definite. Hence, there exists a positive constant e_1 such that, for all k sufficiently large $e_1 \|v_k\|^2 \leq v_k^T Z_k^T B_k Z_k v_k$. Now, since

$$v_k^T Z_k^T B_k Z_k v_k \leq -(Z_k^T \nabla f_k + Z_k^T B_k s_k^n)^T v_k,$$

we can write

$$e_1 \|v_k\| \leq \|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|. \tag{6.1}$$

This completes the proof. □

LEMMA 6.2. *If* $(x_k, \lambda_k) \in \mathcal{N}(x_\star, \lambda_\star)$ *is such that* $\|h_k\| \leq e_2\|s_k\|$ *where* $0 < e_2 \leq \frac{1}{2b_0}$ *and* b_0 *is as in* (4.2), *then*

$$\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\| \geq \frac{e_1}{2}\|s_k\|.$$

Proof. Since $\|u_k\| + \|v_k\| \geq \|s_k\|$, then by using (4.2) and (6.1), we obtain

$$e_1 b_0 \|h_k\| + \|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\| \geq e_1 \|s_k\|.$$

When $\|h_k\|_2 \leq e_2\|s_k\|_2$, we have

$$\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|_2 \geq e_1(1 - e_2 b_0)\|s_k\|.$$

Using $e_2 \leq \frac{1}{2b_0}$, we obtain the desired result. □

LEMMA 6.3. *If* $(x_k, \lambda_k) \in \mathcal{N}(x_\star, \lambda_\star)$ *is such that* $\|h_k\| \leq e_3\|s_k\|$ *where* e_3 *is a positive constant that satisfies*

$$e_3 \leq \min\left[e_2, \frac{e_1 \min(4b, \sqrt{2}e_1)}{64\sqrt{2}bb_5}\right], \tag{6.2}$$

where b *is as in* (4.1), b_5 *is as in Lemma* 5.4, e_1 *is as in Lemma* 6.1, *and* e_2 *is as in Lemma* 6.2, *then*

$$Pred_k \geq \frac{1}{2}Tpred_k + \frac{r_k}{2}Npred_k. \tag{6.3}$$

Proof. From Lemmas 5.3 and 5.4, we have

$$Pred_k \geq \frac{1}{2}Tpred_k + \frac{1}{8}\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\| \min\left[\Delta_k, \frac{\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|}{2b}\right]$$
$$-b_5\|s_k\|\|h_k\| + \frac{r_k}{2}Npred_k. \tag{6.4}$$

Now, since $\|h_k\| \leq e_3\|s_k\|$ and $e_3 \leq e_2$ then by using Lemma 6.2 we have $\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\| \geq \frac{e_1}{2}\|s_k\|$, and using (6.2), we obtain

$$\frac{1}{8}\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\| \min\left[\Delta_k, \frac{\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|}{2b}\right] - b_5\|s_k\|\|h_k\|$$
$$\geq \left\{\frac{e_1}{16}\min\left[\frac{1}{\sqrt{2}}, \frac{e_1}{4b}\right] - b_5 e_3\right\}\|s_k\|^2 \geq 0.$$

The remainder of the proof follows immediately. □

From the proof of the above lemma we see that, if $\|h_k\| \leq e_3\|s_k\|$, then the second term in (6.4) will cancel the third term and the fourth term need never enter into the calculation. This implies that if we set $r_k = \rho_{k-1}$, (see Scheme 3.4), inequality (6.3) remains valid. In this case the algorithm will not update r_k using (3.6) because inequality (3.7) will be satisfied.

LEMMA 6.4. *If for all* k, $(x_k, \lambda_k) \in \mathcal{N}(x_\star, \lambda_\star)$, *then* $r_k \leq r^\star$, *where* $r^\star$ *is a positive constant that does not depend on* k.

Proof. First we follow a proof similar to the proof of Lemma 5.8. We demonstrate the boundedness of the sequence $\{\overline{\rho}_k\}$. The rest of the proof follows because, for all k, $r_k \leq \overline{\rho}_k$. □

LEMMA 6.5. *Under the global and the local assumptions, the sequence r_k is bounded.*

Proof. Because we have a finite number, p say, of local minimizers that satisfies the second order sufficiency condition (see Assumption A), we can find a radius $\bar{\varepsilon}$, such that $\mathcal{B}^i_{\bar{\varepsilon}}(x^i_\star, \lambda^i_\star) \subset \mathcal{N}^i(x^i_\star, \lambda^i_\star)$, for $i = 1, 2, \ldots, p$.

Now consider the set $\mathcal{B}_{\bar{\varepsilon}} = \bigcup_{i=1}^{p} \mathcal{B}^i_{\bar{\varepsilon}}(x^i_\star, \lambda^i_\star)$. If any iterate k is such that $(x_k, \lambda_k) \notin \mathcal{B}_{\bar{\varepsilon}}$ then from the global analysis, there exists a constant $\bar{r}_\star$ such that, $r_k \leq \bar{r}_\star$. Observe that $\bar{r}_\star$ depends on $\bar{\varepsilon}$. Here $\bar{\varepsilon}$ is fixed. On the other hand, if $(x_k, \lambda_k) \in \mathcal{B}_{\bar{\varepsilon}}$ then from Lemma 6.4, there exists a constant $\bar{r}^\star$, such that $r_k \leq \bar{r}^\star$. Now take $\bar{r} = \max(\bar{r}_\star, \bar{r}^\star)$, we can see that the sequence $\{r_k\}$ is bounded by $\bar{r}$. □

Now we follow the argument that comes immediately after the proof of Lemma 5.9, and then follow the proof of Lemma 5.10, we conclude that there exists an integer $\bar{k}$ such that for all $k \geq \bar{k}$ the sequence of penalty parameters reaches its upper bound.

In the following section we study the sequence of points $\{(x_k, \lambda_k)\}$ generated by the algorithm after the penalty parameter reaches its upper bound.

Without loss of generality we may assume that the sequence of penalty parameters is independent of k.

6.2. First order convergence. From the global analysis, there exists a subsequence of points $\{(x_{k_j}, \lambda_{k_j})\}$ generated by the algorithm, such that $(x_k, \lambda_k) \in \mathcal{N}(x_\star, \lambda_\star)$, for all $k \in \{k_j\}$.

Consider the level sets $\mathcal{L}_k \equiv \{(x, \lambda) : \Phi(x, \lambda, r) \leq \Phi(x_k, \lambda_k, r)\}$. There exists an integer $\hat{k}$ sufficiently large, such that $\mathcal{L}_{\hat{k}} \subset \mathcal{N}(x_\star, \lambda_\star)$.

The following lemma proves that there exists an index $\bar{k}$ such that all the subsequent iterates generated by the algorithm will never leave the level set $\mathcal{L}_{\bar{k}}$.

LEMMA 6.6. *Under the global and local assumptions, there exists an index $\bar{k}$ sufficiently large, such that $(x_k, \lambda_k) \in \mathcal{L}_{\bar{k}}$, for all $k > \bar{k}$.*

Proof. From the global analysis and local Assumption A, there exists an index $\bar{k}$ such that $\mathcal{L}_{\bar{k}} \subset \mathcal{B}_{\bar{\varepsilon}}$.

The proof now is by contradiction. Suppose that some iterates leave the set $\mathcal{L}_{\bar{k}}$. Let $m + 1$ be the first iterate that leaves the set. Therefore, $(x_m, \lambda_m) \in \mathcal{L}_{\bar{k}}$ and $(x_{m+1}, \lambda_{m+1}) \notin \mathcal{L}_{\bar{k}}$. Since s_m is an acceptable step, then we have

$$\Phi_m - \Phi_{m+1} = Ared_m \geq \eta_1 Pred_m \geq 0.$$

Then $\Phi_m \geq \Phi_{m+1}$. This implies that $(x_{m+1}, \lambda_{m+1}) \in \mathcal{L}_{\bar{k}}$. This gives a contradiction. Hence the lemma is proved. □

THEOREM 6.7. *Under the global and local assumptions, the algorithm will generate points that satisfy*

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

Proof. The proof is similar to the proof of Theorem 5.12. □

Under the global and local assumptions, Theorem 5.14 can be improved.

THEOREM 6.8. *Under the global and the local assumptions, we have*

$$\lim_{k \to \infty} \|Z_k^T \nabla f_k\| = 0.$$

Proof. First we follow a proof similar to the proof of Theorem 5.14. We demonstrate

$$\liminf_{k\to\infty} \|Z_k^T \nabla f_k\| = 0. \tag{6.5}$$

The rest of the proof follows by contradiction. Suppose there exists a subsequence of indices $\{k_j\}$ such that $k_j \geq \bar{k}$, where $\bar{k}$ is as in Lemma 6.6, and $\|Z_k^T \nabla f_k\| > \sigma_1$ for all $k \in \{k_j\}$, where $\sigma_1 > 0$.

Take an iterate $k' \in \{k_j\}$ sufficiently large such that for all $k \geq k'$, we have

$$\|h_k\| \leq \min\left\{\frac{\sigma_1}{2bb_0}, \frac{\sigma_1}{16\sqrt{2}b_5}\min\left[1, \frac{\sigma_1}{4b\Delta_\star}\right]\right\}. \tag{6.6}$$

For some $\beta > o$ and any $x \in \Omega$, we have

$$\begin{aligned}\|Z(x)^T \nabla f(x)\| &\geq \|Z_{k'}^T \nabla f_{k'}\| - \|Z(x)^T \nabla f(x) - Z_{k'}^T \nabla f_{k'}\| \\ &\geq \|Z_{k'}^T \nabla f_{k'}\| - \beta\|x - x_{k'}\|.\end{aligned}$$

This implies that $\|Z(x)^T \nabla f(x)\| \geq \frac{1}{2}\|Z_{k'}^T \nabla f_{k'}\| > \frac{\sigma_1}{2}$ holds for every $x \in \Omega$ that satisfies

$$\|x - x_{k'}\| \leq \frac{\|Z_{k'}^T \nabla f_{k'}\|}{2\beta}.$$

Therefore, take

$$\sigma_2 = \frac{\|Z_{k'}^T \nabla f_{k'}\|}{2\beta}$$

and consider the ball $U_{\sigma_2} = \{x : \|x - x_{k'}\| \leq \sigma_2\}$. For all $k \geq k'$ such that $x_k \in U_{\sigma_2}$, we have $\|Z_k^T \nabla f_k\| > \frac{\sigma_1}{2}$. As in Lemma 5.13 (because of (6.6)), we have, for all $k \geq k'$

$$\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\| > \frac{\sigma_1}{4}$$

and

$$Pred_k \geq \frac{1}{8}\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\| \min\left\{\Delta_k, \frac{\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|}{2b}\right\}.$$

This implies that for any iterate $k \geq k'$ that lies inside the ball, we have

$$Pred_k > \frac{\sigma_1}{32}\min\left[\Delta_k, \frac{\sigma_1}{8b}\right].$$

Because of (6.5), the iterates, for all $k \geq k'$, cannot stay in this ball. Let $l+1$ be the first integer greater than k' such that the point x_{l+1} does not lie inside the ball $\mathcal{U}_{\sigma_2}$. Hence,

$$\begin{aligned}\Phi_{k'} - \Phi_{l+1} &= \sum_{k=k'}^{l}(\Phi_k - \Phi_{k+1}) \geq \sum_{k=k'}^{l}\eta_1 Pred_k \\ &\geq \sum_{k=k'}^{l}\frac{\eta_1\sigma_1}{32}\min\left[\Delta_k, \frac{\sigma_1}{8b}\right].\end{aligned}$$

Therefore,

$$\Phi_{k'} - \Phi_{l+1} \geq \frac{\eta_1 \sigma_1}{32} \min\left[\frac{\sigma_2}{\sqrt{2}}, \frac{\sigma_1}{8b}\right]. \tag{6.7}$$

Since Φ_k is bounded below and is a decreasing sequence, $\{\Phi_k\}$ converges to some limit Φ_*. Taking the limit as l goes to infinity in inequality (6.7), we obtain

$$\Phi_{k'} - \Phi_* \geq \frac{\eta_1 \sigma_1}{32} \min\left[\frac{\sigma_2}{\sqrt{2}}, \frac{\sigma_1}{8b}\right].$$

If we now take the limit as k' goes to infinity, we obtain

$$0 \geq \frac{\eta_1 \sigma_1}{32} \min\left[\frac{\sigma_2}{\sqrt{2}}, \frac{\sigma_1}{8b}\right],$$

which contradicts the fact that $\sigma_1 > 0$ and $\sigma_2 > 0$. Hence there is no such sequence and the lemma is proved. □

6.3. The local rate of convergence. In this section we prove Lemma 6.9 which is needed in our analysis. Then we prove Lemma 6.10 which proves that under the global and the local assumptions, for k sufficiently large, all the trial steps will be accepted and the trust region will not be decreased. In Theorems 6.11 and 6.12, we study the local rate of convergence of our algorithm. We show that asymptotically the trust region will be inactive and hence the fast local rate of convergence will be maintained.

LEMMA 6.9. *Under the global and local assumptions, there exists a positive constant e_4 independent of k such that*

$$Pred_k \geq e_4 \|s_k\|^2.$$

Proof. If $\|h_k\| \leq e_3 \|s_k\|$, where e_3 is as in (6.2), then using Lemmas 6.3 and 5.3

$$Pred_k \geq \frac{1}{2} Tpred_k \geq \frac{1}{8} \|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\| \min\left[\frac{\|s_k\|}{\sqrt{2}}, \frac{\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\|}{2b}\right].$$

But, since $\|Z_k^T \nabla f_k + Z_k^T B_k s_k^n\| \geq \frac{e_1}{2} \|s_k\|$, then

$$Pred_k \geq \frac{e_1}{16} \min\left[\frac{1}{\sqrt{2}}, \frac{e_1}{4b}\right] \|s_k\|^2.$$

On the other hand, when $\|h_k\| > e_3 \|s_k\|$, from (5.3) and the fact that $r \geq \underline{\rho}_0 = 1$, we have

$$Pred_k \geq \frac{r}{2} \|h_k\| \min\left[\frac{\tau \Delta_k}{b_0}, \|h_k\|\right] \geq \frac{e_3}{2} \min\left[\frac{\tau}{\sqrt{2} b_0}, e_3\right] \|s_k\|^2.$$

If we take

$$e_4 = \min\left\{\frac{e_1}{64\sqrt{2}b} \min\left[4b, \sqrt{2} e_1\right], \frac{e_3}{2\sqrt{2} b_0} \min\left[\tau, \sqrt{2} b_0 e_3\right]\right\},$$

we obtain the desired result. □

We add to our local assumptions the following set of assumptions.

LOCAL ASSUMPTION B. $\nabla_x^2 l$ is Lipschitz continuous in a neighborhood of the solution $x_\star$.

LOCAL ASSUMPTION C. If an approximation to the exact Hessian is used, then for all k, B_k satisfies

$$\lim_{k\to\infty} \frac{\|Z_k(B_k - \nabla_x^2 l_\star)s_k\|}{\|s_k\|} = 0. \tag{6.8}$$

The above assumption is the Boggs–Tolle–Wang characterization of q-superlinear convergence of $\{x_k\}$ to $x_\star$. It is proved by Boggs, Tolle, and Wang (1982)[1] and Powell (1983)[17] that under the local assumptions, Algorithm 1.1 converges to $x_\star$ q-superlinearly. On the other hand, if the exact Hessian is used, the local convergence rate is q-quadratic. (See Goodman (1985)[10].)

The following lemma shows that, for all k large enough, the trust-region radius Δ_k will not be decreased, i.e., the sequence $\{\Delta_k\}$, for k large enough, will form a nondecreasing sequence.

LEMMA 6.10. *Under the global and local assumptions, there exists an integer k_5 sufficiently large, such that for all $k \geq k_5$, we have*

$$\frac{Ared_k}{Pred_k} \geq \eta_2.$$

Proof. We have, using (2.1),

$$\begin{aligned}
\Phi(x_k + s_k, \lambda_{k+1}, r) &= \Phi(x_k, \lambda_{k+1}, r) + \nabla_x \Phi(x_k, \lambda_{k+1}, r)^T s_k \\
&\quad + \frac{1}{2} s_k^T \nabla_x^2 \Phi(x_k, \lambda_{k+1}, r)^T s_k + o(\|s_k\|^2) \\
&= \Phi(x_k, \lambda_k, r) + \nabla_x \Phi(x_k, \lambda_k, r)^T s_k + \frac{1}{2} s_k^T \nabla_x^2 \Phi(x_k, \lambda_k, r)^T s_k \\
&\quad + (\Delta\lambda_k)^T h_k + (\Delta\lambda_k)^T \nabla h_k^T s_k + \frac{1}{2} s_k^T (\Delta\lambda_k)^T \nabla^2 h_k s_k + o(\|s_k\|^2).
\end{aligned}$$

From the above equation, using the definition of $Ared_k$, we obtain

$$\begin{aligned}
Ared_k &= -\nabla_x l(x_k, \lambda_k)^T s_k - \frac{1}{2} s_k^T \nabla_x^2 l(x_k, \lambda_k) s_k \\
&\quad -(\Delta\lambda_k)^T (h_k + \nabla h_k^T s_k) - r[\|h_k + \nabla h_k^T s\|^2 - \|h_k\|^2] \\
&\quad -\frac{1}{2} s_k^T \nabla^2 h_k \Delta\lambda_k s_k - r s_k^T \nabla^2 h_k h_k s_k - o(\|s_k\|^2).
\end{aligned}$$

If we use the definition of $Pred_k$ and the above inequality, we obtain

$$\begin{aligned}
Ared_k \geq Pred_k - o(\|s_k\|^2) - r|s_k^T \nabla^2 h_k h_k s_k| - \frac{1}{2} s_k^T [\nabla_x^2 l_\star - B_k] Z_k v_k \\
-\frac{1}{2} s_k^T [\nabla_x^2 l_k - \nabla_x^2 l_\star] s_k - \frac{\alpha_k}{2} s_k^T \nabla_x^2 l_\star Y_k u_k - \frac{1}{2} (\Delta\lambda_k)^T \nabla h_k^T s_k.
\end{aligned}$$

We show first that the last two terms are $o(\|s_k\|^2) + o(\|s_k\|\|h_k\|)$. By differentiating the normal equation $Y(x)^T[\nabla_x l(x, \lambda(x))] = 0$ at $x = x_\star$, we obtain $Y_\star^T[\nabla_x^2 l_\star + \nabla h_\star \nabla \lambda_\star^T] = 0$, or equivalently $R_\star \nabla \lambda_\star^T = -Y_\star^T \nabla_x^2 l_\star$. Therefore,

$$\begin{aligned}
-\frac{1}{2}(\Delta\lambda_k)^T \nabla h_k^T s_k &= -\frac{\alpha_k}{2} s_k^T \nabla\lambda_k R_k^T u_k + o(\|s_k\|^2) \\
&= -\frac{\alpha_k}{2} s_k^T [\nabla\lambda_k R_k^T - \nabla\lambda_\star R_\star^T] u_k - \frac{\alpha_k}{2} s_k^T \nabla\lambda_\star R_\star^T u_k + o(\|s_k\|^2) \\
&= -\frac{\alpha_k}{2} s_k^T \nabla\lambda_\star R_\star^T u_k + o(\|s_k\|^2) + o(\|s_k\|\|h_k\|).
\end{aligned}$$

Hence,

$$-\frac{\alpha_k}{2}s_k^T\nabla_x^2 l_\star Y_k u_k - \frac{1}{2}(\Delta\lambda_k)^T\nabla h_k^T s_k = -\frac{\alpha_k}{2}s_k^T\nabla_x^2 l_\star[Y_\star - Y_k]u_k + o(\|s_k\|^2) + o(\|s_k\|\|h_k\|), = o(\|s_k\|^2) + o(\|s_k\|\|h_k\|).$$

Using Lemma 6.4, for k large enough, we have

$$\frac{Ared_k}{Pred_k} \geq 1 - \frac{1}{e_4}\left[\frac{o(\|s_k\|^2)}{\|s_k\|^2} + \frac{o(\|s_k\|\|h_k\|)}{\|s_k\|^2} + \frac{r|s_k^T\nabla^2 h_k h_k s_k|}{\|s_k\|^2} + \frac{|s_k^T[\nabla_x^2 l_\star - B_k]Z_k v_k|}{2\|s_k\|^2} + \frac{|s_k^T[\nabla_x^2 l_k - \nabla_x^2 l_\star]s_k|}{2\|s_k\|^2}\right].$$

Using the local assumptions, Theorem 6.7, Theorem 6.8, Lemma 6.1, and inequality (4.2), we conclude that there exists an integer k_5 sufficiently large such that, for all $k \geq k_5$, we have

$$\frac{Ared_k}{Pred_k} \geq \eta_2.$$

Hence, the theorem is proved. □

In our definition of $Pred_k$ we used $\frac{1}{2}s_k^T B_k Z_k v_k$ instead of $\frac{1}{2}s_k^T B_k s_k$ and used $h_k + \frac{1}{2}\nabla h_k^T s_k$ instead of $h_k + \nabla h_k^T s_k$. This way of defining $Pred_k$ allows us, when comparing with the second order approximation of the terms of $Ared_k$, to have two extra terms, namely, $\frac{1}{2}s_k^T B_k Y_k u_k$ and $\frac{1}{2}\nabla h_k^T s_k$. These two terms are very important in our local analysis because they allow us, using local Assumption C, to prove that $Pred_k$ approximates $Ared_k$ to within terms that are of order $o(\|s_k\|^2)$ or $o(\|s_k\|\|h_k\|)$.

Now as $k \to 0$, $\|h_k\| \to 0$ and $\|Z_k^T\nabla f_k\| \to 0$ and hence $\|s_k\| \to 0$. This implies that $\frac{Ared_k}{Pred_k} \to 1$, which means that for k sufficiently large all the steps produced by our algorithm are acceptable. This also means that for k sufficiently large the sequence of trust region radii $\{\Delta_k\}$ is a nondecreasing sequence.

The following two theorems show that the fast local rate of convergence will be maintained.

THEOREM 6.11. *Under the global and local assumptions, if the exact Hessian is used, then for k sufficiently large, $x_k \to x_\star$ q-quadratically.*

Proof. From Lemma 6.10, the trust region radius Δ_k for $k \geq k_5$ is updated according to the rule $\Delta_{k+1} = \min\{\Delta_\star, \max[\Delta_k, a_3\|s_k\|]\}$. Hence, $\Delta_k \geq \Delta_{k_5}$ for all $k \geq k_5$. However, for all k, $\Delta_k \leq \Delta_\star$.

First, we show that the trust region will be inactive for sufficiently large k. Suppose there exists an integer $k_6 \geq k_5$ such that the full normal and tangential components of the step are not taken for all $k \geq k_6$. This implies that for all $k \geq k_6$, $\|R_k^{-T}h_k\| = \|u_k\| > \Delta_k \geq \Delta_{k_6}$ and $\|(Z_k^T B_k Z_k)^{-1}(Z_k^T\nabla l_k + Z_k^T B_k s_k^n)\| > \|v_k\| = \Delta_k \geq \Delta_{k_6}$. But, using (4.2) and Lemma 6.1, this contradicts the fact that $\|h_k\| \to 0$ and $\|Z_k^T\nabla f_k\| \to 0$. Therefore, there exists a subsequence of indices $\{k_j\}$ such that $\|s_{k_j}^t\| \leq \Delta_{k_6}$ and $\|s_{k_j}^n\| \leq \Delta_{k_6}$ where all of $k_j \geq k_6$.

Let $m \in \{k_j\}$ be the smallest integer greater than k_6 such that $\|s_m^t\| \leq \Delta_{k_6}$, $\|s_m^n\| \leq \Delta_{k_6}$, and such that the local method, i.e., Algorithm 1.1, generates steps that are q-quadratic, i.e., satisfies

$$\|s_{k+1}\| \leq \beta_1\|s_k\|^2,$$

where β_1 is a constant. But since the local method converges r-quadratic in the components s_k^t and s_k^n, this implies the existence of an integer $k_7 \geq m$, such that for all $k \geq k_7$, we have

$$\|R_k^{-T} h_k\| \leq \beta_2 (\gamma_1^2)^{k_7}$$

and

$$\|(Z_k^T B_k Z_k)^{-1} (Z_k^T \nabla l_k + Z_k^T B_k s_k^n)\| \leq \beta_3 (\gamma_2^2)^{k_7},$$

where $\beta_2, \beta_3, \gamma_1$, and γ_2 are constants and $\gamma_1, \gamma_2 \in (0,1)$. This means that if we choose k_7 sufficiently large such that $\max\{\beta_2(\gamma_1^2)^{k_7}, \beta_3(\gamma_2^2)^{k_7}\} \leq \Delta_{k_7}$ then we have, $\|R_{k_7}^{-T} h_{k_7}\| \leq \Delta_{k_7}$, $\|(Z_{k_7}^T B_{k_7} Z_{k_7})^{-1} (Z_{k_7}^T \nabla l_{k_7} + Z_{k_7}^T B_{k_7} s_{k_7}^n)\| \leq \Delta_{k_7}$, and for all $k \geq k_7$, we have

$$\|R_k^{-T} h_k\| \leq \Delta_{k_7},$$

and

$$\|(Z_k^T B_k Z_k)^{-1} (Z_k^T \nabla l_k + Z_k^T B_k s_k^n)\| \leq \Delta_{k_7}.$$

But since, for $k \geq k_5$, we have $\Delta_k \leq \Delta_{k+1}$, and all the steps are acceptable, then

$$\|R_{k_7+1}^{-T} h_{k_7+1}\| \leq \Delta_{k_7} \leq \Delta_{k_7+1},$$

and

$$\|(Z_{k_7+1}^T B_{k_7+1} Z_{k_7+1})^{-1} (Z_{k_7+1}^T \nabla l_{k_7+1} + Z_{k_7+1}^T B_{k_7+1} s_{k_7+1}^n)\| \leq \Delta_{k_7} \leq \Delta_{k_7+1}.$$

The above two inequalities and the fact that for all $k \geq k_5$ all the steps are acceptable imply that the full step will be taken at iteration $k_7 + 1$. By induction, for all $k \geq k_7$, the trust region will be inactive and the full step will be accepted.

This means that the sequence $x_k, k \geq k_7$ generated by the algorithm is the sequence of iterates generated by Algorithm 1.1 and consequently the local rate of convergence is q-quadratic. □

THEOREM 6.12. *Under the global and local assumptions, if an approximation to the Hessian of the Lagrangian that satisfies* (6.8) *is used, then for k sufficiently large,* $x_k \to x_\star$ q-*superlinearly.*

Proof. From the above theorem, we have for all $k \geq k_7$, that the trust region will be inactive and the full step will be accepted, where k_7 is some sufficiently large integer. This means that the sequence $\{x_k\}, k \geq k_7$ generated by the algorithm is purely the sequence of iterates that is generated by Algorithm 1.1.

Second, it is proved by Boggs, Tolle, and Wang (1982)[1] that if we use a scheme for approximating B_k in Algorithm 1.1, then $x_k \to x_\star$ q-superlinear if and only if assumption (6.1) is satisfied.

Now as a consequence of the local assumptions and the above two parts of the proof, if k_8 is taken sufficiently large such that the local method, i.e., Algorithm 1.1, generates steps that are q-superlinear, we conclude that the local rate of convergence is q-superlinear. □

7. Concluding remarks. We have presented an algorithm for solving the equality constrained optimization problem. This algorithm has many desirable features. In this algorithm, we use Fletcher's differentiable penalty function as a merit function.

In computing the trial step, after factorizing ∇h_k using QR factorization, two inexpensive subproblems must be solved. One of them is an upper triangular linear system. The second one is a subproblem of smaller dimension $m \times m$ similar to the one we obtain when solving unconstrained optimization problems using a trust-region method.

In our algorithm, to obtain the matrix B_k, the exact Hessian of the Lagrangian can be used. On the other hand, an approximation to the Hessian matrix can also be used. For example, setting B_k to a fixed matrix for all k is valid. However, if B_k is obtained by quasi-Newton updates, the uniform boundedness assumption on B_k, condition (4.1), causes some difficulties. For an analysis of this problem for trust-region algorithms for unconstrained problems see, e.g., Powell (1984)[18], and for minimization problems with convex constraints, see, e.g., Toint (1988)[22]. The question of how to use a secant approximation to the Hessian of the Lagrangian is a research topic. We believe that Tapia (1988)[20] will be of considerable value here.

One of the main advantages of this algorithm is the way that the penalty parameter is updated. It is updated in such way as to ensure that the merit function is decreased at each iteration by at least a fraction of Cauchy decrease in the quadratic model of the linearized constraints and at the same time it can be decreased whenever it is warranted.

We have presented a convergence theory for this algorithm. We showed that the algorithm is well defined and is globally convergent. To the best of our knowledge this is the first time a global convergence theory has been proved for an algorithm with a nonmonotonic penalty parameter updating scheme. This updating scheme should avoid the numerical difficulties that may occur if the penalty parameter is increased at each iteration. We have also proved that the algorithm will terminate at a point that is not bounded away from a stationary point.

We also presented a local analysis for this algorithm. In our local analysis we proved that our globalization strategy will not disrupt the fast local rate of convergence.

Acknowledgments. This work was done while the author was visiting the Department of Computational and Applied Mathematics and the Center of Research on Parallel Computations, Rice University. He thanks Rice University for its financial support and for the congenial scientific atmosphere that it provided.

The author is also greatly indebted to Richard Byrd, John Dennis, and Richard Tapia for their valuable suggestions and comments on an earlier version of this paper.

REFERENCES

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

[2] R. 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. Also available as Technical Report CU-CS-313-85, Department of Computer Science, University of Colorado, Boulder.

[3] T. R. Coleman and A. R. Conn, *Nonlinear programming via an exact penalty function: asymptotic analysis*, Math. Programming, 24 (1982), pp. 123–136.

[4] 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.

[5] 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.

[6] ———, *A global convergence theory for the Celis–Dennis–Tapia trust-region algorithm for constrained optimization*, SIAM J. Numer. Anal., 28 (1991), pp. 266–290.

[7] R. FLETCHER, *Second order correction for nondifferentiable optimization*, in Lectures Notes in Mathematics 912, G.A.Watson, ed., Springer-Verlag, Berlin, New York, 1982.

[8] P. GILL, W. MURRAY, M. SAUNDERS, AND M. WRIGHT, *Some theoretical properties of an augmented Lagrangian merit function*, Report SOL 86-6R, Department of Operations Research, Stanford University, Stanford, CA, 1986.

[9] P. E. GILL AND W. MURRAY, *Newton-type method for linearly constrained optimization*, in Numerical methods for constrained optimization., P. E. Gill and W. Murray, eds., Academic Press, London, New York, San Francisco, 1974, pp. 29–92.

[10] J. GOODMAN, *Newton's method for constrained optimization*, Math. Programming, 33 (1985), pp. 162–171.

[11] M. C. MACIEL, *A global convergence theory for a general class of trust-region algorithms for equality constrained optimization*, Ph.D. thesis, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1992.

[12] N. MARATOS, *Exact penalty function algorithms for finite dimensional and control optimization problems*, Ph.D. thesis, University of London, London, 1978.

[13] D. Q. MAYNE AND E. POLAK, *A superlinearly convergent algorithm for constrained optimization problems*, Math. Prog. Stud., 16 (1982), pp. 45–61.

[14] J. NOCEDAL AND M. OVERTON, *Projected Hessian updating algorithms for nonlinearly constrained optimization*, SIAM J. Numer. Anal., 22 (1985), pp. 821–850.

[15] E. O. OMOJOKUN, *Trust-region strategies for optimization with nonlinear equality and inequality constraints*, Ph.D. thesis, Department of Computer Science, University of Colorado, Boulder, 1989.

[16] 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.

[17] ———, *Variable metric methods for constrained optimization*, in Mathematical Programming: The state of the art, M. G. A. Bachem and B. Korte, eds., Springer-Verlag, Berlin, New York, 1983, pp. 288–311.

[18] ———, *On the global convergence of trust-region algorithms for unconstrained optimization*, Math. Programming, 29 (1984), pp. 297–303.

[19] M. J. D. POWELL AND Y. YUAN, *A trust-region algorithm for equality constrained optimization*, Math. Programming, 49 (1991), pp. 189–211.

[20] R. TAPIA, *On secant update for use in general constrained optimization*, Math. Comp., 51 (1988), pp. 181–202.

[21] 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.

[22] P. TOINT, *Global convergence of a class of trust-region methods for non-convex minimization in Helbert spaces*, IMA J. Numer. Anal., 8 (1988), pp. 231–252.

[23] K. A. WILLIAMSON, *A robust trust-region algorithm for nonlinear programming*, Ph.D. thesis, Department of Mathematical Sciences, Rice University, Houston, TX, 1990.

[24] J. Z. ZHANG AND D. T. ZHU, *Projected quasi-Newton algorithm with trust region for constrained optimization*, J. Optim. Theory Appl., 67 (1990), pp. 369–393.

SIAM J. OPTIMIZATION
Vol. 5, No. 2, pp. 379–407, May 1995

008

A CLASS OF TRUST REGION METHODS FOR NONLINEAR NETWORK OPTIMIZATION PROBLEMS*

A. SARTENAER†

Abstract. We describe the results of a series of tests upon a class of new methods of trust region type for solving the nonlinear network optimization problem. The trust region technique considered is characterized by the use of the infinity norm and of inexact projections on the network constraints. The results are encouraging and show that this approach is particularly useful in solving large-scale nonlinear network optimization problems, especially when many bound constraints are expected to be active at the solution.

Key words. nonlinear optimization, nonlinear network optimization, trust region methods, truncated Newton methods, numerical results

AMS subject classifications. 90C30, 90C35, 65K05

1. Introduction.

We consider the problem

$$\begin{array}{ll} \min_{x\in\mathbf{R}^n} & f(x) \\ \text{subject to} & Ax = b \\ & l \le x \le u, \end{array} \tag{1.1}$$

where $f : \mathbf{R}^n \to \mathbf{R}$ is a twice continuously differentiable partially separable function, A is an $m \times n$ node-arc incidence matrix, $b \in \mathbf{R}^n$ and satisfies $\sum_{i=1}^{m} b_i = 0$, and l and $u \in \mathbf{R}^n$.

Many algorithms for solving the nonlinear network problem (1.1) have been proposed (see [1], [3], [10], [11], [13], [21], and [22], for instance), most of them being of the active set variety. In particular, a sequence of problems is solved for which a subset of the variables (the active set) is fixed at bounds and the objective function is minimized with respect to the remaining variables. Such algorithms typically use linesearches to enforce convergence. A significant drawback of these methods, especially for large-scale problems, is that the active sets are allowed to change slowly and many iterations are necessary to correct a bad initial choice.

In this paper, we propose a new algorithm of *trust region* type that allows rapid changes in the active set. This algorithm is an adaptation of the one proposed by Conn et al. in [4] for which we have already produced a general convergence theory. At iteration k of the algorithm, we define a local model of the objective function at the current iterate, x_k say, and a *region* surrounding x_k where we *trust* this model. The algorithm then finds, in this region, a candidate for the next iterate that sufficiently reduces the value of the model. If the function value calculated at this point matches its predicted value closely enough, then the new point is accepted as the next iterate and the trust region is possibly enlarged. Otherwise, the point is rejected and the trust region size is decreased.

The determination of a candidate for the next iterate requires the computation of a *Generalized Cauchy Point* (GCP) that expands the notion of a Cauchy Point to problems with general convex constraints (see [4]). This has the double advantage of allowing significant changes in the active set at each iteration and permitting

* Received by the editors July 12, 1993; accepted for publication (in revised form) January 12, 1994.

† Department of Mathematics, Facultés Universitaires N.D. de la Paix, 61 rue de Bruxelles, B-5000, Namur, Belgium (as@math.fundp.ac.be).

the extension of well-known convergence results for trust region methods applied to unconstrained problems (see [18]) and to simple bound constrained problems (see [7]).

The calculation of a suitable GCP, which makes use of the first order information, is performed by solving a sequence of *linear network* problems. The GCP is thereafter refined to calculate a candidate for the next iterate using the second order information through a *truncated conjugate gradient* technique. This technique, as well as the linear solver used for the GCP, takes advantage of the network structure in the constraints of problem (1.1) by combining a data structure of the type proposed by Bradley, Brown, and Graves [2] with a partition of the variables similar to that proposed by Murtagh and Saunders [19] implemented in MINOS, also making use of *variable reduction matrices*. Moreover, we use the concept of *maximal basis* that is especially well suited in our context to allow adequate adaptation of the theory developed in [4] for the active set identification strategy. Note that most of the aforementioned techniques are equally exploited in successful existing large-scale nonlinear network solvers, such as GENOS [1] and NLPNET [10].

Section 2 of this paper gives a general introduction to the framework of our algorithm, together with a detailed description of the computation of a GCP and of a candidate for the next iterate. The optimality conditions and the specific algorithm are also presented in this section. Section 3 reports and comments on some numerical experiments, and includes a comparison with an existing available specialized software for the same problem. Finally some conclusions and perspectives are outlined in §4.

2. Description of the algorithm.

2.1. The basic algorithm. As already mentioned, our algorithm is of trust region type and the description given here is a special case of the general framework presented in [4], adapted to the solution of problem (1.1). We first introduce the following concepts. The *feasible region* for problem (1.1) is the polyhedral set

$$X = \{x \in \mathbf{R}^n | Ax = b \text{ and } l \leq x \leq u\},$$

and any point x in the feasible region is called *feasible*. We define the *active set with respect to the vectors l and u at the feasible point x* as the index set

$$\mathcal{A}(x, l, u) = \{i \in \{1, \ldots, n\} | [x]_i = [l]_i \text{ or } [x]_i = [u]_i\},$$

where $[v]_i$ denotes the ith component of the vector v.

At the kth stage of the algorithm, we suppose that we have a feasible point x_k, the exact gradient $\nabla f(x_k)$ (denoted g_k) and the exact Hessian $\nabla^2 f(x_k)$ (denoted H_k) of the objective function at x_k. We also require a scalar $\Delta_k > 0$ for the trust region radius, and choose the quadratic model of the form

$$m_k(x_k + s) \stackrel{\text{def}}{=} f(x_k) + g_k^T s + \frac{1}{2} s^T H_k s$$

to approximate the objective function around x_k. A trial feasible step s_k is then computed by approximately solving the trust region problem

$$\begin{array}{ll} \min_{s \in \mathbf{R}^n} & m_k(x_k + s) \\ \text{subject to} & As = 0 \\ & l \leq x_k + s \leq u \\ \text{and} & \|s\| \leq \Delta_k, \end{array} \tag{2.1}$$

where $\|\cdot\|$ is a suitable chosen norm. The updates of the iterate x_k and of Δ_k are done using the same criteria of acceptance as in trust region methods for unconstrained or bound constrained minimization (see [18] and [7]). That is,

$$x_{k+1} = \begin{cases} x_k + s_k & \text{if } \rho_k > \eta_1, \\ x_k & \text{if } \rho_k \leq \eta_1, \end{cases}$$

and

$$\Delta_{k+1} = \begin{cases} 2\Delta_k & \text{if } \rho_k \geq \eta_2, \\ \Delta_k & \text{if } \eta_1 < \rho_k < \eta_2, \\ \frac{1}{2}\min(\|s_k\|, \Delta_k) & \text{if } \rho_k \leq \eta_1, \end{cases} \tag{2.2}$$

where

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}$$

represents the ratio of the achieved to the predicted reduction of the objective function and $0 < \eta_1 < \eta_2 < 1$ are appropriate numbers. It now remains to describe our approximate solution of (2.1).

The choice of the infinity norm for the trust region constraint in problem (2.1) allows us to replace the bound constraints and the trust region constraint in this problem by the bound constraints

$$\max([l]_i, [x_k]_i - \Delta_k) \stackrel{\text{def}}{=} [l_k]_i \leq [x_k + s]_i \leq [u_k]_i \stackrel{\text{def}}{=} \min([u]_i, [x_k]_i + \Delta_k) \tag{2.3}$$

for $i = 1, \ldots, n$. Problem (2.1) then becomes

$$\begin{array}{ll} \min_{s \in \mathbf{R}^n} & m_k(x_k + s) \\ \text{subject to} & As = 0 \\ & l_k \leq x_k + s \leq u_k. \end{array} \tag{2.4}$$

In order to satisfy the global convergence theory developed in [4], we need to find a feasible point $x_k + s_k$ within the trust region at which the value of the model function is no larger than its value at the GCP. This GCP, denoted x_k^C, is found through a *projected search* on the model along an *approximation* of the projected gradient path (i.e., the projection of the gradient on the feasible set). Note that the determination of the active set (the set of variables that are to be fixed at one of their bounds during the current iteration) takes place when finding the GCP. Since no restriction on the number of variables moving into or out of the active set is imposed from one iteration to the other, rapid changes may occur in the active set. This is extremely useful in large-scale optimization problems since the number of iterations required to find the correct active set may hence be considerably smaller than the number of active bounds at the solution. Subsequently we use second order information to refine the GCP and provide a fast ultimate rate of convergence. Therefore, following Murtagh and Saunders [19], we first partition the matrix A as

$$A = \begin{pmatrix} B & S & N \end{pmatrix},$$

where the $m \times m$ submatrix B is nonsingular, and define

$$\{1, \ldots, n\} = \mathcal{B} \cup \mathcal{S} \cup \mathcal{N}, \tag{2.5}$$

the induced partition of the variable indices. For a node-arc incidence matrix, Dantzig [9, p. 356] has shown that the arcs whose indices are in $\mathcal{B}$ form a *spanning tree* of the network. (These arcs are called *basic arcs* while the arcs whose indices are in $\mathcal{S}$ and $\mathcal{N}$ are called *superbasic arcs* and *nonbasic arcs*, respectively.) In that case, it is worth using a specialized data structure of the type proposed by Bradley, Brown, and Graves [2] that allows us to store and update the basis of the network (i.e., the spanning tree) in a very efficient manner.

According to assumption (AS.8) in [4],

$$\mathcal{A}(x_k^C, l, u) \subseteq \mathcal{A}(x_k + s_k, l, u) \tag{2.6}$$

— the variables of x_k^C that are at a bound must remain fixed when finding a better approximation of a minimizer of (2.4). We then set, at each iteration k,

$$\mathcal{N} = \mathcal{A}(x_k^C, l_k, u_k) \setminus \mathcal{B} \quad \text{and} \quad \mathcal{S} = \{1, \ldots, n\} \setminus (\mathcal{B} \cup \mathcal{N}). \tag{2.7}$$

Since $j \notin \mathcal{A}(x_k^C, l_k, u_k) \Longrightarrow j \notin \mathcal{A}(x_k^C, l, u)$, this choice for $\mathcal{N}$ produces a correct set for $\mathcal{S}$ according to assumption (2.6), namely, an index set of arcs $\notin \mathcal{B}$ strictly between the bounds l and u. Note that this choice imposes more than the assumption requires, since it further fixes the components of the GCP that are on the trust region boundary (even if they are not at a bound l or u), which seems quite natural.

Using a *variable reduction matrix* Z as proposed by Murtagh and Saunders [19] (that is, a matrix formed by column vectors that belong to the nullspace of A, yielding the relation $AZ = 0$) and choosing

$$Z = \begin{pmatrix} -B^{-1}S \\ I \\ 0 \end{pmatrix}, \tag{2.8}$$

we then solve approximately problem (2.4) by applying a conjugate gradient algorithm, starting from the GCP, to the equation

$$Z^T H_k Z \, [s]_{\mathcal{S}} = -Z^T g_k.$$

Let $[s_k]_{\mathcal{S}}$ be the approximation found. We define the full trial step s_k by $s_k = ([s_k]_{\mathcal{B}}, [s_k]_{\mathcal{S}}, [s_k]_{\mathcal{N}})$, where $[s_k]_{\mathcal{B}}$ and $[s_k]_{\mathcal{N}}$ satisfy

$$B \, [s_k]_{\mathcal{B}} = -S \, [s_k]_{\mathcal{S}} \tag{2.9}$$

and

$$[s_k]_{\mathcal{N}} = 0.$$

We defer to §2.3 the management of the constraints $l_k \leq x_k + s \leq u_k$ during the conjugate gradient schemes solving problem (2.4). Note that the matrix Z in (2.8) exhibits a useful structure [16]. Indeed, the jth column of Z corresponds to the cycle formed by adding the jth superbasic arc to the spanning tree associated with the basis. This cycle can be decomposed in the jth superbasic arc, joining nodes e and f, say, and its associated *flow augmenting path* (also called *basic equivalent path*), which is the (unique) path between nodes e and f belonging to the tree. Let β_j be the set

of indices of the arcs of this path. The element (i,j) of $-B^{-1}S$ is then given by

$$(2.10)\quad \left[-B^{-1}S\right]_{ij} = \begin{cases} +1 & \text{if } i \in \beta_j \text{ and the } i\text{th basic arc has an orientation identical to that of the } j\text{th superbasic arc in the cycle,} \\ -1 & \text{if } i \in \beta_j \text{ and the } i\text{th basic arc has an orientation opposite to that of the } j\text{th superbasic arc in the cycle,} \\ 0 & \text{if } i \notin \beta_j. \end{cases}$$

This special structure allows for a compact storage of the matrix Z, as well as for very efficient techniques for computing products that involve this matrix or its transpose (see [22] for more details). Moreover, this last structure is analogous to that of the matrix $-B^{-1}N$ that arises in the computation of the Lagrange multiplier estimates,

$$[g_k]_{\mathcal{N}} - N^T B^{-T} [g_k]_{\mathcal{B}},$$

with the only difference being that the flow augmenting path is now associated with a nonbasic variable instead of a superbasic one.

In order to be sure that assumption (2.6) holds, we further need to impose that the basic arcs whose indices are in $\mathcal{A}(x_k^C, l, u)$ remain fixed when finding the candidate step s_k. But this can be automatically induced by using the concept of *maximal spanning tree*, as introduced by Dembo and Klincewicz in [12], that is a spanning tree which has a maximal number of arcs whose flows are strictly between the bounds l and u (see also [23]). With such a spanning tree, a basic arc whose flow is at a bound is not allowed to belong to the flow augmenting path of a *free* arc (that is an arc whose flow is strictly between its bounds), since otherwise, the replacement of this basic arc with the free one would increase the number of free arcs in the spanning tree, in contradiction with its property of maximality. Given the way the index sets $\mathcal{N}$ and $\mathcal{S}$ are defined in (2.7), every superbasic arc is ensured to be strictly between the bounds l and u, and the use of maximal spanning trees therefore prevents any basic arc that belongs to the flow augmenting path of a superbasic arc to be at one of its bounds. Consequently, since a basic component of s_k computed from (2.9) may be nonzero *only if* its corresponding arc belongs to the flow augmenting path of at least one superbasic arc (see (2.10)), we are sure that the only basic arcs allowed to change during the process are those that are strictly between the bounds l and u. Moreover, using the same argument, we force the basic arcs that are on the trust region boundary to remain fixed by imposing that the spanning tree be maximal *also with respect* to the bounds l_k and u_k (that is to have a maximal number of arcs whose flows are strictly between the bounds l_k and u_k).

Under condition (2.6) and a nondegeneracy condition, the strategy described above is sufficient to ensure that the correct active set is identified after a finite number of iterations (see [4]). We now give, in the next two sections, more details on the computations of the GCP x_k^C and the trial step s_k.

2.2. The Generalized Cauchy Point. Following [4], in order to find a GCP, we first need to determine an approximation of a suitable point on the projected gradient path. By this, we mean a feasible point $x_k^C = x_k + s_k^C$ inside the trust region that satisfies the inequality

$$(2.11)\quad g_k^T s_k^C \leq -\mu_3 \alpha_k(t_k)$$

for some fixed $\mu_3 \in (0,1]$ and $t_k > 0$. Here $\alpha_k(t_k) > 0$ represents the magnitude of the maximum decrease of the linearized model achievable on the intersection of the feasible domain with a box of radius t_k centered at x_k:

$$-\alpha_k(t_k) \stackrel{\text{def}}{=} \begin{array}{ll} \min_{d\in\mathbf{R}^n} & g_k^T d \\ \text{subject to} & Ad = 0 \\ & l_{t_k} \leq d \leq u_{t_k}, \end{array} \tag{2.12}$$

where

$$[l_{t_k}]_j \stackrel{\text{def}}{=} \max([l]_j, [x_k]_j - t_k) - [x_k]_j$$

and

$$[u_{t_k}]_j \stackrel{\text{def}}{=} \min([u]_j, [x_k]_j + t_k) - [x_k]_j$$

for $j = 1, \ldots, n$. Furthermore, this point must satisfy the two Goldstein-like conditions

$$m_k(x_k + s_k^C) \leq m_k(x_k) + \mu_1 g_k^T s_k^C \tag{2.13}$$

and

$$\text{either } t_k \geq \min[\nu_1 \Delta_k, \nu_2] \text{ or } m_k(x_k + s_k^C) \geq m_k(x_k) + \mu_2 g_k^T s_k^C, \tag{2.14}$$

where $0 < \mu_1 < \mu_2 < 1$, $\nu_1 \in (0,1)$ and $\nu_2 \in (0,1]$ are appropriate constants.

The GCP *Algorithm* given in [4] is a model algorithm for computing a GCP that verifies conditions (2.11), (2.13) and (2.14). It is iterative and uses bisection. At each iteration i, given a bisection parameter value $t_i > 0$, it computes first a candidate step s_i that satisfies condition (2.11) (with $t_k = t_i$ and $s_k^C = s_i$), checking then conditions (2.13) and (2.14) (with $t_k = t_i$ and $s_k^C = s_i$), until either an acceptable GCP is found or two candidates $x_k + s_k^l$ and $x_k + s_k^u$ are known that violate condition (2.13) and condition (2.14). Thus, if an acceptable GCP is not yet found, the algorithm carries out a simple bisection linesearch on the model along a particular path between these two points, yielding a suitable GCP in a finite number of iterations. This particular path, called the *restricted path*, is obtained by applying the so-called *restriction operator*,

$$R_{x_k}[y] \stackrel{\text{def}}{=} \arg \min_{z \in [x_k, y] \cap X} \|z - y\|_2,$$

where $[x_k, y]$ is the segment between x_k and y, on the piecewise linear path consisting of the segment $[x_k + s_k^l, x_k + s_k^p]$ followed by $[x_k + s_k^p, x_k + s_k^u]$, where

$$s_k^p = \max\left[1, \frac{\|s_k^u\|_\infty}{\|s_k^l\|_\infty}\right] s_k^l.$$

This restricted path is an approximation of the unknown projected gradient path between the points $x_k + s_k^l$ and $x_k + s_k^u$ in the sense that each point on this path satisfies condition (2.11) for some $t_k > 0$. It also closely follows the boundary of the feasible domain, as does the projected gradient path. We refer the reader to [4] for a detailed discussion of these concepts.

In order to perform the simple bisection linesearch along the restricted path, a call to the RS *Algorithm* given below is made in the GCP Algorithm. The inner iterations of Algorithm RS are denoted by the index j.

RS ALGORITHM.

Step 0. Initialization. Set $\delta_p = \|s_k^p - s_k^l\|_2$, $\delta_u = \delta_p + \|s_k^u - s_k^p\|_2$, $l_0 = 0$, $u_0 = \delta_u$ and $j = 0$. Then define $\delta_0 = \frac{1}{2}(l_0 + u_0)$.

Step 1. Compute the point on the restricted path corresponding to δ_j.

Step 1.0. Compute the step from x_k to the piecewise linear path. Set

$$s_j \stackrel{\text{def}}{=} \begin{cases} \frac{\delta_j}{\delta_p} s_k^p + (1 - \frac{\delta_j}{\delta_p}) s_k^l & \text{if } \delta_j \leq \delta_p, \\ \frac{\delta_j - \delta_p}{\delta_u - \delta_p} s_k^u + (1 - \frac{\delta_j - \delta_p}{\delta_u - \delta_p}) s_k^p & \text{if } \delta_j \geq \delta_p. \end{cases}$$

Step 1.1. Calculate the smallest value of α such that $x_k + \alpha s_j$ hits a bound. Set

$$\alpha^* = \min \left[\min_{\{i \in \{1,\ldots,n\} | [s_j]_i < 0\}} \left| \frac{[x_k]_i - [l]_i}{[s_j]_i} \right|, \min_{\{i \in \{1,\ldots,n\} | [s_j]_i > 0\}} \left| \frac{[x_k]_i - [u]_i}{[s_j]_i} \right| \right].$$

Step 1.2. Compute the point on the restricted path. Set

$$\alpha_j = \min[1, \alpha^*]$$

and

$$x_j = x_k + \alpha_j s_j.$$

Step 2. Check the stopping conditions. If

$$m_k(x_j) > m_k(x_k) + \mu_1 g_k^T (x_j - x_k), \tag{2.15}$$

then set

$$l_{j+1} = l_j \quad \text{and} \quad u_{j+1} = \delta_j,$$

and go to Step 3. Else, if

$$m_k(x_j) < m_k(x_k) + \mu_2 g_k^T (x_j - x_k), \tag{2.16}$$

then set

$$l_{j+1} = \delta_j \quad \text{and} \quad u_{j+1} = u_j,$$

and go to Step 3; else (that is if both (2.15) and (2.16) fail), set $x_k^C = x_j$ and STOP.

Step 3. Choose the next parameter value by bisection. Increment j by one, set

$$\delta_j = \frac{1}{2}(l_j + u_j)$$

and go to Step 1.

Note that the point x_j calculated at Step 1 satisfies the constraint $Ax = b$ and minimizes the distance from $x_k + s_j$ in the direction $-s_j$ while satisfying the constraints $l \leq x_j \leq u$, as expected.

As mentioned before, at a given iteration i of the GCP Algorithm we first compute a candidate step s_i that satisfies condition

$$g_k^T s_i \leq -\mu_3 \alpha_k(t_i), \tag{2.17}$$

where t_i is the current bisection parameter value. We obtain s_i by applying a simplex-like algorithm to problem (2.12) (where t_k is replaced by t_i) and by stopping this algorithm as soon as an admissible iterate d_ℓ has been found that verifies

$$|g_k^T d_\ell| \geq \mu_3 \min_{r=1,\ldots,\ell} \left[l_{t_i}^T A^T \pi_r + (u_{t_i} - l_{t_i})^T \mu_r - g_k^T l_{t_i} \right], \tag{2.18}$$

where

$$\pi_r = [g_k]_{\mathcal{B}_r}^T B_r^{-1} \quad \text{and} \quad [\mu_r]_j = \max(0, \pi_r A e_j - [g_k]_j) \quad (j = 1, \ldots, n), \tag{2.19}$$

where B_r is the admissible basis associated with some previous candidate d_r, $[g_k]_{\mathcal{B}_r}$ is the basic part of g_k and e_j is the jth vector of the canonical basis of $\mathbf{R}^n$. Indeed, the right-hand side of condition (2.18) is an upper bound on the value of $\mu_3 \alpha_k(t_i)$ and (2.18) thus implies condition (2.17) for $s_i = d_\ell$ (see [4] for more details).

Now we give the GCP Algorithm itself. Its inner iterations are denoted by the index i.

GCP ALGORITHM.

Step 0. Initialization. Choose $\lambda \in (0,1)$. Set $l_0 = 0$, $u_0 = \Delta_k$, $s_0^l = 0$ and $i = 0$. Also choose s_0^u an arbitrary vector such that $\|s_0^u\|_\infty > \Delta_k$ and an initial parameter $t_0 \in (0, \Delta_k]$.

Step 1. Compute a candidate step. Compute a vector s_i such that

$$As_i = 0 \quad \text{and} \quad l_{t_i} \leq s_i \leq u_{t_i},$$

and

$$g_k^T s_i \leq -\mu_3 \alpha_k(t_i).$$

Step 2. Check the stopping rules on the model and step. If

$$m_k(x_k + s_i) > m_k(x_k) + \mu_1 g_k^T s_i, \tag{2.20}$$

then set

$$u_{i+1} = t_i \quad s_{i+1}^u = s_i$$

and

$$l_{i+1} = l_i \quad s_{i+1}^l = s_i^l,$$

and go to Step 3. Else, if

$$m_k(x_k + s_i) < m_k(x_k) + \mu_2 g_k^T s_i \tag{2.21}$$

and

$$t_i < \min[\nu_1 \Delta_k, \nu_2], \tag{2.22}$$

then set

$$u_{i+1} = u_i \quad s_{i+1}^u = s_i^u$$

and

$$l_{i+1} = t_i \quad s_{i+1}^l = s_i,$$

and go to Step 3. Else (that is if (2.20) and either (2.21) or (2.22) fail), then set

$$x_k^C = x_k + s_i$$

and STOP.

Step 3. Define a new trial step by bisection. We distinguish two mutually exclusive cases.

Case 1. $s_{i+1}^l = s_0^l$ or $s_{i+1}^u = s_0^u$. Set

$$t_{i+1} = \lambda(l_{i+1} + u_{i+1}),$$

increment i by one and go to Step 1.

Case 2. $s_{i+1}^l \neq s_0^l$ and $s_{i+1}^u \neq s_0^u$. Set

$$s_k^l = s_{i+1}^l \quad \text{and} \quad s_k^u = s_{i+1}^u,$$

define

$$s_k^p = \max\left[1, \frac{\|s_k^u\|_\infty}{\|s_k^l\|_\infty}\right] s_k^l,$$

apply the RS Algorithm to find a GCP x_k^C and STOP.

For the computation of s_i in Step 1, we have implemented a self-contained routine that uses the same data structure as that representing problem (1.1) and is a particular implementation of the simplex algorithm specialized to network problems, along the lines described in [2], [16], and [17]. This routine includes at each iteration the computation of the vectors π_r and μ_r from (2.19) as well as the update of the upper bound on the value of $\mu_3\alpha_k(t_i)$ given in (2.18), and stops as soon as an appropriate inexact solution is computed. This implementation provides in particular a *total pricing* routine (see [17]) for seeking a nonbasic candidate to enter the basis, since the vector μ_r must be totally evaluated at each iteration. In order to compare the performances of this last algorithm with one that completely solves problem (2.12) (as required if μ_3 is set to 1), we have also implemented a routine that finds the exact solution of (2.12), without adding the extra burden of computing the quantities required for an approximate solution (namely μ_r and the upper bound on $\mu_3\alpha_k(t_i)$), but rather using a *partial pricing* routine to select a nonbasic arc to be moved. More precisely, we select sets of thirty variables taken at regular intervals among the nonbasic variables and test each variable in the successive sets until a candidate to enter the basis is found.

We have left unspecified the parameter $\lambda \in (0,1)$ in the GCP Algorithm (see Case 1 of Step 3) in order to test the effect of varying its value. Indeed, in order to avoid an excessive number of computations of a candidate step s_i in Step 1 — the most costly calculation of the algorithm — it could be worthwhile to accelerate the branching to the second case of Step 3 by choosing a smaller value for λ than the classical 0.5.

The above algorithm for the calculation of a GCP has the advantage of avoiding the repeated computation of the projection on the feasible domain, which is a *quadratic* program. Instead we repeatedly compute an approximate solution of *linear* programs. This can be related to the convex combination algorithm originally suggested by Frank and Wolfe (see [20]) for solving quadratic programming problems with linear constraints. The Frank and Wolfe algorithm is based on finding a descent direction by

minimizing a linear approximation to the function subject to the linear constraints. A linesearch on the quadratic objective function along the descent direction found is then performed to determine the next iterate.

2.3. The candidate step s_k. In this section, we develop an algorithm for solving problem (2.4), or more precisely, for finding an approximate solution to the reduced equation

$$Z^T H_k Z\,[s]_{\mathcal{S}} = -Z^T g_k. \tag{2.23}$$

The strategy considered uses a *truncated conjugate gradient* technique, starting from the GCP, which handles the bound constraints

$$l_k \leq x_k + s \leq u_k \tag{2.24}$$

during the conjugate gradient iteration.

The conjugate gradient method is well suited to solving (2.23) without forming the reduced Hessian $Z^T H_k Z$ (which may be considerably denser than both Z and H_k), since it only requires matrix-vector products of the form $Z^T H_k Z v$. These products can be computed relatively cheaply by forming, in turn, $v_1 = Zv$, $v_2 = H_k v_1$ and $v_3 = Z^T v_2$. This is all the cheaper here as a sparse Hessian H_k and a sparse representation of Z can be stored, due to the partially separable structure of the objective f and the structure of the matrix Z (see (2.8) and (2.10)).

The TCG *Algorithm* terminates the conjugate gradient iteration in the solution of (2.23) at the point $x = x_k + s$ whenever:

- The *reduced residual norm* at x (i.e., the norm of the reduced gradient of the model at the point x) is small enough, that is

$$\|Z^T H_k Z\,[s]_{\mathcal{S}} + Z^T g_k\|_2 \leq \eta_k,$$

where

$$\eta_k = \max\left[\sqrt{\epsilon_M}, \min[0.01, \|Z^T g_k\|_2]\right] \|Z^T g_k\|_2 \tag{2.25}$$

and ϵ_M is the relative machine precision. This stopping rule allows for better and better approximations to the solution of the Newton equation (2.23) when close to a local minimizer of problem (1.1) and is the essence of a *truncated* Newton scheme.

- $\mathcal{S} = \emptyset$, i.e., there is no way to better refine the current solution x.
- A direction of negative curvature has been encountered.
- An excessive number of iterations has been taken.

The main characteristics of the TCG Algorithm are the following. At each recurrence of a conjugate gradient iteration, the TCG Algorithm will verify if feasibility with respect to the bound constraints (2.24) is still respected. In the case where a bound is reached, the conjugate gradient iteration is temporarily stopped and the current maximal spanning tree is possibly updated, depending on the type of bound encountered. Thereafter, the active set is updated according to the decomposition (2.5), where the index set $\mathcal{S}$ corresponds to the arcs whose current flow is strictly between the bounds l_k and u_k. The conjugate gradient iteration is then possibly restarted.

Now we specify the TCG Algorithm in more detail. In the description given below, we denote by r the *residual* vector $-(g_k + H_k s)$. For a given vector v and

a given partition $\mathcal{B} \cup \mathcal{S} \cup \mathcal{N}$ of the set $\{1, \ldots, n\}$, we also define the corresponding *reduced* vector v^r as the vector of $\mathbf{R}^n$ defined componentwise by

$$[v^r]_i \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } i \in \mathcal{B} \cup \mathcal{N}, \\ [Z^T v]_j & \text{if } i \text{ is the } j\text{th element of } \mathcal{S} \text{ (that is the } j\text{th superbasic arc).} \end{cases} \tag{2.26}$$

TCG ALGORITHM.

Step 0. Initialization. The GCP, $x_k^C = x_k + s_k^C$, and an initial maximal spanning tree whose indices define the set $\mathcal{B}$ are given. Set $x = x_k^C$ and $r = -(g_k + H_k s_k^C)$.

Step 1. The conjugate gradient iteration. As long as there exist arcs $\notin \mathcal{B}$ that are strictly between the bounds l_k and u_k, we continue the conjugate gradient iteration to further minimize the reduced model of the objective function at the point x_k. Each time a restarting is considered, we redefine the index set $\mathcal{S}$ and (2.23) accordingly, and solve this last equation starting from the current point x, until one of the stopping rules mentioned above is satisfied or a bound is encountered.

Step 1.0. Define the active set. Set

$$\mathcal{N} = \mathcal{A}(x, l_k, u_k) \setminus \mathcal{B}$$

and deduce $\mathcal{S}$ from the partition (2.5). If $\mathcal{S} = \emptyset$, go to Step 2. Otherwise, compute the matrices B, S, N, and Z from the partition (2.5) and (2.8).

Step 1.1. Restart the conjugate gradient iteration. Now that the subspace where the minimization can take place is fixed (namely, the space spanned by the superbasic variables indexed by $\mathcal{S}$), we can proceed with the conjugate gradient iteration.

Step 1.1.0. Initialization before restarting. Compute the *reduced residual* r^r from (2.26) and the relative accuracy level η_k from (2.25). Set $d = 0$, $\beta = 0$ and $\rho_2 = \|r^r\|_2^2$.

Step 1.1.1. Test for the required accuracy. If $\rho_2 \leq \eta_k^2$, go to Step 2.

Step 1.1.2. Conjugate gradient recurrences. Compute

$$[d]_{\mathcal{S}} = [r^r]_{\mathcal{S}} + \beta [d]_{\mathcal{S}},$$

$[d]_{\mathcal{B}}$ from

$$B [d]_{\mathcal{B}} = -S [d]_{\mathcal{S}}$$

and set $d = ([d]_{\mathcal{B}}, [d]_{\mathcal{S}}, 0)$. Compute the vectors $y = H_k d$, y^r from (2.26), and the curvature $\gamma = d^T y$. Find α_1, the largest value of α for which $l_k \leq x + \alpha d \leq u_k$. If $\gamma \leq 0$, then set

$$x = x + \alpha_1 d$$

and go to Step 1.2. Otherwise, calculate $\alpha_2 = \rho_2 / \gamma$. If $\alpha_2 \geq \alpha_1$, then set

$$x = x + \alpha_1 d,$$

$$r = r - \alpha_1 y,$$

and go to Step 1.2. Otherwise, set

$$x = x + \alpha_2 d,$$

$$r = r - \alpha_2 y,$$

$$r^r = r^r - \alpha_2 y^r,$$

$$\rho_1 = \rho_2,$$

$$\rho_2 = \|r^r\|_2^2,$$

$$\beta = \frac{\rho_2}{\rho_1},$$

and go to Step 1.1.1.

Step 1.2. Update the maximal spanning tree. Update the index set $\mathcal{B}$ in order to keep a maximal spanning tree (see further on). If $\gamma \leq 0$, go to Step 2. Otherwise, go to Step 1.

Step 2. Termination of the conjugate gradient iteration. Set

$$s_k = x - x_k.$$

In order to maintain a maximal spanning tree when a pivoting step is required, we need to take into account the bound constraints of both problem (1.1) and problem (2.4) (the latter varying from one iteration to the other, depending on the trust region size), since the spanning tree must remain maximal with respect to the bounds l and u *and* l_k and u_k. As illustrated by the following example, it is not sufficient to only consider the maximality of the spanning tree with respect to the bounds l_k and u_k. Suppose indeed, when moving to a point x, that the basic arc of index i hits a bound such that not only $i \in \mathcal{A}(x, l_k, u_k)$, but also $i \in \mathcal{A}(x, l, u)$, and that no arc $j \notin \mathcal{B}$ satisfying $[l_k]_j < [x]_j < [u_k]_j$ may be found to pivot with. In that case, if the current maximal spanning tree remains unmodified and if there exists an arc of index j, say, such that $j \in \mathcal{A}(x, l_k, u_k)$, $i \in \beta_j$ (i.e., arc i belongs to the flow augmenting path of arc j), but $j \notin \mathcal{A}(x, l, u)$, this spanning tree will not be maximal any more with respect to the bounds l and u as soon as the trust region constraint vanishes from (2.4) or is modified. Therefore, based on the observation that $i \in \mathcal{A}(x, l, u) \Longrightarrow i \in \mathcal{A}(x, l_k, u_k)$ and $j \notin \mathcal{A}(x, l_k, u_k) \Longrightarrow j \notin \mathcal{A}(x, l, u)$, we consider the following algorithm for maintaining a maximal spanning tree.

If, for some $i \in \mathcal{B}$,

$$i \in \mathcal{A}(x, l, u)$$

or

$$i \notin \mathcal{A}(x, l, u) \quad \text{and} \quad i \in \mathcal{A}(x, l_k, u_k),$$

then determine (if possible) $j \notin \mathcal{B}$ such that $i \in \beta_j$, $\min \left[| [x]_j - [l_k]_j |, | [x]_j - [u_k]_j | \right]$ is maximum and either

$$j \notin \mathcal{A}(x, l, u)$$

or

$$j \notin \mathcal{A}(x, l_k, u_k),$$

respectively. Then redefine the set $\mathcal{B}$ by

$$\mathcal{B} = \mathcal{B} \setminus \{i\} \cup \{j\}$$

and update the submatrix B accordingly, performing a pivoting step as described in [2].

Note that the choice of j in the above description is intended to allow larger steps in the next search, which may result in a more useful decrease of the cost function (see [21]).

2.4. Optimality test. We consider that optimality for problem (1.1) is reached whenever the objective function cannot be further reduced at the current iterate x_k. This may be checked in the following manner.

• Select the arcs that allow for a possible improvement. This amounts to finding the arcs $\notin \mathcal{B}$ which are either strictly between the bounds l and u or at one of these bounds, but whose release may induce a decrease in the objective function. (These last arcs are found through an examination of the corresponding Lagrange multipliers.)

• Remove the so-called *blocked* arcs [11], that is the arcs at a bound l or u whose release causes the immediate violation of another bound l or u for one of the arcs of their flow augmenting path. This may be easily verified using the following test.

If arc j is such that, either

$$[x_k]_j = [u]_j \quad \text{and} \quad \exists i \in \beta_j \text{ such that}$$

$$\left([-B^{-1}N]_{ij} = 1 \text{ and } [x_k]_i = [l]_i\right) \text{ or } \left([-B^{-1}N]_{ij} = -1 \text{ and } [x_k]_i = [u]_i\right),$$

or

$$[x_k]_j = [l]_j \text{ and } \exists i \in \beta_j \text{ such that}$$

$$\left([-B^{-1}N]_{ij} = 1 \text{ and } [x_k]_i = [u]_i\right) \text{ or } \left([-B^{-1}N]_{ij} = -1 \text{ and } [x_k]_i = [l]_i\right),$$

then it is blocked.

(Note that this situation cannot occur for the arcs that are strictly between the bounds l and u, because of the properties of the maximal spanning tree.)

• Denoting by $\mathcal{S}$ the set of indices obtained from the above selection, deduce the set $\mathcal{N}$ from the partition (2.5) and define an active set accordingly.

• Check if the current iterate x_k is optimal on this active set, that is, if the corresponding reduced gradient at x_k is null.

This framework may be summarized by the following algorithm.

OT ALGORITHM.

Step 0. $\mathcal{B}$ is given. Set $\mathcal{S} = \emptyset$ and $\mathcal{N} = \{1, \ldots, n\} \setminus \mathcal{B}$.

Step 1. For each $j \in \{1, \ldots, n\} \setminus \mathcal{B}$, redefine $\mathcal{S}$ and $\mathcal{N}$ in the following way. If

$$j \notin \mathcal{A}(x_k, l, u),$$

then redefine $\mathcal{S}$ and $\mathcal{N}$ by

$$\mathcal{S} = \mathcal{S} \cup \{j\} \text{ and } \mathcal{N} = \mathcal{N} \setminus \{j\}.$$

Otherwise, compute the Lagrange multiplier estimate associated with the jth variable, namely,

$$[\sigma]_j = [g_k]_j + \sum_{i \in \beta_j} \left[-B^{-1}N\right]_{ij} [g_k]_i .$$

If $[x_k]_j$ is not potentially blocked (see above), and if either

$$[\sigma]_j < 0 \text{ and } [x_k]_j = [l]_j$$

or

$$[\sigma]_j > 0 \text{ and } [x_k]_j = [u]_j ,$$

then redefine $\mathcal{S}$ and $\mathcal{N}$ by

$$\mathcal{S} = \mathcal{S} \cup \{j\} \text{ and } \mathcal{N} = \mathcal{N} \setminus \{j\}.$$

Step 2. Compute the matrices B, S, N, and Z from the partition (2.5) and (2.8). If $\mathcal{S} = \emptyset$ or

$$\|Z^T g_k\|_\infty \leq \eta_3,$$

STOP (x_k is a local optimum within the required accuracy).

The constant η_3 whose choice controls the final accuracy requirement will be specified later.

2.5. The specific algorithm. We are now in position to specify our trust region algorithm for nonlinear network optimization in its entirety.

TRNNO Algorithm.

Step 0. The bounds l and u, the vector b and the network associated with the matrix A are given. Compute a feasible starting point x_0 (if not given) and an initial trust region radius Δ_0. Compute $f(x_0)$, g_0 and H_0. Find an initial maximal spanning tree of the network, defining a set of basic indices $\mathcal{B}$. Set $k = 0$.

Step 1. Given η_3, test the optimality of the current iterate x_k using the OT Algorithm of §2.4 and STOP if x_k is optimal.

Step 2. Calculate the bounds l_k and u_k from (2.3). Given μ_3, find a GCP x_k^C using the GCP Algorithm detailed in §2.2. (Also include an updating phase for the maximal spanning tree.)

Step 3. Compute the active set $\mathcal{A}(x_k^C, l_k, u_k)$ and apply the TCG Algorithm proposed in §2.3, using a truncated conjugate gradient scheme, to find an approximation $x_k + s_k$ to the minimizer of the trust region problem (2.4), with the additional restriction that the variables whose indices are in $\mathcal{A}(x_k^C, l_k, u_k)$ remain fixed at the corresponding values of x_k^C. (Also include an updating phase for the maximal spanning tree.)

Step 4. Compute $f(x_k + s_k)$ and

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}.$$

Step 5. If $\rho_k > \eta_1$, then set

$$x_{k+1} = x_k + s_k$$

and update g_{k+1} and H_{k+1} accordingly. Otherwise, set

$$x_{k+1} = x_k$$

and update the maximal spanning tree (with respect to the bounds l and u only). Set Δ_{k+1} according to (2.2), increment k by one and go to Step 1.

3. Numerical experiments. In this section, we analyse and compare the various versions of our algorithm and we also briefly interpret our results when varying the storage scheme, the conditioning, the dimension, and the nonlinearity of the problem, the final accuracy level (η_3), and the type of bounds imposed on the variables, as in [21] and [22]. We then consider our algorithm in comparison with the LSNNO[1] routine, developed by Toint and Tuyttens in [21], [22], that uses a linesearch approach rather than a trust region approach to solve problem (1.1). Although it might have been instructive to compare the present algorithm with another such as GENOS [1] and an interior point method like [3], the amount of additional work would have been prohibitive and we preferred to use a competitive algorithm for which we had direct access to both the authors and the software.

We have experimented on all the test problems of [21] for which the first and second derivatives were available. Indeed, though the framework presented here is well suited to large dimensional problems and can be used in conjunction with partitioned secant updating techniques on the general class of partially separable problems (see [14] and [15]), the purpose of this paper is to show the viability of the framework proposed and studied in [4], as well as its efficiency on large-scale nonlinear problems. Consequently, the results are presented for problems with easily computable first and second derivatives. For the same reason, we did not consider any preconditioning in our present implementation.

We have mainly tested problems obtained by varying the five parameters of the so-called model test problem $P(\ell, a, c, i, r)$ constructed by Toint and Tuyttens [21], where

ℓ defines the number of arcs $n = 2(2\ell + 1)(2\ell + 2)$ and the number of nodes $n_n = (2\ell + 2)^2$ of the problem;

a defines the nonlinearity of the function (for $a = 0$ the function is a simple quadratic);

c is an estimate of the condition number of the objective's Hessian matrix projected in the subspace of variables that satisfy the network constraints;

i and r determine a specific set of bounds on the flows (for $i = 0$ no bounds are imposed, for $i = 1$ a lower bound equal to r is imposed on the flows whose index is a multiple of three, for $i = -1$ some flows are fixed while others are bounded, principally those on the border of the grid with lower bound equal to r).

A brief description of this model test problem follows, the reader being refered to [21] for more details. The network is constructed as a square planar grid. An example with $\ell = 2$ is shown in Fig. 1. The supply/demand vector is

$$b_1 = +10, \quad b_j = 0 \ (j = 2, \ldots, n_n - 1), \quad b_{n_n} = -10.$$

[1] LSNNO is available from NETLIB.

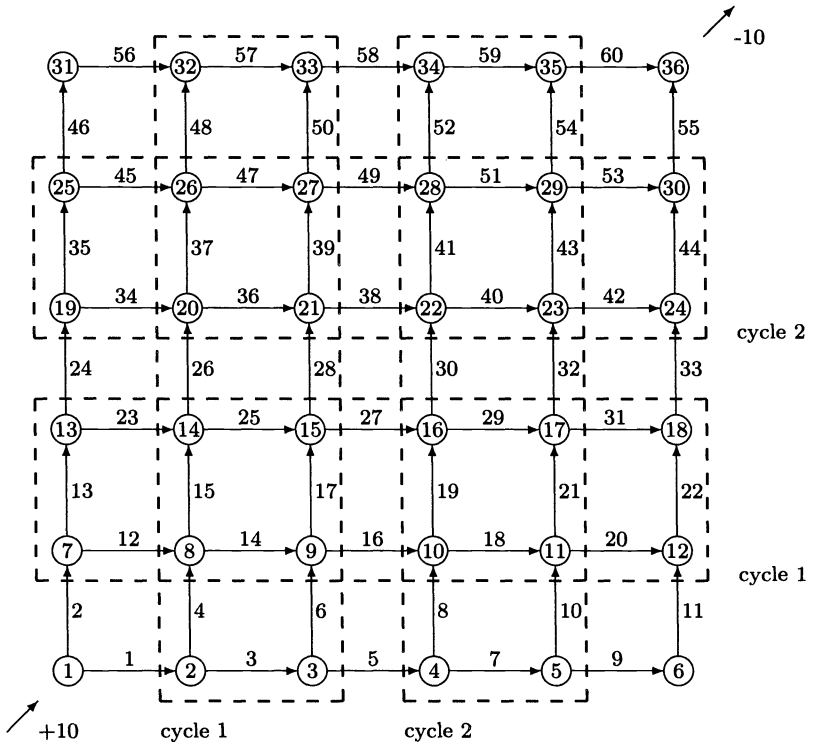


FIG. 1. *The network of the model test problem for* $\ell = 2$.

Furthermore, sets of ℓ horizontal cycles and ℓ vertical cycles are distinguished in the grid (see the dashed lines in the example of Fig. 1). We define, for $i = 1, \ldots, n$,

$$j(i) \stackrel{\text{def}}{=} \begin{cases} s & \text{if the } i\text{th arc belongs to cycle } s \text{ (horizontal or vertical),} \\ 0 & \text{if the } i\text{th arc does not belong to any cycle.} \end{cases}$$

The objective function is then given by

$$f(x) = \frac{1}{100}\sum_{i=1}^{n} \alpha_i [x]_i^2 + \frac{a}{100}\left(\sum_{i=1}^{n-1} \sqrt{1 + [x]_i^2 + ([x]_i - [x]_{i+1})^2}\right. \tag{3.1}$$
$$\left. + \frac{1}{1200}\left[10 + \sum_{i=1}^{n} (-1)^i [x]_i\right]^4\right),$$

where

$$\alpha_i \stackrel{\text{def}}{=} \begin{cases} 10^{\frac{j(i)-1}{\ell-1}\log_{10} c} & \text{if } j(i) \geq 1, \\ 1 & \text{if } j(i) = 0. \end{cases}$$

We have also tested the so-called Dembo's test problems given by Dembo in [11]. These problems are summarized in Table 1 (where n denotes the number of arcs and n_n denotes the number of nodes). All of them are *totally* separable, convex, and rather ill conditioned (the condition number of the reduced Hessian at the solution

TABLE 1
Dembo's test problems.

Name	n	n_n	Description
W30	46	30	Small Dallas water distribution model
W150	196	150	Medium Dallas water distribution model
W666	906	666	Large Dallas water distribution model
MB64	117	64	Small Thai matrix balancing problem
MB1116	2230	1116	Large Thai matrix balancing problem

varying between 10^4 and 10^8), and the number of bounds active at the solution is small, compared to n.

All the computations have been performed in double precision on a DEC VAX 3500, under VMS, using the standard Fortran Compiler ($\epsilon_M \simeq 1.39 \times 10^{-17}$).

The tests reported below all use the following values for the algorithm's constants (suggested in [4]):

$$\eta_1 = 0.25 \text{ and } \eta_2 = 0.75, \quad \mu_1 = 0.1 \text{ and } \mu_2 = 0.9, \quad \nu_1 = 10^{-5} \text{ and } \nu_2 = 0.01.$$

In order to allow the initial parameter t_0 in the GCP Algorithm to be more refined than Δ_k itself (since this last value represents a trust region radius for the quadratic model much more than for the linear model used in Step 1), we have selected the following value,

$$t_0 = \min\left[\left|\frac{g_k^T g_k}{g_k^T H_k g_k}\right| \|g_k\|_\infty, \Delta_k\right], \tag{3.2}$$

where the first quantity in brackets is the distance from x_k to the minimum of the quadratic model in the steepest descent direction, computed in the infinity norm. The value of μ_3 in the GCP Algorithm (that can be interpreted as the level of solution of the linear network problem (2.12)) is specified for each table of results given below. We have chosen the value 0.1 (rather than the classical value 0.5) for the scalar λ in the GCP Algorithm. This is intended to speed up the branching to the second case of Step 3 in this algorithm, therefore possibly reducing the number of times a candidate step s_i is computed in Step 1, since this last calculation is expected to be expensive compared with the rest of the algorithm. The final accuracy level η_3 in the OT Algorithm is specified for each model problem and is set to 10^{-2} for the Dembo's test problems, as recommended in [11]. In all cases, the (possibly infeasible) starting point is the origin. A feasible starting point x_0 as required in the statement of the TRNNO Algorithm is then computed via an "all artificial start Phase 1" (see [21]). This allows comparison with the LSNNO routine that starts with the same point. (Note that since the cpu time for the computation of this point, when required, is always negligible compared with the overall cpu time, it will be ignored in the cpu times given in the tables.) Finally, the initial trust region radius is fixed to the following value in our tests:

$$\Delta_0 = \min[\|g_0\|_2, 100]. \tag{3.3}$$

3.1. Comparison between the different versions. In this section we comment on the five Dembo's test problems and on twenty others selected from particular choices in [21], [22] of the model test problem's parameters. Table 2 reports the characteristics of these twenty test problems which are divided into six subsets, according

TABLE 2
The model test problems.

	Name	ℓ	a	c	i	r	η_3	Storage
Storage scheme	MP1	8	1	10^3	-1	$\frac{1}{10}$	10^{-5}	I
	MP2	8	1	10^3	-1	$\frac{1}{10}$	10^{-5}	E
	MP3	8	1	10^3	-1	$\frac{1}{10}$	10^{-5}	O
Conditioning	MP4	8	1	1	1	$\frac{1}{10}$	10^{-5}	I
	MP5	8	1	10	1	$\frac{1}{10}$	10^{-5}	I
	MP6	8	1	10^2	1	$\frac{1}{10}$	10^{-5}	I
	MP7	8	1	10^3	1	$\frac{1}{10}$	10^{-5}	I
	MP8	8	1	10^4	1	$\frac{1}{10}$	10^{-5}	I
	MP9	8	1	10^5	1	$\frac{1}{10}$	10^{-5}	I
Dimension	MP10	4	1	10^2	1	$\frac{1}{10}$	10^{-5}	I
	MP11	8	1	10^2	1	$\frac{1}{10}$	10^{-5}	I
	MP12	12	1	10^2	1	$\frac{1}{10}$	10^{-5}	I
Nonlinearity	MP13	8	0	10^2	1	$\frac{1}{10}$	10^{-5}	I
	MP14	12	0	10^2	-1	$\frac{1}{10}$	10^{-5}	I
Final accuracy	MP15	8	1	10^2	1	$\frac{1}{10}$	10^{-3}	I
	MP16	8	1	10^2	1	$\frac{1}{10}$	10^{-5}	I
	MP17	8	1	10^2	1	$\frac{1}{10}$	10^{-7}	I
Type of bounds	MP18	8	1	10^2	0	0	10^{-5}	I
	MP19	8	1	10^2	1	$\frac{1}{10}$	10^{-5}	I
	MP20	8	1	10^2	1	$\frac{1}{5}$	10^{-5}	I

to the different features tested in [21], [22] and mentioned above. Note that the symbols I, E, and O in the last column of this table are used to denote storage using internal dimensions (I), elemental dimensions (E), or using one element (O) for the Hessian matrix (see [21], [22] for more details). When required for comparisons in the next section, we report the relevant numerical results of all the twenty tests, even if some of these are identical (namely, MP6, MP11, MP16, and MP19).

We introduce the notation used in the tables presenting the results.

it: the number of major iterations (in the TRNNO Algorithm of §2.5);

gcp: the total number of iterations in the GCP calculations;

avn: the average number of GCP calculations per major iteration;

cg: the total number of conjugate gradient recurrences;

nf: the number of function evaluations (i.e., the number of element function evaluations divided by the number of elements);

ng: the number of gradient evaluations (i.e., the number of element gradient evaluations divided by the number of elements);

nH: the number of Hessian evaluations (i.e., the number of element Hessian evaluations divided by the number of elements);

np: the number of maximal spanning tree updates where a pivoting step occurs;

gcpcpu: the cpu time in seconds for the GCP calculations;

cgcpu: the cpu time in seconds for the conjugate gradient recurrences;

totcpu: the total cpu time in seconds (Phase 1 excluded).

Note that fractional numbers of function, gradient, or Hessian evaluations are expected, since the partial separability of the objective allows skipping the reevaluation of the elements whose variables have not been modified since the last evaluation. On the other hand, for the sake of clarity, we round off the cpu times to the nearest integer number.

We first turn our attention to the computation of a candidate step s_i at Step 1 of the GCP Algorithm. As already mentioned in §2.2, we have implemented a *total*

TABLE 3
Total pricing, $\mu_3 = 1$, 0.9, *and* 0.6.

Problem	μ_3	it	gcp	avn	cg	gcpcpu	cgcpu	totcpu
MP4	1	11	31	2.8	587	118	110	241
	0.9	11	31	2.8	564	103	106	223
	0.6	11	31	2.8	530	78	99	190
MP8	1	7	30	4.3	2433	60	441	510
	0.9	9	43	4.8	3315	63	607	681
	0.6	11	47	4.3	3475	31	619	663
MP12	1	7	28	4.0	1078	333	479	833
	0.9	8	34	4.2	1254	280	554	857
	0.6	9	31	3.4	1125	88	489	602
MP14	1	5	16	3.2	291	30	29	68
	0.9	6	24	4.0	357	20	35	66
	0.6	13	39	3.0	603	12	58	88
MP16	1	8	30	3.7	815	83	152	244
	0.9	8	29	3.6	871	70	164	243
	0.6	8	27	3.4	726	49	134	193
MP18	1	12	45	3.7	823	113	166	295
	0.9	12	45	3.7	810	109	164	288
	0.6	12	45	3.7	827	78	166	260
W150	1	15	61	4.1	362	6	5	16
	0.9	15	61	4.1	405	5	6	14
	0.6	18	73	4.1	442	3	6	14
W666	1	19	97	5.1	1314	247	110	385
	0.9	21	108	5.1	1890	146	159	334
	0.6	26	130	5.0	2076	64	162	260
MB1116	1	39	60	1.6	15039	2834	4632	7541
	0.9	40	57	1.4	13806	1886	3643	5599
	0.6	40	57	1.4	14587	1481	4739	6302

pricing routine that *approximately* solves problem (2.12). We have tested this routine for different values of μ_3. The results are presented in Table 3 for a representative sample of the twenty-five problems.

We first observe that the number of major iterations usually increases when the value of μ_3 decreases (especially for $\mu_3 = 0.6$). The reason is that for smaller and smaller values of μ_3, the GCP is allowed to be chosen further and further from the projected gradient path. This exhibits the importance of the part played by the GCP in our class of trust region methods and the need of computing a sufficiently good approximation of this point on the projected gradient path. The total number of GCP iterations increases accordingly. However, we observe that the average number of GCP calculations per major iteration decreases with the value of μ_3, while the cpu times for the GCP calculations considerably decrease, particularly for larger problems (such as MP12, MP14, W666, and MB1116). This is due to the fact that the solution of the linear network problem (2.12) may be stopped prematurely when finding an approximate solution. Nevertheless, comparing the total cpu times, we conclude that it is worthwhile solving (2.12) approximately whenever the GCP found does not depart too much from the projected gradient path and the total number of iterations is largely unaffected (see MP4, MP16, and MP18). This means that the value of μ_3 must be reduced with care.

We have also tested the *partial pricing* routine that *completely* solves problem (2.12) (hence setting $\mu_3 = 1$). These results are reported in Table 4. The total cpu times are better than those given in Table 3. This is due to much better cpu times

TABLE 4
Partial pricing, $\mu_3 = 1$.

Problem	it	nf	ng	nH	np	gcp	cg	gcpcpu	cgcpu	totcpu
MP1	6	5.9	5.9	5.0	0	27	393	8	63	79
MP2	6	5.9	5.9	5.0	0	27	402	53	628	709
MP3	6	7.0	7.0	6.0	0	27	400	50	571	649
MP4	11	11.3	11.3	10.4	6	31	583	22	111	147
MP5	9	9.5	9.5	8.5	4	29	683	18	133	163
MP6	8	8.5	8.5	7.5	5	30	816	17	154	182
MP7	8	8.5	8.5	7.5	5	36	1610	19	306	334
MP8	7	7.5	7.5	6.6	5	31	2450	15	453	477
MP9	10	10.3	10.3	9.3	3	50	6649	24	1260	1297
MP10	5	5.6	5.6	4.7	3	17	190	2	10	14
MP11	8	8.5	8.5	7.5	5	30	816	17	154	182
MP12	7	7.5	7.5	6.6	54	28	1116	39	492	551
MP13	5	5.4	5.4	4.5	7	19	521	7	37	49
MP14	5	3.7	3.7	3.1	3	16	295	6	29	44
MP15	7	7.5	7.5	6.5	5	23	713	14	134	157
MP16	8	8.5	8.5	7.5	5	30	816	17	155	182
MP17	9	9.5	9.5	8.5	5	39	1024	21	196	227
MP18	12	13.0	13.0	12.0	0	45	809	24	166	206
MP19	8	8.5	8.5	7.5	5	30	816	17	155	182
MP20	7	7.1	7.1	6.2	7	30	631	12	112	133
W30	15	13.9	13.9	13.0	1	72	113	1	1	2
W150	15	12.4	12.4	11.6	4	61	351	4	5	13
W666	16	15.1	15.1	14.2	6	79	1087	37	91	151
MB64	55	50.6	50.6	49.6	42	56	2551	8	30	43
MB1116	36	23.4	23.4	22.6	142	55	12823	255	4064	4391

TABLE 5
The effect of varying the initial trust region radius for MB1116.

Δ_0	it	gcp	cg	gcpcpu	cgcpu	totcpu
10^5	35	55	10166	243	3033	3343
10^9	34	56	10805	241	3373	3678
10^{14}	34	57	10645	240	3238	3542
10^{18}	34	57	10645	239	3244	3547

for the GCP calculations. Indeed, problems MP4 and MP18, for instance, present similar numbers of GCP calculations and yet, the exact solution's calculation using partial pricing is less expensive than the approximate solution's calculation, even when $\mu_3 = 0.6$. This can be explained by the additional amount of work required for maintaining the upper bound on the value of $\mu_3 \alpha_k(t_i)$ in (2.18) when approximately solving (2.12). This additional work is not sufficiently balanced by the use of the upper bound and leads to the conclusion that it is not worth solving approximately the linear problem (2.12) in the GCP calculation, at least in the presence of *network* constraints, since a fast solver can then be implemented to solve problem (2.12) exactly. We therefore abandon, from now on, the approximate solution of (2.12) in favour of the exact one using partial pricing.

We now further analyse the results reported in Table 4 for the twenty-five test problems. The number of iterations used in the GCP calculations are generally quite reasonable when compared with the number of major iterations or with the total number of conjugate gradient recurrences. The same conclusion applies when comparing the respective cpu times. This is partly due to the choice of a small value for λ in the GCP Algorithm. Indeed, we have tested the same code with $\lambda = 0.1$ replaced

by $\lambda = 0.5$, and we have clearly detected a substantial increase in the number of iterations and the cpu times for the GCP calculations. This thus justifies a choice for λ that allows a rapid branching to the RS Algorithm (Case 2 of Step 3 in the GCP Algorithm), therefore avoiding an unnecessarily high number of solutions of the linear network problem (2.12). Moreover, we also observe that the amount of work in the GCP calculations grows more slowly with the size of the problem than in the conjugate gradient scheme (compare MP10, MP11, and MP12, or the Dembo's problems, for instance). This is true for the number of iterations as well as for the cpu times.

We have also tested the impact of the choice of the initial trust region radius Δ_0 on the performances of the method. Indeed, the initial value given in (3.3) is rather heuristic, and we actually observed that eighteen of the twenty-five test problems selected $\Delta_0 = 100$. Table 5 reports the results obtained when solving problem MB1116 for different initial trust region radii, with the GCP Algorithm with $\mu_3 = 1$ (economical version). These results, compared with those of Table 4, show a possible saving of up to 25% in the total cpu times, depending only on the value of Δ_0. This saving occurs essentially in the conjugate gradient iteration counts. This emphasizes the importance of a good choice for this last value.

3.2. Variation of the test problems' features. We briefly interpret here our results on the twenty model test problems of §3.1 when varying the six items mentioned at the beginning of §3. The reader is invited to consult Table 4 to confirm the comments given below.

We first observe essentially identical behaviour for the method of this manuscript and that of [22] when using the three different storage schemes for the Hessian matrix (see MP1–MP3). Our cpu times are clearly in favour of the internal storage technique, although, for example, the additional subroutine calls necessary in this context can be quite significant. We also observe a small increase in function, gradient, and Hessian counts when going from the elemental dimension storage to that of one element, the number of conjugate gradient steps and the iteration counts being approximately unchanged. This effect is due to the loss of the partially separable character of the objective in the latter case, which prevents partial evaluations of the function or of its derivatives. The gains in cpu time for the storage using one element as opposed to the elemental dimension storage is caused by the fact that the products involving the Hessian matrices are cheaper to compute (see [21]).

We also see, as in [21], [22], that the method is sensitive to variations of conditioning (see MP4–MP9). This is due to the use of the conjugate gradient method, which is a conditioning sensitive method and leads to an increase in the number of conjugate gradient recurrences (while the GCP calculations remain comparable).

The problem becomes slightly more difficult when its size increases, mostly because of the added complexity of the bound constraints (see for example MP10–MP12). Nevertheless, the difficulty seems to increase faster in [21] than for our code. This will be confirmed in the next section.

Moreover, when the objective function is quadratic (i.e., when $a = 0$ in Table 2 or in (3.1)), we can say, unlike [21], that the problem is easier in terms of major iterations, function, gradient, and Hessian evaluations, as well as in terms of conjugate gradient steps and cpu times (see MP13, to compare with MP6, and MP14).

As observed in [21], [22], a tighter requested accuracy on the solution does not cause a large increase in the number of major iterations (see MP15–MP17). This is explained by the rapid rate of convergence achieved by both methods. The number of conjugate gradient recurrences may however be significantly increased by a tighter

TABLE 6
Number of arcs and nodes for a given ℓ.

ℓ	11	12	13	14	15	16	19	22
n	1104	1300	1512	1740	1984	2244	3120	4140
n_n	576	676	784	900	1024	1156	1600	2116

TABLE 7
Comparison with LSNNO *for Case* $i = 1$.

r	ℓ	it		%act
		TRNNO	LSNNO	
0.15	16	13	19	9.6
	19	9	28	11.4
	22	17	34	13.4
0.35	16	15	18	23.3
	19	9	32	24.6
	22	10	66	25.1
0.55	16	9	25	26.2
	19	7	41	26.6
	22	7	42	27.0
0.75	16	5	24	27.4
	19	8	63	27.6
	22	7	41	28.0

accuracy requirement, because a large part of this computational effort occurs in the last iterations of the algorithm, where the linear system (2.23) must be solved accurately.

Finally, unlike [21], [22], we note that the introduction of bounds does not increase the number of major iterations, but even decreases it (see MP18–MP20). This is discussed in the next section. On the other hand, as in [21], [22], the number of pivoting steps increases with the tightness of the bounds. This is due to the fact that the basic variables are increasingly constrained.

3.3. Comparison with the LSNNO routine. In this section, we compare the TRNNO routine (partial pricing, Δ_0 given by (3.3)) with the LSNNO code of Tuyttens, both tested on the same machine.

We first consider the results of Table 4 in comparison with those produced by LSNNO in [21], [22] when using Newton's method without preconditioning. We observe, in most cases, a decrease in the number of major iterations for TRNNO (especially for problems MP9 and MP12). This implies fewer function, gradient, and Hessian evaluations. On the other hand, the number of conjugate gradient recurrences generally increases, mainly because the TCG Algorithm allows the restarting of the conjugate gradient scheme. So we may conclude that the TRNNO code requires fewer iterations than the LSNNO code, but that one iteration is more expensive for TRNNO, due to the GCP calculations and the restarting steps in the conjugate gradient iterations. For this first set of problems (whose characteristics are summarized in Table 2), we may observe that LSNNO generally outperforms TRNNO in cpu times, except, in particular, for the large model test problems (MP12 and MP14) and when the bounds on the variables become tighter (MP18–MP20), that is, when the number of bounds potentially active at the solution increases. In order to investigate this issue further, we have extended our original set of problems and tested both codes on the model test problem for different sizes and types of bounds, with the fixed parameters $a = 1$, $c = 10^2$, $\eta_3 = 10^{-5}$ for the final accuracy, and using a storage with internal

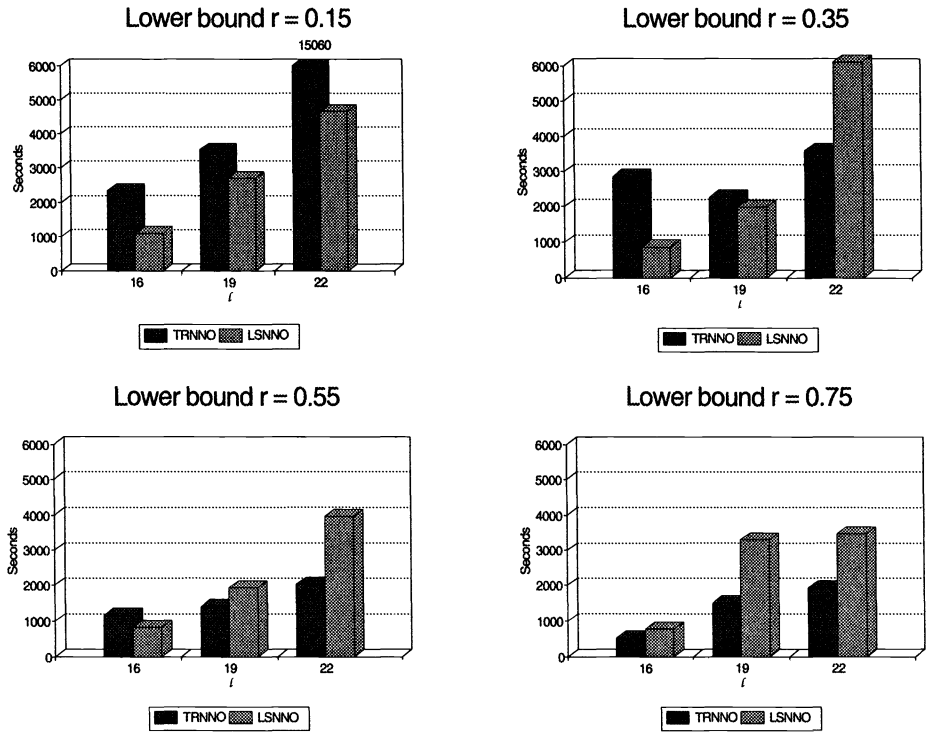


FIG. 2. *Comparison with* LSNNO *on cpu times for fixed bounds* (*Case* $i = 1$).

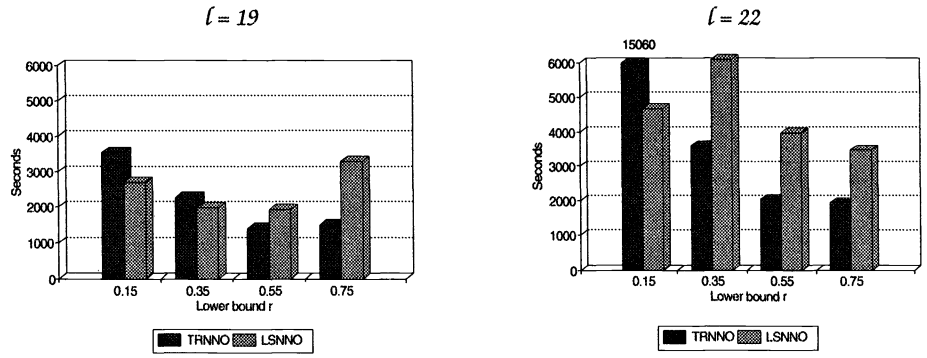


FIG. 3. *Comparison with* LSNNO *on cpu times for fixed sizes* (*Case* $i = 1$).

dimension (I). This last choice is indeed the most common choice made in the original set of problems (see Table 2). We report the results in Tables 7–10 and in Figs. 2–8. The various sizes specified by the parameter ℓ are given in Table 6. We have selected three types of bounds.

Case $i = 1$. We impose that $r \leq [x]_j \leq \infty$ for all index j such that mod(j,3) = 0, all other variables being unconstrained. r is successively equal to 0.15, 0.35, 0.55, and 0.75.

Case $i = 2$. We impose that $r \leq [x]_j \leq \infty$ for all index j such that mod(j,3) = 1,

TABLE 8
Comparison with LSNNO *for Case* $i = 2$.

r	ℓ	it		%act
		TRNNO	LSNNO	
0.0	16	16	11	0.6
	19	12	19	0.7
	22	21	29	0.7
0.5	16	8	29	26.1
	19	6	28	26.8
	22	8	39	27.2
1.0	16	10	41	28.9
	19	7	53	28.9
	22	7	63	29.3
2.5	16	8	50	30.9
	19	9	42	31.1
	22	10	103	31.3

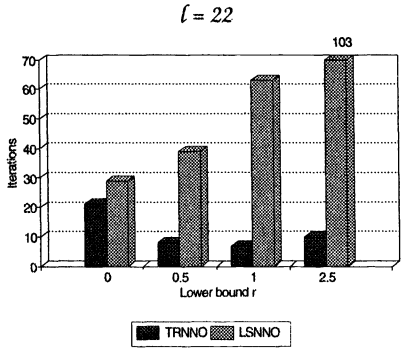


FIG. 4. *Comparison with* LSNNO *on iterations for fixed size* (*Case* $i = 2$).

TABLE 9
Comparison with LSNNO *for Case* $i = 3$.

r	ℓ	it		%act
		TRNNO	LSNNO	
0.0	11	13	54	1.1
	12	13	58	1.1
	13	12	68	1.1
	14	13	53	1.1
	15	13	150	3.4
	16	14	76	1.3
0.1	11	10	54	13.1
	12	12	52	14.0
	13	9	48	14.7
	14	12	101	14.7
	15	9	85	17.9
	16	11	132	17.1
0.2	11	9	83	28.6
	12	23	142	30.8
	13	9	159	35.2
	14	11	237	40.1
	15	9		
	16	8		

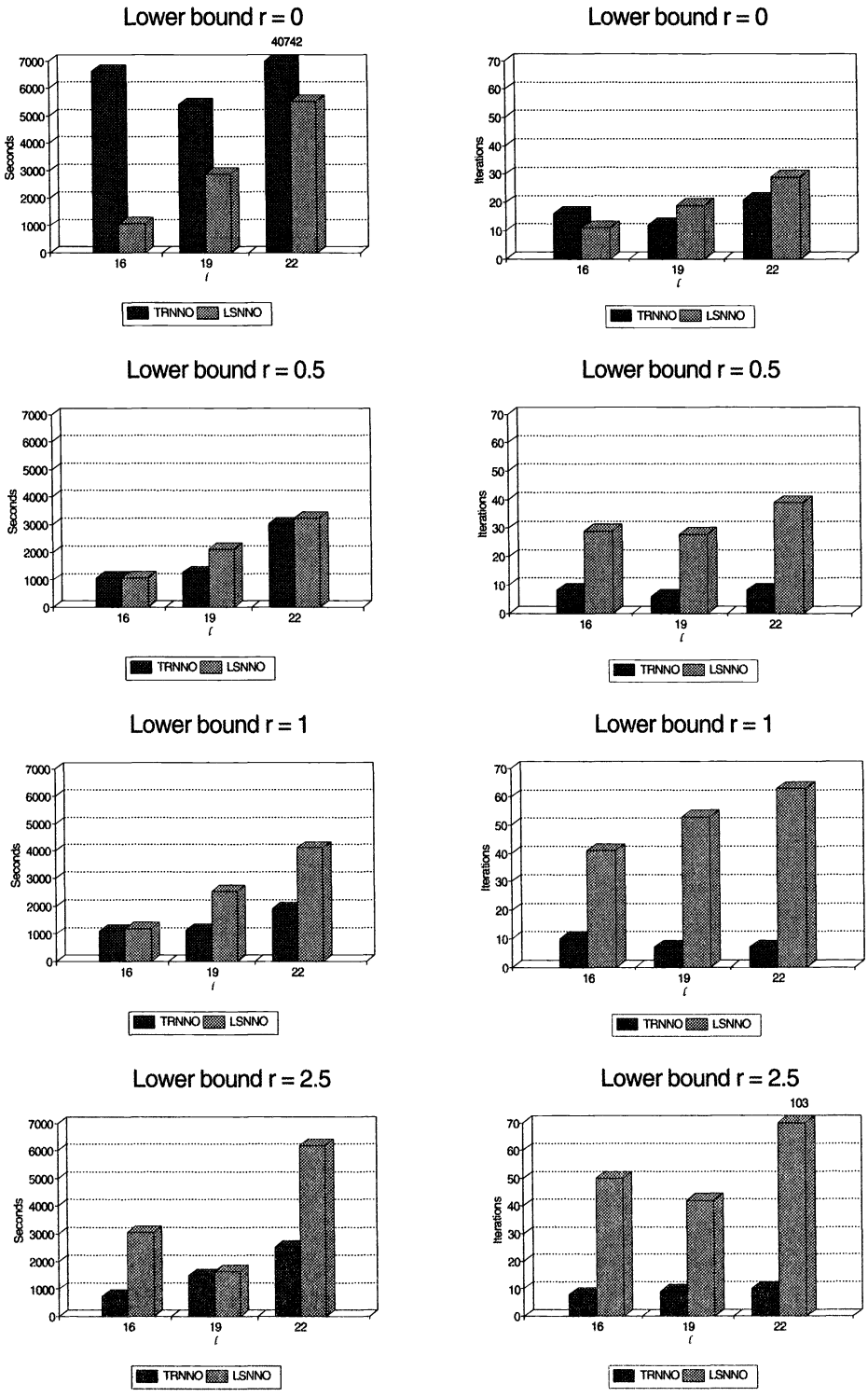


FIG. 5. *Comparison with* LSNNO *on cpu times and iterations for fixed bounds* (*Case* $i = 2$).

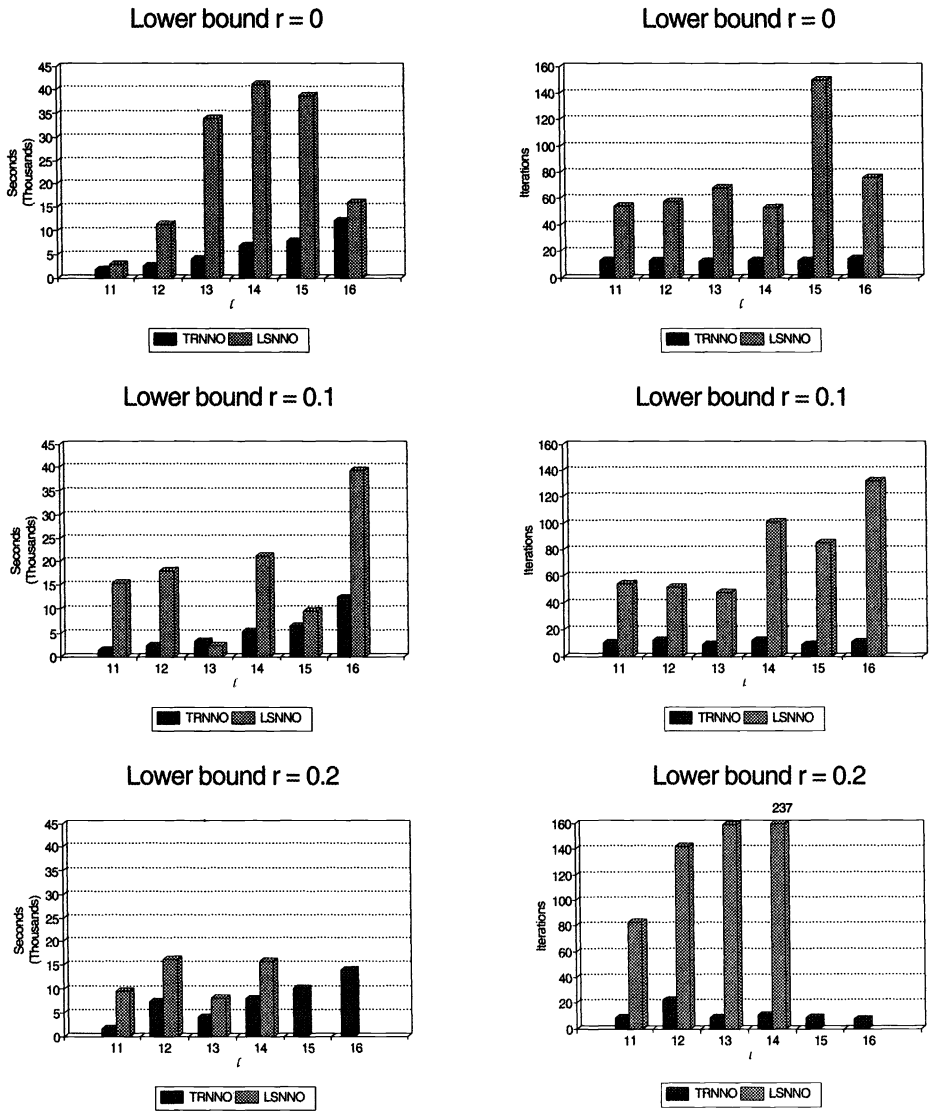


FIG. 6. *Comparison with* LSNNO *on cpu times and iterations for fixed bounds* (*Case* $i = 3$).

all other variables being unconstrained. r is successively equal to 0, 0.5, 1, and 2.5.

Case $i = 3$. We impose that $r \leq [x]_j \leq \infty$ for all index j whose corresponding arc is on horizontal lines or alternate vertical lines (beginning at the first) of the grid, all other variables being unconstrained. r is successively equal to 0, 0.1, and 0.2.

For the two first cases, one-third of the variables are constrained while this ratio increases to three-quarters for Case $i = 3$.

First, we comment on the results given in Table 7 and Figs. 2 and 3 for Case $i = 1$. In Table 7 and the following ones, "%act" denotes the percentage of active bounds at the solution (computed by LSNNO).

The results of Table 7 show that on the whole the number of major iterations

TABLE 10
Comparison with LSNNO *for Case* $i = 3$ *(higher dimensions).*

r	ℓ	it		%act
		TRNNO	LSNNO	
0.0	19	12	142	1.1
0.1	22	10	495	26.0

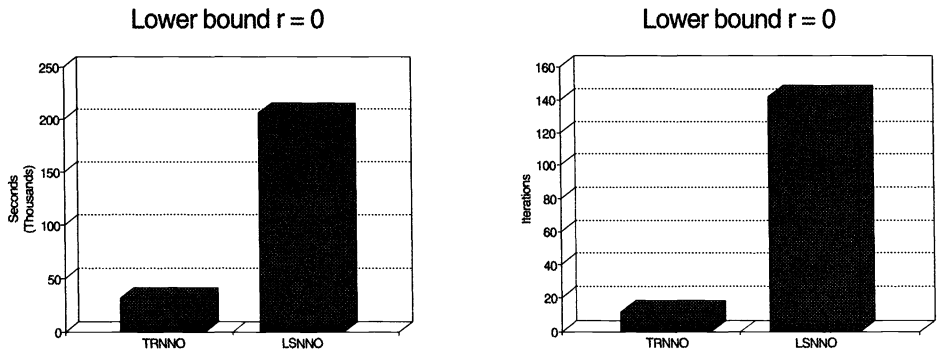


FIG. 7. *Comparison with* LSNNO *for* $l = 19$ *(Case* $i = 3$*).*

decreases when bounds become tighter for TRNNO, as mentioned in the previous section. On the other hand, these numbers increase for LSNNO. Now comparing the cpu times given in Fig. 2, we may observe that this behaviour has the effect of improving the performances of TRNNO while those of LSNNO deteriorate. Moreover, we observe that TRNNO outperforms LSNNO for the largest problem first, then for the medium one and finally for the smallest one. Figure 3 once again confirms the above observation. For tighter and tighter bounds, TRNNO produces better and better cpu times (except for $l = 19$ when going from $r = 0.55$ to $r = 0.75$), while those for LSNNO behave erratically but are consistently worse. Finally, from Fig. 2 and the last column of Table 7, we can see that the cpu times are overwhelmingly in favour of TRNNO when about a quarter of the bounds are active at optimality.

The second case is reported in Table 8 and Figs. 4 and 5. These results corroborate the conclusions made for the previous case. It further shows (see Fig. 5) how constant the number of iterations for TRNNO remains when the bounds and the size vary, while these numbers grow for LSNNO. Figure 4 displays this characteristic for $l = 22$.

Finally, Table 9 and Fig. 6 show the results for the third case of bounds. The absence of results for LSNNO means that it stopped with a flag error before having solved the problem. The results also confirm the above comments, except that this time TRNNO outperforms LSNNO immediately, even when about 1% of the bounds are active at optimality. In particular, Fig. 6 clearly shows the uniform behaviour of TRNNO. Indeed, for the three different bounds, the iterations numbers stay alike while the cpu times grow slowly with the dimension of the problem. The tightness of the bounds does not seem to affect the performances of the code. On the other hand, it is not possible to attribute the same stability to LSNNO. Moreover, although optimal function values usually agree for both codes in Cases $i = 1$ and 2, we have observed here a significant difference, always in the favor of TRNNO, for three-quarters of the test problems. We also observed that the strict complementarity slackness condition

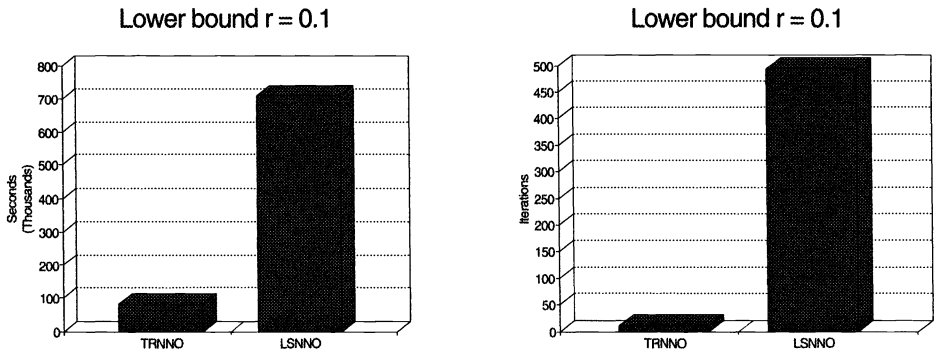


FIG. 8. *Comparison with* LSNNO *for* $l = 22$ (*Case* $i = 3$).

did usually not hold at the solution for these problems.

Furthermore, for Case $i = 3$, Table 10 and Figs. 7 and 8 report the results for higher dimensions. They confirm the efficiency of TRNNO on large-scale problems. We also tested other cases of bounds that are not reported in this paper. They all corroborate the conclusions made in this section.

4. Conclusions and perspectives. In this paper, we propose a new algorithm of trust region type to solve the nonlinear network problem (1.1). We consider practical implementation issues, including an explicit procedure for computing an approximate GCP and a truncated conjugate gradient strategy for calculating a candidate step at each iteration. Numerical tests are reported and discussed, showing the efficiency of the trust region approach, especially for large-scale problems with potentially many active bound constraints at the solution. We believe that part of the success may be attributed to the ability of the GCP calculation to swiftly determine the set of (nondegenerate) active bounds at the solution.

The encouraging results show that the framework presented is worth considering for the solution of problem (1.1), especially in view of the good theoretical properties of the framework given in [4] and the numerical results for large problems. It also suggests some directions for future research and continued development. The method given here could be adapted for solving *general large-scale linearly constrained problems*. We could then envisage to produce effective methods for solving *general large-scale nonlinear programming problems* by combining the nonlinear constraints in a suitable fashion with the objective function (for instance in an augmented Lagrangian function [8], [5], [6]), and solving the resulting sequence of linearly constrained problems using the method described in this paper.

Acknowledgments. The author wishes to thank Michel Bierlaire, Andy Conn, Nick Gould, Philippe Toint, and Daniel Tuyttens for their useful comments and suggestions. Daniel Tuyttens also kindly gave access to his code, his test problems, and his numerical results.

REFERENCES

[1] D. P. AHLFELD, R. S. DEMBO, J. M. MULVEY, AND S. A. ZENIOS, *Nonlinear programming on generalized networks*, ACM Trans. Math. Software, 13 (1987), pp. 350–367.

[2] G. H. BRADLEY, G. G. BROWN, AND G. W. GRAVES, *Design and implementation of large-scale transshipment algorithms*, Management Sci., 24 (1977), pp. 1–34.

[3] T. M. CARPENTER, I. J. LUSTIG, J. M. MULVEY, AND D. F. SHANNO, *Higher-order predictor-corrector interior point methods with application to quadratic objectives*, SIAM J. Optim., 3 (1993), pp. 696–733.

[4] A. R. CONN, N. GOULD, A. SARTENAER, AND P. L. TOINT, *Global convergence of a class of trust region algorithms for optimization using inexact projections on convex constraints*, SIAM J. Optim., 3 (1993), pp. 164–221.

[5] ———, *Global convergence of two augmented Lagrangian algorithms for optimization with a combination of general equality and linear constraints*, Tech. Report 93/01, Dept. of Mathematics, FUNDP, Namur, Belgium, 1993.

[6] ———, *Local convergence properties of two augmented Lagrangian algorithms for optimization with a combination of general equality and linear constraints*, Tech. Report 93/20, Dept. of Mathematics, FUNDP, Namur, Belgium, 1993.

[7] A. R. CONN, N. I. M. GOULD, AND P. L. TOINT, *Global convergence of a class of trust region algorithms for optimization with simple bounds*, SIAM J. Numer. Anal., 25 (1988), pp. 433–460. Correction, same journal, 26 (1989), pp. 764–767.

[8] ———, *A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds*, SIAM J. Numer. Anal., 28 (1991), pp. 545–572.

[9] G. B. DANTZIG, *Linear Programming and Extensions*, Princeton University Press, Princeton, NJ, 1963.

[10] R. S. DEMBO, *The performance of NLPNET, a large scale nonlinear network optimizer*, Math. Programming, Series B, 26 (1986), pp. 245–249.

[11] ———, *A primal truncated Newton algorithm with application to large scale nonlinear network optimization*, Math. Programming, Series B, 31 (1987), pp. 43–71.

[12] R. S. DEMBO AND J. G. KLINCEWICZ, *A scaled reduced gradient algorithm for network flow problems with convex separable costs*, Math. Programming, 15 (1981), pp. 125–147.

[13] L. F. ESCUDERO, *A motivation for using the truncated Newton approach in a very large scale nonlinear network problem*, Math. Programming Stud., 26 (1986), pp. 240–245.

[14] A. GRIEWANK AND P. L. TOINT, *On the unconstrained optimization of partially separable functions*, in Nonlinear Optimization 1981, M. J. D. Powell, ed., London, New York, 1982, Academic Press, pp. 301–312.

[15] ———, *Partitioned variable metric updates for large structured optimization problems*, Numer. Math., 39 (1982), pp. 119–137.

[16] M. D. GRIGORIADIS, *An efficient implementation of the network simplex method*, Math. Programming, 26 (1986), pp. 83–111.

[17] J. L. KENNINGTON AND R. V. HELGASON, *Algorithms for Network Programming*, John Wiley, New York, 1980.

[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. Grötschel, and B. Korte, eds., Springer Verlag, Berlin, 1983, pp. 258–287.

[19] B. A. MURTAGH AND M. A. SAUNDERS, *Large-scale linearly constrained optimization*, Math. Programming, 14 (1978), pp. 41–72.

[20] Y. SHEFFI, *Urban Transportation Networks*, Prentice-Hall, Englewood Cliffs, NJ, 1985.

[21] P. L. TOINT AND D. TUYTTENS, *On large scale nonlinear network optimization*, Math. Programming, Ser. B, 48 (1990), pp. 125–159.

[22] ———, LSNNO: *a Fortran subroutine for solving large scale nonlinear network optimization problems*, ACM Trans. Math. Software, 18 (1992), pp. 308–328.

[23] D. TUYTTENS AND J. TEGHEM, *Théorie des matroides et optimisation combinatoire*, Belgian J. Oper. Res., Statist. Computer Sci., 26 (1986), pp. 27–62.

SIAM J. OPTIMIZATION
Vol. 5, No. 2, pp. 408–420, May 1995

009

LADDERS FOR TRAVELLING SALESMEN *

SYLVIA C. BOYD†, WILLIAM H. CUNNINGHAM‡, MAURICE QUEYRANNE§, AND YAOGUANG WANG‡

Abstract. We introduce a new class of valid inequalities for the symmetric travelling salesman polytope. The family is not of the common handle-tooth variety. We show that these inequalities are all facet-inducing and have Chvátal rank 2.

Key words. polyhedra, facets, travelling salesman problem

AMS subject classification. Primary: 90C08

1. Introduction.

The *symmetric travelling salesman polytope* $STSP(V)$ is the convex hull of incidence vectors of edge-sets of Hamiltonian cycles of the complete graph on node set V. A description of this polytope by linear inequalities would essentially reduce the travelling salesman problem to a linear program. While there are reasons to believe that we cannot hope to obtain such a complete description, known partial descriptions of the polytope have proved to be remarkably useful in cutting plane approaches to the problem. (See [4], [9], for example.) A good deal of progress has been made in extending these partial descriptions by finding new classes of facet-inducing inequalities and in incorporating this additional knowledge into the computational approaches.

In this paper we introduce a new class of valid inequalities for $STSP(V)$ called *ladder inequalities.* These inequalities differ from most of the inequalities discovered so far in that they are not of the usual handle-tooth variety. On the other hand, they arise from a strengthening of certain inequalities of this type. A computational study in [4] demonstrates the use of ladder inequalities to improve the bounds of linear programming (LP) relaxations. We prove that all ladder inequalities are facet-inducing. We also show that they all have Chvátal rank exactly 2.

2. Preliminaries.

Let V be any node set with $n \equiv |V| \geq 3$. We deal with the undirected complete graph $K_n = (V, E)$, and we write elements of E as (i, j) or ij. Note that $ij = ji$. For $S \subseteq V$, let $E(S)$ denote $\{ij \in E : i, j \in S\}$. For $S, T \subseteq V$ with $S \cap T = \emptyset$, let $E(S : T)$ denote $\{ij \in E : i \in S, j \in T\}$. For any $v \in V$, define $\delta(v)$ to be $E(\{v\} : V \setminus \{v\})$. For $B \subseteq E$ and $x \in \boldsymbol{R}^E$, let $x(B)$ denote $\sum(x_{ij} : ij \in B)$. Given $c \in \boldsymbol{R}^E$, the (*symmetric*) *travelling salesman problem* (TSP) can be stated as

$$\text{(1)} \qquad \text{minimize } \sum (c_{ij} x_{ij} : ij \in E)$$
$$\text{subject to}$$

$$\text{(1a)} \qquad \sum (x_{ij} : 1 \leq j \leq n,\ j \neq i) = 2, \quad i \in V;$$

* Received by the editors September 9, 1992; accepted for publication (in revised form) December 7, 1993. This research was partially supported by grants from the Natural Sciences and Engineering Research Council of Canada.

† Department of Computer Science, University of Ottawa, Ottawa, Ontario, Canada K1N 6N5.

‡ Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1 (whcunnin@math.uwaterloo.ca).

§ Faculty of Commerce and Business Administration, University of British Columbia, Vancouver, British Columbia, Canada V6T 1Z2.

(1b) $$x(E(S)) \leq |S| - 1, \quad S \subset V, \ 2 \leq |S| \leq n-2;$$

(1c) $$x_{ij} \geq 0, \quad ij \in E;$$

(1d) $$x_{ij} \text{ integer}, \quad ij \in E.$$

Any feasible solution x^0 of (1) is the incidence vector of (the edge-set of) a Hamiltonian circuit or *tour* of K_n. We identify a tour (or more generally a path) of K_n with its edge-set or its node-sequence. The convex hull of feasible solutions to (1) is called an *STS polytope* and is denoted $STSP(V)$. The symmetric TSP is equivalent to the linear program

$$\min\left(\sum (c_{ij}x_{ij} : ij \in E) : x \in STSP(V)\right),$$

and in order to apply the methods of linear programming, we would like to describe it as an optimization subject to linear constraints. It is known ([6], for example) that the affine hull of $STSP(V)$ is just the set of solutions of the *degree constraints* (1a), and hence its dimension is $\binom{n}{2} - n$. Therefore, an inequality $ax \leq a_0$ that is valid for $STSP(V)$ is *facet-inducing* if and only if $\{x \in STSP(V) : ax = a_0\}$ has dimension $\binom{n}{2} - n - 1$. Moreover, two such inequalities $ax \leq a_0$ and $bx \leq b_0$ are *equivalent* (that is, induce the same face) if and only if there exist $\lambda \in \boldsymbol{R}^V$ and a positive scalar λ_0 such that $(b, b_0) = \lambda(A, \bar{2}) + \lambda_0(a, a_0)$, where A is the node-edge incidence matrix of K_n, and $\bar{2}$ is a vector of 2's. One such class of inequalities consists of the *nonnegativity* constraints (1c). Another consists of the *subtour elimination* (SE) constraints (1b).

Many of the known classes of valid inequalities arose from generalizations of the comb inequalities, which we now describe. They were first defined by Chvátal [3] and later generalized by Grötschel and Padberg [5]. Given a *handle* $H \subset V$ and mutually disjoint *teeth* $T_1, T_2, \ldots, T_{2k+1} \subset V$ (k integer, $k \geq 1$) such that

$$T_j \cap H \neq \emptyset \neq T_j \setminus H, \qquad 1 \leq j \leq 2k+1,$$

the associated *comb* inequality is

$$x(E(H)) + \sum_{j=1}^{2k+1} x(E(T_j)) \leq |H| + k + \sum_{j=1}^{2k+1} (|T_j| - 2).$$

It is proved in [5] that every comb inequality is facet-inducing for $STSP(V)$.

3. Ladder inequalities. Let H_1 and H_2 be mutually disjoint subsets of V called *handles*. Let $T_1, T_2, \ldots, T_{t+m}$ be pairwise disjoint proper subsets of V called *teeth*, where $t \geq 2$, $m \geq 0$, and $t + m$ is even and at least 4. A tooth T_j is *degenerate* if $T_j \setminus (H_1 \cup H_2) = \emptyset$; otherwise it is *nondegenerate*. Assume that $T_1, T_2, \ldots, T_t$ are nondegenerate teeth and (if $m \geq 1$) that $T_{t+1}, \ldots, T_{t+m}$ are degenerate teeth. Assume also that T_1 intersects only H_1, T_2 intersects only H_2, and T_k, $k = 3, \ldots, t+m$, intersects both H_1 and H_2. T_1 and T_2 are called *pendent teeth*; the others are *nonpendent*. The *ladder inequality* associated with $H_1, H_2, T_1, \ldots, T_{t+m}$ is defined as follows:

$$(2) \quad \sum_{i=1}^{2} x(E(H_i)) + \sum_{j=1}^{t} x(E(T_j)) + \sum_{j=t+1}^{t+m} 2x(E(T_j)) + x(E(T_1 \cap H_1 : T_2 \cap H_2))$$
$$\leq \sum_{i=1}^{2} |H_i| + t + m - 2 + \sum_{j=1}^{t} (|T_j| - d_j - 1) + \sum_{j=t+1}^{t+m} 2(|T_j| - 2),$$

where d_j denotes the number of handles intersected by tooth T_j.

Many of the known classes of valid inequalities for $STSP(V)$ are generalizations of the comb inequalities, and are determined by two families of node subsets, called handles and teeth. These include clique tree inequalities [7], bipartition inequalities [1], and binested inequalities [8]. However, in all of these classes the left-hand side is of the form

$$\sum \alpha_i x(E(H_i)) + \sum \beta_j x(E(T_j)).$$

The last term of the left-hand side of the ladder inequalities does not fit this model. In fact, if that term is dropped, (2) becomes a special kind of bipartition inequality. The smallest ladder inequality (on eight nodes) was introduced in [1] to illustrate a way in which a bipartition inequality can fail to be facet-inducing.

A general ladder inequality $ax \leq a_0$ is presented in Fig. 1(a). Nodes are numbered in such a way that the handles are $H_1 = \{2k : k = 1, 2, \ldots, t+m-1\}$ and $H_2 = \{2k+1 : k = 1, 2, \ldots, t+m-1\}$, and the pendent teeth $T_1 = \{1, 2\}$ and $T_2 = \{h, 3\}$. The hollow nodes w, u, g, and g' are optional; any of them may be present or absent. Any node may appear any number of times, at least once for each node 1,...,7 and h. Additional copies of a node are called *clones* and are discussed in §5. In the dashed box, we allow any *even* number (possibly zero) of additional nonpendent teeth to be present. Every nonpendent tooth may be either nondegenerate (if a node like g or g' is present) or degenerate (if there is no such node). In the latter case, the tooth is contained in the union of the handles. Every coefficient a_{ij} in the corresponding ladder inequality $ax \leq a_0$ is determined by the total weight of all sets containing both nodes i and j. The weights for the degenerate teeth are 2. (For instance, if node g in Fig. 1(a) does not exist, then tooth $\{6, 7\}$ is degenerate and thus has weight 2.) All other weights are 1. The weights are not shown on the figure, to avoid overcrowding it. The fourth term on the left-hand side of inequality (2) is represented by a bipartite graph, reduced to a single edge in Fig. 1(a). Finally, the right-hand side a_0 is as given in inequality (2). Part (b) of Fig. 1 is explained in §4.

We now prove the validity of the ladder inequalities. For $i = 1, 2$, let $\hat{T}_i = T_i \setminus H_i$ and $\hat{H}_i = H_i \setminus (\cup_{j=1}^{t+m} T_j)$.

THEOREM 3.1. *The ladder inequality* (2) *is valid for* $STSP(V)$.

Proof. Add the following valid inequalities for $STSP(V)$, and divide the resulting inequality by 3:

(i) the comb inequality obtained by deleting $\hat{H}_2$ and T_2,

(ii) the comb inequality obtained by deleting $\hat{H}_2$, T_2, and $H_2 \cap T_j$ for $j = 3, \ldots, t$,

(iii) the sum of the degree constraints for each $v \in H_2$,

(iv) the sum of the degree constraints for each $v \in (T_1 \cap H_1) \cup (T_2 \cap H_2)$,

(v) the SE inequality for $(\cup_{j=3}^{t+m} T_j) \cup \hat{H}_1 \cup \hat{H}_2$,

(vi) the sum of the SE inequalities for $T_j \cap H_1$, $j = 3, \ldots, t$,

(vii) the sum of the SE inequalities for $\hat{T}_1$, T_2, and $T_2 \cap H_2$,

(viii) twice the sum of the SE inequalities for $T_j \cap H_2$, $j = 3, \ldots, t+m$,

(ix) twice the sum of the SE inequalities for T_j, $j = t+1, \ldots, t+m$,

(x) twice the sum of the SE inequalities for $T_j \cap H_1$, $j = t+1, \ldots, t+m$,

(xi) twice the SE inequality for $\hat{T}_2$.

It is straightforward to check that for all edges e, the integer part of the coefficient

of x_e in the resulting inequality is its coefficient in (2). The right-hand side RHS is

$$\begin{aligned}
\text{RHS} &= \frac{1}{3}\left(|H_1| + |T_1| - 2 + \sum_{j=3}^{t+m}(|T_j| - 2) + \frac{t+m-2}{2}\right) \\
&+ \frac{1}{3}\left(|H_1| + |T_1| - 2 + \sum_{j=3}^{t}(|T_j \setminus H_2| - 2) + \sum_{j=t+1}^{t+m}(|T_j| - 2) + \frac{t+m-2}{2}\right) \\
&+ \frac{2}{3}|H_2| + \frac{2}{3}|T_1 \cap H_1| + \frac{2}{3}|T_2 \cap H_2| + \frac{1}{3}\left(\sum_{j=3}^{t+m}|T_j| + |\hat{H}_1| + |\hat{H}_2| - 1\right) \\
&+ \frac{1}{3}\sum_{j=3}^{t}(|T_j \cap H_1| - 1) + \frac{1}{3}(|\hat{T}_1| - 1) + \frac{1}{3}(|T_2| - 1) + \frac{1}{3}(|T_2 \cap H_2| - 1) \\
&+ \frac{2}{3}\sum_{j=3}^{t+m}(|T_j \cap H_2| - 1) + \frac{2}{3}\sum_{j=t+1}^{t+m}(|T_j| - 1) + \frac{2}{3}\sum_{j=t+1}^{t+m}(|T_j \cap H_1| - 1) + \frac{2}{3}(|\hat{T}_2| - 1) \\
&= \sum_{i=1}^{2}|H_i| + t + m - 2 + \sum_{j=1}^{t}(|T_j| - d_j - 1) + \sum_{j=t+1}^{t+m}2(|T_j| - 2) + \frac{2}{3}.
\end{aligned}$$

Rounding down each coefficient and the right-hand side to the nearest integer, we obtain the desired result. □

4. Primitive ladder inequalities. For any inequality $ax \leq a_0$, we define its *support graph* to be $G_a = (V, E_a)$, where $E_a = \{e \in E : a_e \neq 0\}$. In this section, we consider a subclass of ladder inequalities $ax \leq a_0$ that have a *spanning* support graph (that is, G_a contains no isolated nodes) and satisfy the following properties:

- $|H_i \cap T_j| \leq 1$ for any pair H_i and T_j,
- $|T_j \setminus (H_1 \cup H_2)| = 1$ for $j = 1, \ldots, t$, and
- $|H_i \setminus (\cup_{j=1}^{t+m} T_j)| \leq 1$ for $i = 1, 2$.

The inequalities in this class are called *primitive* ladder inequalities. Thus, Fig. 1(a) shows a general primitive ladder inequality if no node has any clone. (Hollow nodes may be present or absent, and there may be any even number of teeth in the dashed box).

Note that any $ax \leq a_0$ can be written in the form

$$\sum_{i=1}^{l}\omega_i x(E(L_i)) + bx \leq a_0,$$

where the L_i's are subsets of V. By *complementing* L_i with respect to $ax \leq a_0$, we mean adding to the inequality the multiples of degree constraints $-\frac{\omega_i}{2}x(\delta(v)) = -\omega_i$ for all $v \in L_i$ and $\frac{\omega_i}{2}x(\delta(v)) = \omega_i$ for all $v \in V \setminus L_i$. The resulting inequality is clearly equivalent to $ax \leq a_0$ but has different coefficients. To facilitate the polyhedral proof, we need a *unique* representation of valid inequalities for $STSP(V)$. This representation is given by the following lemma.

LEMMA 4.1. *Let $ax \leq a_0$ be any valid inequality for $STSP(V)$, and let h, u, and v be any three distinct nodes in V. Define $B \equiv \delta(h) \cup \{(u, v)\}$. Then there is a unique (up to positive multiples) inequality $cx \leq c_0$ that is equivalent to $ax \leq a_0$ and satisfies $c_e = 0$ for all $e \in B$.*

The lemma follows directly from Remark 4.2 in Grötschel and Padberg [5] by observing that B corresponds to a basis of the column vectors in the node-edge incidence matrix. We call such a representation, $cx \leq c_0$, an (h, uv)-*canonical form*, or

an (h, uv)-*canonical inequality.* An example of a ladder inequality in $(h, 13)$-canonical form $cx \leq c_0$ is presented in Fig. 1(b). This can be obtained by complementing tooth T_2. Note that $c_{31} = c_{3i} = 0$ for all $i \geq 4$ and even, $c_{21} = c_{24} = c_{26} = 2$, $c_{52} = c_{51} = 1$, etc. Note also that if g is absent, then $c_{67} = 3$.

For any valid inequality $bx \leq b_0$ for $STSP(V)$, a Hamiltonian cycle C on V is said to be b-*tight* if $b(C) = b_0$, where $b(C) \equiv \sum_{e \in C} b_e$.

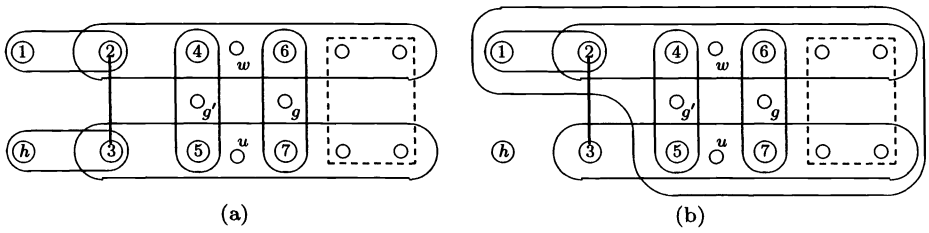


FIG. 1. *Ladder inequalities.* (a) *A ladder inequality.* (b) *The ladder in* $(h, 13)$*-canonical form.*

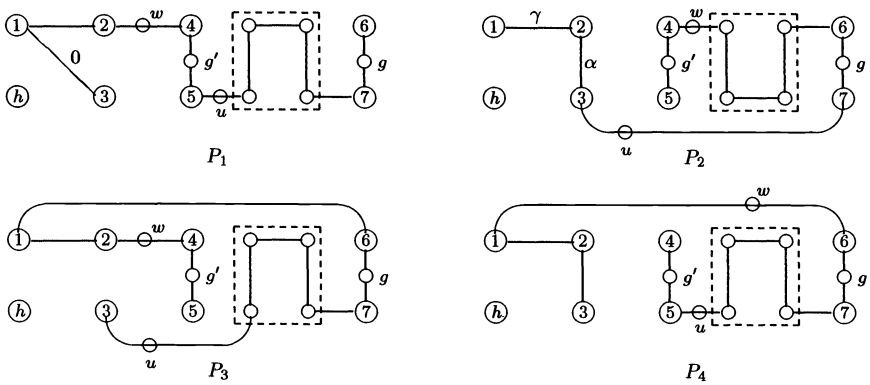


FIG. 2. *Four c-tight paths.*

We now outline the polyhedral proof. In this proof, we make reference to the general primitive ladder inequality shown in Fig. 1. In particular, we use the node labels (numbers 1,...,6, and letters u, w, g, g') as shown in that figure. The hollow nodes g, g' may be assigned to nondegenerate teeth, $\{6, g, 7\}$ and $\{4, g', 5\}$, respectively, as needed in the proof. The other hollow nodes w and u represent the cases that some node in a handle may not be contained in any tooth. Unless otherwise specified, the statements of the proof are true with and without any subset of hollow nodes.

Let $cx \leq c_0$ be the $(h, 13)$-canonical ladder inequality shown in Fig. 1(b), and let $fx \leq f_0$ be a facet-inducing $(h, 13)$-canonical inequality that *dominates* $cx \leq c_0$, that is, such that, for all $x \in STSP(V)$, $cx = c_0$ implies $fx = f_0$. Since $f_e = c_e = 0$ for all edges e in $\delta(h)$, the *star* of h, any c-tight Hamiltonian path P, that is, $c(P) = c_0$, on $V \setminus \{h\}$, is also f-tight, that is, $f(P) = f_0$. (Indeed, path P can be converted, in a unique way, into a c-tight cycle C by connecting its endnodes to node h, and thus $f_0 = f(C) = f(P)$.) Therefore, it suffices to *compare* pairs of c-tight paths on $V \setminus \{h\}$: P and P', that is, compute $f(P) - f(P') = 0$ to derive the coefficients of $fx \leq f_0$. Each comparison and its implication are denoted by

$$P \sim P' \Longrightarrow \text{“some expression.”}$$

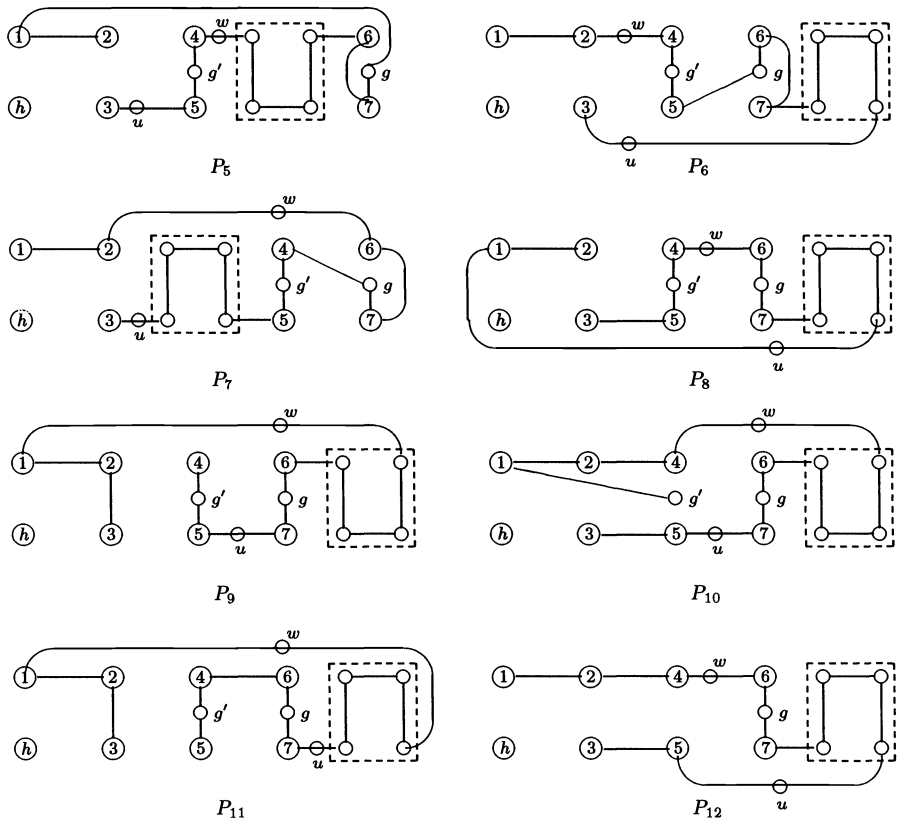


FIG. 3. *Eight other c-tight paths.*

Note that the above implication may involve some obvious node (or tooth) permutations and use earlier results on the f coefficients. Such steps are iterated until $fx \leq f_0$ is shown to be some multiple α of $cx \leq c_0$. It then follows that $cx \leq c_0$, hence $ax \leq a_0$, is facet-inducing.

Figures 2 and 3 present 12 types of c-tight paths on $V \setminus \{h\}$ used in the proof. Each path may be represented by either the corresponding edge set or the sequence of nodes.

We are now in a position to prove the following result.

PROPOSITION 4.2. *All primitive ladder inequalities are facet-inducing.*

Proof. For simplicity, let + stand for set union and − for set difference. Let $\alpha \equiv f_{23}$ and $\gamma \equiv f_{12}$.

CLAIM 1. $f_e = 0$ for all e such that $c_e = 0$.

Proof. Since by definition $f_{13} = 0$, $P_1 \sim P_1 - (1,3) + (3,6) \Longrightarrow f_{3i} = f_{13} = 0$ for all $i \geq 4$ and even.

Next, for any nondegenerate tooth, say, $\{6, g, 7\}$, let $P_1' \equiv P_1 - (7,g) + (7,6) = (312w4g'5 \cdots 76g)$.

Then $P_1' \sim P_1' - (1,3) + (3,g) \Longrightarrow f_{3g} = 0$ for all g. If node w does not exist, we are done; else consider edge $(3,w)$. Let $P_1'' \equiv P_1 - (2,w) + (2,6) = (3126g7 \cdots u5g'4w)$.

Then $P_1'' \sim P_1'' - (1,3) + (3,w) \Longrightarrow f_{3w} = 0$. □

CLAIM 2. $f_e = \alpha$ for all e such that $c_e = 1$.

Proof. $P_2 \sim P_2 - (2,3) + (1,5) \Longrightarrow f_{1i} = \alpha$ for all $i \geq 5$ and odd.

$P_{11} \sim P_{11} - (2,3) + (3,5) \Longrightarrow f_{3i} = \alpha$ for all $i \geq 5$ and odd.

$P_2 \sim P_2 - (2,3) + (2,5) \Longrightarrow f_{2i} = \alpha$ for all $i \geq 5$ and odd.

$P_3 \sim P_3 - (1,6) + (3,5) \Longrightarrow f_{1i} = \alpha$ for all $i \geq 4$ and even.

$P_3 \sim P_3 - (1,6) + (5,6) \Longrightarrow f_{ij} = \alpha$ for all $i, j \geq 4$ such that i and j belong to both different teeth and different handles.

If there is a nondegenerate tooth, $\{6, g, 7\}$, use three types of c-tight paths P_5, P_6, and P_7.

$P_5 \sim P_5 - (1,g) + (2,3) \Longrightarrow f_{1g} = \alpha$.

$P_5 \sim P_5 - (1,g) + (2,g) \Longrightarrow f_{2g} = \alpha$.

$P_6 \sim P_6 - (5,g) + (1,g) \Longrightarrow f_{ig} = \alpha$ for all $i \geq 5$, $i \neq 7$ and odd.

$P_7 \sim P_7 - (4,g) + (1,4) \Longrightarrow f_{ig} = \alpha$ for all $i \geq 4$, $i \neq 6$ and even.

If there are at least two nonpendent, nondegenerate teeth, say, $\{6, g, 7\}$ and $\{4, g', 5\}$, we define $P_6' \equiv P_6 - (4,g') + (4,5) - (5,g) + (g,g') = (12w45g'g67 \cdots u3)$. Then we have

$P_6' \sim P_6' - (g,g') + (1,g) \Longrightarrow f_{gg'} = \alpha$.

If all nodes in the handles are contained in the union of teeth, we are done. Otherwise, do the following.

(i) If node w exists, the values of f_e for all edges $e \in \delta(w)$ such that $c_e = 1$ are derived as follows.

$P_4 \sim P_4 - (1,w) + (1,4) \Longrightarrow f_{1w} = \alpha$.

Let $P_3' \equiv P_3 - (2,w) - (4,w) + (2,4) - (1,6) + (1,w) + (6,w) = (5g'421w6g7 \cdots u3)$ and, if g' exists, $P_3'' \equiv P_3' - (4,g') + (4,5) = (g'5421w6g7 \cdots u3)$.

$P_3' \sim P_3' - (1,w) + (5,w) \Longrightarrow f_{5w} = f_{1w} = \alpha$. So $f_{kw} = \alpha$ for all $k \geq 5$ and odd.

$P_3'' \sim P_3'' - (1,w) + (g',w) \Longrightarrow f_{g'w} = f_{1w} = \alpha$.

When both w and u exist, construct $P_3''' \equiv (u5g'421w6g7 \cdots 3)$.

$P_3''' \sim P_3''' - (1,w) + (u,w) \Longrightarrow f_{uw} = f_{1w} = \alpha$.

(ii) If node u exists, the values of f_e for all edges $e \in \delta(u)$ such that $c_e = 1$ are derived as follows.

$P_3 \sim P_3 - (3,u) + (3,5) \Longrightarrow f_{3u} = f_{35} = \alpha$.

$P_2 \sim P_2 - (3,u) + (1,u) \Longrightarrow f_{1u} = f_{3u} = \alpha$.

$P_8 \sim P_8 - (1,u) + (2,u) \Longrightarrow f_{2u} = f_{1u} = \alpha$.

For any nondegenerate tooth, $(354g'u7g6 \cdots 21) \sim (g'453u7g6 \cdots 21) \Longrightarrow f_{g'u} = f_{3u} = \alpha$.

Let $P \equiv (12w4g'5u3 \cdots 7g6)$. $P \sim P - (3,u) + (6,u) \Longrightarrow f_{6u} = f_{3u} = \alpha$. So $f_{ku} = \alpha$ for all $k \geq 4$ and even.

This completes the proof for Claim 2. □

CLAIM 3. $f_e = \gamma$ for all e such that $c_e = 2$.

Proof. $P_4 \sim P_4 - (1,2) + (2,4) \Longrightarrow f_{2i} = \gamma$ for all $i \geq 4$ and even.

To derive the remaining f_e in the handles with $c_e = 2$, we distinguish, for node w and for node u, the cases with or without that node.

(i) If node w does not exist, then $P_8 \sim P_8 - (4,6) + (2,4) \Longrightarrow f_{ij} = \gamma$ for all distinct $i, j \geq 4$ and even. Otherwise, P_8 includes w and we have

$P_8 \sim P_8 - (4,w) + (2,4) \Longrightarrow f_{kw} = \gamma$ for all $k \geq 4$ and even.

Defining $P_8' \equiv P_8 - (6,w) + (2,w) = (35g'4w21u \cdots 7g6)$, we also have

$P_8' \sim P_8' - (4,w) + (4,6) \Longrightarrow f_{ij} = \gamma$ for all distinct $i, j \geq 4$ and even, and

$P_8 \sim P_8' \Longrightarrow f_{2w} = f_{6w} = \gamma$.

(ii) If node u does not exist, then $P_9 \sim P_9-(5,7)-(2,3)+(2,4)+(3,5) \Longrightarrow f_{ij} = \gamma$ for all distinct $i, j \geq 5$ and odd. Otherwise, P_9 includes u and we have

$P_9 \sim P_9 - (2,3) - (5,u) + (2,4) + (3,5) \Longrightarrow f_{ku} = \gamma$ for all $k \geq 5$ and odd, and

$P_3 \sim P_3 - (u,v) - (1,6) + (5,v) + (1,u) \Longrightarrow f_{5v} = f_{uv} = \gamma$, where $(u,v) \in P_3$, $v \geq 7$ and odd. This shows that $f_{ij} = \gamma$ for all distinct $i, j \geq 5$ and odd.

For any nondegenerate tooth $\{4, g', 5\}$, we have

$P_{10} \sim P_{10} - (v,5) + (5,g') \Longrightarrow f_{5g'} = \gamma$, where $v = u$ if u exists and $v = 7$ otherwise.

$P_{10} \sim P_{10} - (2,4) + (4,g') \Longrightarrow f_{4g'} = \gamma$.

$P_{11} \sim P_{11} - (4,g') + (4,5) \Longrightarrow f_{45} = f_{4g'} = \gamma$.

This completes the proof for Claim 3. □

CLAIM 4. $\gamma = 2\alpha$.

Proof. By Claims 1, 2, and 3, $P_1 \sim P_4 \Longrightarrow \gamma = 2\alpha$. □

CLAIM 5. For every degenerate tooth T, say $T = \{4,5\}$ (without g'), we have $f_{45} = 3\alpha$.

Proof. $P_8 \sim P_{12} \Longrightarrow f_{45} = 2\gamma - \alpha = 4\alpha - \alpha = 3\alpha$. □

From Claims 1–5, it follows that $f_e = \alpha c_e$ for all $e \in E(V)$. The proof of Proposition 4.2 is complete. □

5. Lifting ladder inequalities. We have shown that all *primitive* ladder inequalities are facet-inducing for STS polytopes. In this section, we show by node lifting and cloning that *all* ladder inequalities are facet-inducing. We begin with the following simple lemma on (h, uv)-canonical forms, which is used in our proofs.

LEMMA 5.1. *Let $cx \leq c_0$ be an (h,uv)-canonical facet-inducing inequality for $STSP(V)$. If an (h,uv)-canonical inequality $fx \leq f_0$ satisfies $f(P) = f_0$ for all c-tight paths P on $V \setminus \{h\}$, then $f = c$ and $f_0 = c_0$, up to a positive multiple.*

Proof. Assume that $cx \leq c_0$ and $fx \leq f_0$ satisfy the assumptions of the lemma. Consider any c-tight cycle C and let $P \equiv C \setminus \delta(h)$. Since P is a Hamiltonian path on $V \setminus \{h\}$ and $c(P) = c(C) = c_0$, we have $f(P) = f_0$, implying $f(C) = f_0$. Since $cx \leq c_0$ is facet-inducing and both $cx \leq c_0$ and $fx \leq f_0$ are in (h,uv)-canonical form, this implies $f = c$ and $f_0 = c_0$, up to a positive multiple. □

We say that a valid inequality induces a *nontrivial* facet if it is not equivalent to either a nonnegativity constraint $x_e \geq 0$ or a bound constraint $x \leq 1$. The following two results show how large classes of nontrivial facets can be obtained by node-lifting.

The first theorem allows us to add *isolated nodes*, that is, nodes that are not in the union of all handles and teeth, and therefore whose incident edges have zero coefficients in the ladder inequality (1). Actually, this node-lifting theorem applies to a broad class of STSP facet-inducing inequalities, such as the well-known clique tree class. An inequality $ax \leq a_0$ for $STSP(V)$ is a *2-tooth inequality* if it satisfies:

(i) it is a nontrivial valid inequality for $STSP(V)$;

(ii) $a \geq 0$;

(iii) there exist (at least) two disjoint *teeth* $T_1 = \{t_1, h_1\}$ and $T_2 = \{t_2, h_2\}$ such that for each $i = 1,2$, we have $a_{t_i h_i} > 0$, and $a_{t_i v} = 0$ for all $v \neq h_i$;

(iv) either $a_{h_1 v} \geq a_{h_1 t_1}$ or $a_{h_1 v} = 0$ for all $v \in V$.

Many of the known valid inequalities have this property, including all primitive clique tree, ladder, and chain inequalities as well as many bipartition inequalities.

THEOREM 5.2. (Adding an isolated node). *Suppose that the 2-tooth inequality $ax \leq a_0$ defines a nontrivial facet of $STSP(V)$, and $q \notin V$. Let $a^*x^* \leq a_0^*$ be a lifted inequality for $STSP(V^*)$, where $V^* = V \cup \{q\}$, obtained by letting $a_0^* = a_0$, $a_e^* = a_e$ for all $e \in E(V)$ and zero otherwise. Then $a^*x \leq a_0^*$ is facet-inducing for $STSP(V^*)$.*

Proof. Consider a facet-inducing 2-tooth inequality $ax \leq a_0$. Without loss of generality, we may assume that $a_{t_1h_1} = 1$. Define $Y \equiv \{v \in V \setminus \{t_1\} : v = h_1 \text{ or } a_{h_1v} > 0\}$ and $Z \equiv V \setminus T_1$. Note that (i) implies that both Y and Z are nonempty. Since $h_1 \in Y$ and $t_2 \in Z \setminus Y$, both Y and $Z \setminus Y$ are nonempty subsets of $V \setminus \{t_1\}$. Let $cx \leq c_0$ be the (t_1, t_2h_1)-canonical inequality obtained from $ax \leq a_0$ by complementing T_1. It is easily verified that this inequality satisfies the following properties.

(P1) $c \geq 0$ and the support graph $G_c = (V, E_c)$ of $cx \leq c_0$ consists of the isolated node t_1 and a *bi-clique structure* induced by subsets Z and Y of V; that is, $E_c = E(Z) \cup E(Y)$, where $Z \cup Y = V \setminus \{t_1\}$, and $Y \setminus Z = \{h_1\}$.

(P2) $c_e \geq 1$ for all $e \in E(Z)$.

(P3) $c_{h_1v} \geq 1$ for all $v \in Y$ and $c_{h_1v} = 0$ for all $v \in Z \setminus Y$, and

(P4) $c_{t_2h_1} = 0$; $c_{t_2h_2} > 1$ and $c_{t_2v} = 1$ for all $v \in Z \setminus \{h_2\}$.

Let $a^*x \leq a_0^*$ be as defined in the theorem. Conditions (i) and (ii) imply that $a^*x \leq a_0^*$ is valid for $STSP(V^*)$. Let $c^*x \leq c_0^*$ be the (t_1, t_2h_1)-canonical inequality obtained from $a^*x \leq a_0^*$ by complementing the tooth $\{t_1, h_1\}$. Comparing this inequality with the (t_1, t_2h_1)-canonical inequality $cx \leq c_0$, we observe that $c_e^* = c_e$ for all $e \in E(V)$, that $c_{qh_1}^* = 0$ and $c_{qv}^* = 1$ for all $v \in Z$, and that $c_0^* = c_0 + 1$.

Let $fx \leq f_0$ be any (t_1, t_2h_1)-canonical facet-inducing inequality for $STSP(V^*)$ that dominates $c^*x \leq c_0^*$. Let $\alpha \equiv f_{qt_2}$.

CLAIM 1. $f_{qh_1} = 0$ and $f_{qz} = \alpha$ for all $z \in Z \setminus Y$.

Proof. We have assumed that $ax \leq a_0$, and thus $cx \leq c_0$ as well, is not equivalent to a trivial inequality $x_e \geq 0$. Therefore, for every $z \in Z \setminus Y$, there exists a c-tight path P on $V \setminus \{t_1\}$ containing edge (z, h_1). By (P3), $c_{zh_1} = 0$, and thus the edge e connecting the endnodes of P satisfies $c_e = 0$, for otherwise $c(P \cup \{e\} \setminus \{(z, h_1)\}) > c_0$. This implies by (P1) that path P has the form $P = (u \cdots zh_1)$ with $c_{uh_1} = 0$ and $u \in Z \setminus Y$. Let $P' \equiv P \cup (q, u)$, $P'' \equiv (h_1qu \cdots z)$ and note that both P' and P'' are c^*-tight paths on $V^* \setminus \{t_1\}$.

(i) First, let $z = t_2$. Comparing P' with the c^*-tight path $(h_1qu \ldots t_2)$ implies $f_{qh_1} = f_{h_1t_2} = 0$.

(ii) Next, comparing P'' with $(h_1qz \cdots u)$ yields $f_{qz} = f_{qu}$.

(iii) Now, consider any other $z \in Z \setminus Y$, $z \neq t_2$. If $u = t_2$, then comparing P' and the c^*-tight path $(u \ldots zqh_1)$ yields $f_{qz} = \alpha$ and Claim 1 is proved for node z. Else, $u \neq t_2$ and we may write $P = (u \cdots vt_2s \cdots zh_1)$. By (P4), we have $c_{vt_2} = 1$ or $c_{t_2s} = 1$ (or both). If $c_{vt_2} = 1$, then comparing c^*-tight paths $(h_1u \cdots vqt_2s \cdots z)$ and $(h_1u \cdots vqz \cdots st_2)$ yields $f_{qz} = f_{qt_2} = \alpha$. If $c_{t_2s} = 1$, then comparing $(u \cdots vt_2qs \cdots zh_1)$ and $(t_2v \cdots uqs \cdots zh_1)$ yields $f_{qu} = \alpha$, and therefore by (ii), $f_{qz} = f_{qu} = \alpha$. We have shown that $f_{qz} = \alpha$ for all $z \in Z \setminus Y$ and the proof of Claim 1 is complete. □

CLAIM 2. $f_{qw} = \alpha$ for all $w \in Z \cap Y$.

Proof. Since $cx \leq c_0$ is a nontrivial inequality, for any $w \in Z \cap Y$, there exists a c-tight cycle C on V containing edge (t_1, w). Thus, there exists a c-tight path $P \equiv (w \cdots s)$ on $V \setminus \{t_1\}$, obtained by deleting from C the edges incident with t_1. Note that, by (P2) and (P3), $c_{ws} \geq 1$. By property (P4), path P must contain an edge (u, v) incident with t_2 and with $c_{uv} = 1$. (Otherwise, P would contain (h_2, t_2) and (t_2, h_1) with $c_{t_2h_1} = 0$, implying that $c(P \cup \{(w, s)\} \setminus \{(t_2, h_1)\}) \geq c_0 + 1$, a contradiction.) Let $P \equiv (w \ldots uv \ldots s)$ and $P' \equiv P \cup \{(w, s)\} \setminus \{(u, v)\}$. Comparing P' and P yields $c_{ws} \leq 1$ and therefore $c_{ws} = 1$. Thus P' is also a c-tight path on $V \setminus \{t_1\}$. Now comparing $P' \cup \{(q, u)\}$ and $P' \cup \{(q, v)\}$ yields $f_{qu} = f_{qv} = \alpha$, since $t_2 \in \{u, v\}$. Finally, comparing the two c^*-tight paths $(w \cdots uqv \cdots s)$ and

$(u \cdots wqv \cdots s)$, we obtain $f_{qw} = f_{qu} = \alpha$. The proof of Claim 2 is complete. □

Consider the following inequality for $STSP(V)$,

$$\sum_{e \in E(V \setminus \{t_1\})} f_e x_e \leq f_0 - \alpha. \tag{3}$$

Denote this inequality by $\hat{f}x \leq \hat{f}_0$ and observe that it is in $(t_1, t_2 h_1)$-canonical form. Consider any Hamiltonian path P on $V \setminus \{t_1\}$, say $P = (u \ldots v)$. By property (P1), P must have at least one endnode v in Z. Letting $P^* \equiv (u \ldots vq)$, we have $f_0 \geq f(P^*) = \hat{f}(P) + \alpha$. This shows that inequality (3) is satisfied by any Hamiltonian path on $V \setminus \{t_1\}$. Furthermore, if P is c-tight on $V \setminus \{t_1\}$, then P^* is c^*-tight, and therefore also f-tight, on $V^* \setminus \{t_1\}$. That is, $f_0 = f(P^*) = \hat{f}(P) + \alpha$. Thus, every c-tight path on $V \setminus \{t_1\}$ satisfies (3) with equality. Since $cx \leq c_0$ is facet-inducing for $STSP(V)$, Lemma 5.1 implies that, with the appropriate positive multiple, $c_0 = \hat{f}_0 = f_0 - \alpha$ and $c^*_e = c_e = \hat{f}_e = f_e$ for all $e \in E(V \setminus \{t_1\})$.

Finally, from the c-tight path $P = (w \cdots uv \cdots s)$ in the proof of Claim 2, we obtain two c^*-tight paths $(w \cdots uqv \cdots s)$ and $(qw \cdots uv \cdots s)$. Since $c_{uv} = 1$, we have $f_{uv} = 1$. Therefore comparing these paths yields $\alpha + 1 = 2\alpha$. So $\alpha = 1$.

This shows that $f_0 = c_0 + 1 = c^*_0$ and $f_e = c^*_e$ for all $e \in E(V^*)$, implying that $c^*x \leq c^*_0$, or equivalently $a^*x \leq a^*_0$, is facet-inducing for $STSP(V^*)$. The proof of Theorem 5.2 is complete. □

We remark that the above theorem is not only of theoretical interest but also of practical importance in polyhedral computations for the TSP. Since all facet-inducing 2-tooth inequalities for small STS polytopes also induce facets for large STS polytopes by adding isolated nodes, they can be effectively used as cutting planes for solving the large TSPs. Moreover, they have small support graphs, and thus require far less computer memory to store. As a consequence, we may expect facet-inducing 2-tooth inequalities derived from the study of small STS polytopes to play a role in the efficient solution of large STS problems. Denis Naddef pointed out to us that an example arose in computation for which ladder inequalities improved the LP bound. This example is discussed in detail in [4].

To show that any ladder inequality is facet-inducing, we use the following *node-cloning* result, which is an extension of Theorem 4.1 in Queyranne and Wang [10].

THEOREM 5.3. (A sufficient condition for node cloning). *Let u and q be any two nodes such that $u \in V$ and $q \notin V$. Let $V^* \equiv V \cup \{q\}$. Assume that $cx \leq c_0$ is a nontrivial facet-inducing (u, pw)-canonical inequality for $STSP(V)$ satisfying $c_e \geq 1$ for all e with $c_e \neq 0$, and moreover the following condition.*

CONDITION $\mathcal{B}(u, D; \omega)$. *There exists a scalar $\omega \geq 1$ and a partition $(\{u\}, D, U, U')$ of V such that:*

B1. $c_e = 0$ *for all* $e \in E(D : U')$;

B2. $1 \leq c_e \leq \omega$ *for all* $e \in E(D : U)$; *and*

B3. $c_e \geq \omega$ *for all* $e \in E(U)$.

Then the inequality $c^u x \leq c^u_0$, defined by $c^u_0 = c_0$, $c^u_e = c_e$ for all $e \in E(V)$ and $c^u_e = 0$ for all $e \in \delta(q)$, is facet-inducing for $STSP(V^*)$.

Proof. Let $d \in D$, and let $fx \leq f_0$ be a (u, qd)-canonical inequality that dominates $c^u x \leq c^u_0$ and defines a facet of $STSP(V^*)$.

CLAIM 1. $f_{qv} = 0$ for all $v \in U' \cup D$.

Proof. Consider any nodes $v \in D$ and $v' \in U'$. Note $c_{vv'} = 0$ by (B1). Since $cx \leq c_0$ is not equivalent to any $x_e \geq 0$, there is a c-tight cycle C on V, thus

a c-tight path $P \equiv C \setminus \delta(u) = (s \cdots vv' \cdots t)$ on $V \setminus \{u\}$ containing (v, v'). Let $P' \equiv P \cup \{(s,t)\} \setminus \{(v,v')\}$. Since $0 \leq c(P) - c(P') = c_{vv'} - c_{st}$, P' is also c-tight. Comparing $P' \cup \{(q,v)\}$ and $P' \cup \{(q,v')\}$ yields $f_{qv} = f_{qv'}$. Since $f_{qd} = 0$, the claim follows.

CLAIM 2. $f_{qv} = 0$ for all $v \in U$.

Proof. Consider any node $v \in U$. Let C' be a c-tight cycle on V containing uv. Then $P' \equiv C' \setminus \delta(u) = (v \cdots v')$ is the c-tight path on $V \setminus \delta(u)$. If $v' \in U' \cup D$ then construct two c-tight paths as in (i) to show that $f_{qv} = 0$. Otherwise $v' \in U$. In this case P' has the form $(v \cdots rs \cdots v')$ where $r \in U$ and $s \in D$. (Note that P' contains no edge $e_0 \in E(D : U')$, since otherwise $c(P' \cup \{(vv')\} \setminus \{e_0\}) > c_0$, a contradiction.) By (B2) and (B3), $P'' \equiv P' \cup \{(r,v')\} \setminus \{(r,s)\}$ is a c-tight path on $V \setminus \delta(u)$. Comparing $P'' \cup \{(q,v)\}$ and $P'' \cup \{(q,s)\}$ yields $f_{qv} = f_{qs} = 0$. So Claim 2 also holds.

Finally, consider any c-tight cycle C on V. Clearly $C^* = C \cup \{(q,u),(q,v)\} \setminus \{(u,v)\}$, where $(u,v) \in C \cap \delta(u)$, is a Hamiltonian cycle on V^* satisfying $c^u(C^*) = c_0^u$, and hence is f-tight. Furthermore using the above claim, we have $f(C) = f(C^*) = f_0$. Since $cx \leq c_0$ defines a facet, by Lemma 5.1, we have $f_e = c_e$ for all $e \in E(V)$ and $f_0 = c_0$. □

THEOREM 5.4. *All ladder inequalities are facet-inducing.*

Proof. Let $bx \leq b_0$ be any ladder inequality. Clearly, there exists a corresponding facet-inducing primitive ladder inequality $a'x \leq a'_0$ obtained by discarding all isolated nodes in G_b and shrinking each nonempty set $H_i \cap T_j$, $T_j \setminus (H_1 \cup H_2)$ and $H_i \setminus \left(\cup_{j=1}^{t+m} T_j\right)$ into a singleton set. If G_b contains s isolated nodes, we apply Theorem 5.2 s times to $a'x \leq a'_0$ to obtain a facet-inducing ladder inequality $ax \leq a_0$ with G_a containing s isolated nodes. To clone any other node u, we consider $ax \leq a_0$ as being a general facet-inducing ladder inequality for $STSP(V)$. Recall that $V^* = V \cup \{q\}$. We need to show that the inequality $a^u x \leq a_0^u$, obtained by replacing $\{u\}$ with $\{u,q\}$, is also facet-inducing for $STSP(V^*)$. Let $cx \leq c_0$ and $c^u x \leq c_0^u$ be their respective (u, vw)-canonical inequalities. Then $c^u x \leq c_0^u$ is exactly the inequality obtained in Theorem 5.3 from $cx \leq c_0$. Thus, to show that $a^u x \leq a_0^u$ is facet-inducing, it is enough to check that $cx \leq c_0$ satisfies the conditions of Theorem 5.3 for each of the following cases. (Note that by symmetry, the following also applies to the cases with respect to H_1.)

Case 1. $u \in T_2 \setminus H_2$. Construct $cx \leq c_0$ by complementing T_2, as in Fig. 1(b). Then, $cx \leq c_0$ satisfies the required conditions and $\mathcal{B}(u, T_2 \cap H_2; 1)$.

Case 2. $u \in T_j \setminus (H_1 \cup H_2)$, $3 \leq j \leq t$. Construct $cx \leq c_0$ by complementing T_j. Then, $cx \leq c_0$ satisfies the required conditions and $\mathcal{B}(u, T_j \cap H_2; 1)$.

Case 3. $u \in H_2 \setminus \left(\cup_{j=1}^{t+m} T_j\right)$. Construct $cx \leq c_0$ by complementing H_2. Then, $cx \leq c_0$ satisfies the required conditions and $\mathcal{B}(u, T_2 \cap H_2; 1)$.

Case 4. $u \in H_2 \cap T_j$. $j \geq 3$. Construct $cx \leq c_0$ by complementing H_2 and T_j. Then, $cx \leq c_0$ satisfies the required conditions and $\mathcal{B}(u, T_j \setminus (H_1 \cup H_2); 1)$ if $3 \leq j \leq t$; or $\mathcal{B}(u, T_j \cap H_1; 2)$ if $t+1 \leq j \leq t+m$.

Case 5. $u \in H_2 \cap T_2$. Construct $cx \leq c_0$ by complementing H_2, T_2 and then adding the degree constraints $-x(\delta(s)) = -2$ for all $s \in S \equiv T_1 \cap H_1$. Then, $cx \leq c_0$ satisfies the required conditions and $\mathcal{B}(u, T_2 \setminus H_2; 1)$. (Note that $U = V \setminus (T_2 \cup H_2 \cup (T_1 \cap H_1))$ in the partition $(\{u\}, D, U, U')$.)

The proof is complete. □

6. The Chvátal rank of ladder inequalities. Let P be a rational polyhedron in $\boldsymbol{R}^E$, that is, $P = \{x : Ax \leq b\}$, where A and b are rational, and let P_I denote the convex hull of the integral points in P. Define P^0 to be P and for $i \geq 1$, P^i to be

the set of points satisfying all *integral* inequalities $ax \leq a_0$ derived from P^{i-1} by the following rounding procedure: For any finite set of m (say) inequalities $Cx \leq d$ valid for P^{i-1} and $\lambda \in \boldsymbol{R}_+^m$ such that λC is integral, take $a = \lambda C$ and $a_0 = \lfloor \lambda d \rfloor$. So each P^i contains P_I and $P^0 \supseteq P^1 \supseteq \cdots \supseteq P^i$. These definitions were introduced by Chvátal [3], and the rounding procedure is closely related to the cutting plane methods of Gomory. It can be proved that each P^i is itself a polyhedron, and that there is an integer k, depending on P, such that $P^k = P_I$. (See Chvátal [3] for details.)

The (Chvátal) *rank* of an inequality $ax \leq a_0$ valid for P_I is the least i such that $ax \leq a_0$ is valid for P^i. It is a measure of the complexity of the derivation of the inequality by the above procedure. Suppose that we take P to be a *subtour polytope*, that is, the solution set of (1a), (1b), and (1c). Then P_I is $STSP(V)$, and it is of interest to classify facet-inducing inequalities by their rank. Of course, the nonnegativity and SE inequalities have rank 0. It is well known that comb inequalities have rank 1. See [2]. From this and our proof for the validity of the ladder inequalities, it follows that each ladder inequality has rank at most 2.

In the remainder of this section, we prove that each ladder inequality has rank at least 2, hence exactly 2. There is an apparently "obvious" technique for proving that an integral inequality $ax \leq a_0$, which is valid for P_I, cannot be obtained from inequalities of rank 0 by the rounding procedure. Namely, we show that there is no solution λ to

$$\lambda A = a, \quad \lambda \geq 0, \quad \lambda b < a_0 + 1.$$

By the duality theorem of LP, this is equivalent to showing that there is $\bar{x} \in P$ with $a\bar{x} \geq a_0 + 1$. However, there is a difficulty with this argument. It may be that there are inequalities of which $ax \leq a_0$ is a nonnegative combination, that are obtainable by rounding, although $ax \leq a_0$ itself is not. This difficulty does not disappear even if we know that $ax \leq a_0$ is facet-inducing for P_I, since it still may have an equivalent form that is obtainable by rounding.

An instructive example that arises from the 6-node TSP is the following inequality:

$$ax = x_{12} + x_{13} + x_{23} + 2x_{14} + 2x_{25} + 2x_{36} + x_{45} + x_{46} + x_{56} \leq 8 = a_0.$$

This inequality is facet-inducing for $STSP(V)$ with $|V| = 6$. In fact, it is equivalent to a comb inequality with handle $\{1,2,3\}$ and teeth $\{1,4\},\{2,5\},\{3,6\}$. Hence it has rank 1. However, the point $\bar{x} = \frac{1}{2}a$ satisfies (1a), (1b), and (1c) with $a\bar{x} = 9 = a_0 + 1$.

Actually, this difficulty was overlooked in some previous papers [1], [2], where it was claimed using the above argument that certain inequalities have rank at least 2. These results are correct, but their proofs contain gaps that can be filled by the following result from [11]. Let $Gx = g$ be the *equality system* for P_I, that is, the linearly independent equations whose solution set is the affine hull of P_I.

If G is written (G_B, G_N) such that G_B is a nonsingular square matrix, we say that a valid inequality $ax \leq a_0$ for P_I is an *integral B-canonical form* if $a = (a_B, a_N)$ with $a_B = 0$ and all components of a_N being relatively prime integers. Notice that for every rational valid inequality, there is a unique integral B-canonical form to which it is equivalent.

PROPOSITION 6.1. *Let $ax \leq a_0$ be an integral B-canonical form that is facet-inducing for P_I, and suppose that $G_B^{-1}G$ is integral. Then $ax \leq a_0$ has Chvátal rank at most 1 if and only if $z(a) \equiv \max\{ax : x \in P\} < a_0 + 1$.*

For the *STSP* case, the equality system $Gx = g$ consists of the degree constraints (1a). Consider the integral B-canonical form of Lemma 4.1. It is easy to see that, for any column g_{pq} of G_N, the vector $G_B^{-1} g_{pq}$, that is, the vector d that satisfies $G_B d = g_{pq}$ has components 0, -1, $+1$. Namely, the $+1$ and -1 components alternate on the edges of the unique odd-length edge-simple path in B joining p to q. Hence Proposition 6.1 can be applied.

We are now in a position to prove the main result of this section.

THEOREM 6.2. *The ladder inequality* (2) *has Chvátal rank* 2.

Proof. From the proof of Theorem 3.1, it follows that every ladder inequality $cx \leq c_0$ has Chvátal rank at most 2. We now show that it has Chvátal rank at least 2. To do so, we first construct its $(h, 13)$-canonical form $ax \leq a_0$ where, as in §4 and Fig. 1(a), nodes $h \in T_2 \setminus H_2$, $1 \in T_1 \setminus H_1$ and $3 \in H_2 \cap T_2$. Hence by Proposition 6.1, we just need to construct a feasible solution $\bar{x}$ to the subtour polytope satisfying $a\bar{x} \geq a_0 + 1$.

For $j = 3, \ldots, t+m$, let P_j be a Hamiltonian path on T_j that saturates both $T_j \cap H_1$ and $T_j \cap H_2$ with the endpoints $v_j^1 \in T_j \cap H_1$ and $v_j^2 \in T_j \cap H_2$. Let P_2 be a Hamiltonian path on $V \setminus (\cup_{j=3}^{t+m} T_j)$ that saturates $T_i \setminus H_i$, $H_i \cap T_i$, $\hat{H}_i$ for $i = 1, 2$ with endpoints $v_1 \in H_1$ and $v_2 \in H_2$. Define the edge set

$$P_1 \equiv (\cup_{j=2}^{t+m} P_j) \cup \{(v_{j+i-1}^i, v_{j+i}^i) \in E(H_i) : i = 1, 2;\ j \text{ is even and } 4 \leq j \leq t+m-2\},$$

and node sets $S_1 \equiv \{v_1, v_3^1, v_{t+m}^1\}$, $S_2 \equiv \{v_2, v_3^2, v_4^2\}$. Then P_1 is a path system with all nodes in $S_1 \cup S_2$ of degree 1 and all other nodes of degree 2. Now define $\bar{x} \in \boldsymbol{R}^E$ by $\bar{x}_e = 1$ for all $e \in P_1$, $\bar{x}_e = \frac{1}{2}$ for all $e \in E(S_1) \cup E(S_2)$ and $\bar{x}_e = 0$ otherwise. It is easily verified, using the $(h, 13)$-canonical form $ax \leq a_0$ of the ladder inequality, that we have $a\bar{x} = a_0 + 1$. □

REFERENCES

[1] S. C. BOYD AND W. H. CUNNINGHAM, *Small travelling salesman polytopes*, Math. Oper. Res., 16 (1991), pp. 259–271.

[2] S. C. BOYD AND W. R. PULLEYBLANK, *Optimizing over the subtour polytope of the travelling salesman problem*, Math. Programming, 49 (1991), pp. 163–187.

[3] V. CHVÁTAL, *Edmonds polytopes and a hierarchy of combinatorial problems*, Discrete Math., 4 (1973), pp. 305–337.

[4] J. CLOCHARD AND D. NADDEF, *Using path inequalities in a branch and cut code for the symmetric travelling salesman problem*, The Third IPCO Conference, 1993, Erice, Italy, pp. 291–311.

[5] M. GRÖTSCHEL AND M. W. PADBERG, *On the symmetric travelling salesman problem* II*: lifting theorems and facets*, Math. Programming, 16 (1979), pp. 281–302.

[6] M. GRÖTSCHEL AND M. W. PADBERG, *Polyhedral Theory*, in The Travelling Salesman Problem, E. L. Lawler, J.-K. Lenstra, A. H. G. Rinnooy-Kan, and D. Shmoys, eds., J. Wiley & Sons, New York, 1985, pp. 251–305.

[7] M. GRÖTSCHEL AND W. R. PULLEYBLANK, *Clique tree inequalities and the symmetric travelling salesman problem*, Math. Oper. Res., 11 (1986), pp. 537–569.

[8] D. NADDEF, *The binested inequalities for the symmetric travelling salesman polytope*, Math. Oper. Res., 17 (1992), pp. 882–900.

[9] M. W. PADBERG AND G. RINALDI, *A branch-and-cut algorithm for the resolution of large scale symmetric travelling salesman problems*, SIAM Rev., 33 (1991), pp. 60–100.

[10] M. QUEYRANNE AND Y. WANG, *Hamiltonian path and symmetric travelling salesman polytopes*, Math. Programming, 58 (1993), pp. 89–110.

[11] M. QUEYRANNE AND Y. WANG, *On the Chvátal rank of certain inequalities*, manuscript.

SIAM J. OPTIMIZATION
Vol. 5, No. 2, pp. 421–435, May 1995

010

ON THE CONVERGENCE OF FENCHEL CUTTING PLANES IN MIXED-INTEGER PROGRAMMING*

E. ANDREW BOYD†

Abstract. Fenchel cutting planes are based on the dual relationship between separation and optimization and can be applied in many instances where alternative cutting planes cannot. They are deep in the sense of providing the maximum separation between a point $\hat{x}$ and a polyhedron P as measured by an arbitrary norm which is specified in the process of generating a Fenchel cut. This paper demonstrates a number of fundamental convergence properties of Fenchel cuts and addresses the question of which norms lead to the most desirable Fenchel cuts. The strengths and weaknesses of the related class of 1-polar cuts are also examined.

Key words. integer programming, mixed-integer programming, cutting planes

AMS subject classifications. 90C10, 90C11

1. Introduction.

In the last decade cutting plane methods have come to dominate the research in integer programming. Beyond providing some of the most theoretically attractive results found in discrete optimization, cutting plane methods have proven to be remarkably successful in solving large integer programs in practice. Grötschel and Holland [11], Padberg and Rinaldi [18], [19], and others, have solved larger traveling salesman problems to optimality than would have been conceivable only a decade ago. Crowder, Johnson, and Padberg [9] won the Lanchester prize for solving some integer programs that were at one time considered forever unsolvable.

Recent successful cutting planes commonly arise from theoretical studies of facets of polyhedra P defined by the convex hull of feasible integer points for well-known integer programs. For example, the facial structure of the polyhedron P found in knapsack problems has been studied in great detail in [1], [3], [4], [13], [17], [23], and elsewhere, and these results were used by Crowder, Johnson, and Padberg in [9] to generate cutting planes for the integer programs they studied based on relaxations provided by the individual constraints of the problems. Furthermore, once a class of cutting planes has been identified, for the class to prove useful in practice it must be possible to quickly solve the *separation problem*: find an inequality that is valid for the underlying integer program but violated by the optimal solution of the linear programming relaxation. These two factors — the theoretical study of classes of cutting planes and algorithms for solving the associated separation problem — are the key determining factors in developing useful classes of cutting planes.

An alternative method of generating cutting planes was proposed by the author in [5]. The proposed *Fenchel* cutting planes are based on the dual relationship between separation and optimization and are generated using assumptions different from those used in generating most cutting planes. Rather than assuming any a priori knowledge of the facial structure of a polyhedron P, Fenchel cuts assume only the existence of an oracle for optimizing a linear function on P. This assumption is not new, having been used extensively in another popular technique for solving integer programs, namely, Lagrangian relaxation. The computational value of Fenchel cuts was demonstrated in

* Received by the editors October 18, 1992; accepted for publication (in revised form) December 23, 1993. This work was sponsored in part by National Science Foundation grant DDM-9101578 and the Office of Naval Research.

† Department of Industrial Engineering, Texas A&M University, College Station, Texas 77843-3131 (boyd@marvin.tamu.edu).

[5]–[7]. One of the more interesting uses of Fenchel cuts is in the solution of mixed-integer programs, where alternative cutting plane techniques often cannot be applied.

Fenchel cuts are deep in the sense of providing the maximum separation between a point $\hat{x}$ and a polyhedron P as measured by an arbitrary norm which is specified in the process of generating a Fenchel cut. In a sense, Fenchel cuts can be seen as bypassing the issue of understanding the facial structure of P and attacking the separation problem directly. As will be discussed, the process of generating a Fenchel cut either finds a cutting plane strictly separating $\hat{x}$ from P or provides a proof that no such cutting plane exists.

The purpose of the present paper is to address the convergence properties of an algorithm based on Fenchel cuts for strengthening the linear programming relaxation of an integer program, and to discuss how the polyhedral properties of the problem and the norm associated with the Fenchel cuts affect the speed of convergence of this algorithm. Somewhat surprisingly, it is shown that *finite* convergence is guaranteed under the most general possible conditions, although the provable speed of the algorithm can vary dramatically. The convergence properties of the algorithm using a class of cutting planes different from Fenchel cuts, but very closely related, are also discussed.

In order to be more specific regarding the results to be presented, consider the following arbitrary mixed-integer program.

$$\text{(MIP)} \qquad \begin{array}{ll} \max & cx \\ \text{s.t.} & Ax \leq b \\ & \text{some } x_i \text{ integer.} \end{array}$$

Let P_i, $i = 1, \ldots, m$ be a collection of polyhedra containing the feasible region of (MIP). In practice, such a collection of polyhedra might be defined by the convex hull of feasible solutions for mixed-integer relaxations of (MIP) obtained by eliminating some subset of complicating constraints. The cutting plane algorithm to be considered is the following.

ALGORITHM CUT

Given: A mixed-integer program (MIP) and a collection of m polyhedra P_i containing the feasible region of (MIP).

0. *Initialize.* Let $A^0 = A$, $b^0 = b$, and set $t = 0$.
1. Let x^t be the optimal solution obtained by maximizing cx subject to the constraints $A^t x \leq b^t$.
2. For $i = 1, \ldots, m$ generate a cutting plane separating x^t from P_i or prove that no such cutting plane exists.
3. If no cutting planes were generated in step 2, stop. Otherwise, append all cutting planes generated in step 2 to the system $A^t x \leq b^t$, set $t = t + 1$, and return to step 1.

It is clear that upon termination of the algorithm the point $x^t \in \bigcap_{i=1}^m P_i$. What is not clear is that the algorithm will necessarily terminate or, if not, if the sequence of iterates x^t will converge to a point in $\bigcap_{i=1}^m P_i$.

Algorithm Cut is an idealized version of an approach that has been used in solving many integer programs. Some algorithms explicitly employ Algorithm Cut, such as the award winning work of Crowder, Johnson, and Padberg for solving general 0/1 integer programs [9]. Other algorithms employ Algorithm Cut implicitly. For example, cutting plane algorithms for the well-known traveling salesman problem (TSP) commonly generate facet-inducing cuts for various polyhedra that are relaxations of the

TSP polyhedron, e.g., the subtour elimination polyhedron and the 2-matching polyhedron [19]. Such cutting plane algorithms are implicitly employing Algorithm Cut, although it is not common to view the algorithms in this way due to the context in which the cutting planes are generated. When facet-inducing cuts of the polyhedra P_i are used, then finite termination of Algorithm Cut is guaranteed by virtue of the fact that a polyhedron has a finite number of facets. On the other hand, it is possible to construct very simple examples of sequences of cuts — even face-inducing cuts — for which Algorithm Cut does not terminate and the sequence of iterates x^t generated by the algorithm do not converge to a point in $\bigcap_{i=1}^m P_i$. While Fenchel cuts are deep in a well-defined sense and are guaranteed to be face inducing, their convergence properties are not known, and this is the primary question motivating the results presented in the remaining sections.

It is important to recognize that the results presented in this paper are motivated by practical questions that were encountered when Algorithm Cut was used in a code for solving integer programs. When using Fenchel cuts, will the algorithm terminate finitely or must an ad hoc criterion be invoked for termination? Even if Fenchel cuts do lead to finite termination, will they take a long time to converge or do they have properties suggesting good convergence in practice? How do Fenchel cuts compare to other dual-based procedures for generating cutting planes? Finally, what parameters can be beneficially controlled in the procedure for generating Fenchel cuts? Empirical observations provided some insight into these questions, but these observations actually raised more questions than they answered, and this served to inspire a theoretical study of the convergence properties of Fenchel cuts.

2. Background and notation. In order to provide an unambiguous foundation for the work that follows, we present notation and a number of basic results that are used throughout the paper.

Given a polyhedron $P = \{x \in R^n : Ax \leq b\}$ a *face* of P is any nonempty set F that can be expressed as $F = \{x \in P : \lambda x = \lambda_0\}$ where the constraint $\lambda x \leq \lambda_0$ is *valid* for P; that is, all $x \in P$ satisfy $\lambda x \leq \lambda_0$. The hyperplane defined by any valid inequality whose intersection with P is F is said to *define* F, and if $F \neq P$ then F is a *proper* face. Given any description $Ax \leq b$ of P and any face F of P let $A^= x \leq b^=$ be the subset of constraints whose equality sets contain F and let $A^< x \leq b^<$ be the remaining constraints. A standard proposition relating the constraints $A^= x \leq b^=$ to F is the following.

PROPOSITION 2.1. *The smallest affine space containing F is a translation of the nullspace of $A^=$. In particular,* rank $A^=$ + dim $F = n$.

Given two vectors $\lambda, \gamma \in R^n$, we denote the angle between these vectors by $\angle(\lambda, \gamma)$. A *subgradient* of a convex function $f(x)$ at a point $\bar{x}$ is a vector $\bar{\lambda}$ defining a supporting hyperplane of $f(x)$ at $\bar{x}$; that is, a vector satisfying $f(x) \geq f(\bar{x}) + (x - \bar{x})\bar{\lambda}$.

Throughout this paper we let $\|x\|_a$ denote an arbitrary norm on R^n. For notational convenience, if $a = p$ is some positive real number we use $\|x\|_p$ to denote the L^p norm, $(\sum_{i=1}^n |x_i|^p)^{1/p}$. The following proposition states a number of properties of norms which will be explicitly referenced in the course of this paper.

PROPOSITION 2.2. *The following statements are true for any norms $\|x\|_a$ and $\|x\|_b$.*

(a) *There exist positive scalars m and M such that $m\|x\|_a \leq \|x\|_b \leq M\|x\|_a$.*

(b) *The vector $\bar{\lambda}$ is a subgradient of $\|x\|_a$ at $\bar{x}$ if and only if $\bar{\lambda}$ is a subgradient of $\|x\|_a$ at $\alpha\bar{x}$, where $\alpha > 0$ is any scalar.*

(c) *There exists a constant $C_{ab} > 0$ such that for any subgradient λ of $\|x\|_a$ at*

$x \neq 0$, $\|\lambda\|_b \geq C_{ab}$.

Proposition 2.2(a) is the well-known equivalence of norms in finite dimensions, while Proposition 2.2(b) can be verified from the defining properties of norms and Proposition 2.2(c) can be verified from Proposition 2.2(b) and the defining properties of norms.

Given an arbitrary norm $\|x\|_a : R^n \to R$, the dual norm $\|\lambda\|_a^* : R^n \to R$ is defined by

$$\begin{array}{ll} \max & \lambda x \\ \text{s.t.} & \|x\|_a \leq 1. \end{array}$$

It can be shown here that $\|x\|_a^{**} = \|x\|_a$, and it is useful to note that $\|\lambda\|_2^* = \|\lambda\|_2$, $\|\lambda\|_1^* = \|\lambda\|_\infty$, and by the observation just made $\|\lambda\|_\infty^* = \|\lambda\|_1$. For a polyhedron $P \subseteq R^n$ and a point $\hat{x} \in R^n$ the problem of finding a point $\bar{x} \in P$ closest to $\hat{x}$ as measured by the norm $\|x\|_a$ can be stated succinctly as

$$\text{(M)} \qquad \begin{array}{ll} \min & \|x - \hat{x}\|_a \\ \text{s.t.} & x \in P. \end{array}$$

Similarly, the problem of finding a hyperplane separating $\hat{x}$ and P can be formulated as

$$\text{(D)} \qquad \begin{array}{ll} \max & \lambda\hat{x} - f(\lambda) \\ \text{s.t.} & \lambda \in \Lambda, \end{array}$$

where Λ is a full-dimensional set containing the origin in its strict interior and

$$\begin{aligned} f(\lambda) = \max\ & \lambda x \\ \text{s.t. } & x \in P. \end{aligned}$$

As a notational matter, when the domain of (M) is denoted by P_i rather than P we will refer to (M) as (M_i) and to (D) as (D_i). The fact that (D) solves the separation problem can be seen by observing that if $\hat{x} \in P$ then (D) must have a nonpositive optimal value, while if there exists a separating hyperplane $\lambda x \leq \lambda_0$ with $\lambda\hat{x} > \lambda_0$ and $\lambda x \leq \lambda_0$ for all $x \in P$, then $\alpha\lambda$ is feasible for (D) for some sufficiently small $\alpha > 0$ and $(\alpha\lambda)\hat{x} - f(\alpha\lambda) = (\alpha\lambda)\hat{x} - \alpha f(\lambda) \geq (\alpha\lambda)\hat{x} - \alpha\lambda_0 = \alpha(\lambda\hat{x} - \lambda_0) > 0$. If $\bar{\lambda}$ is feasible for (D) and has a positive objective function value then the associated *Fenchel cut* is $\bar{\lambda}x \leq f(\bar{\lambda})$.

A domain of very special interest is $\Lambda = \{\lambda : \|\lambda\|_a^* \leq 1\}$, which in effect is the unit sphere in an arbitrary norm. The importance of this domain is that it establishes a dual relationship between the problems (M) and (D).

PROPOSITION 2.3. *When* $\Lambda = \{\lambda : \|\lambda\|_a^* \leq 1\}$ *the following statements are true of the problems* (M) *and* (D).

(a) *If* $\bar{x}$ *is feasible for* (M) *and* $\bar{\lambda}$ *is feasible for* (D) *then* $\|\bar{x} - \hat{x}\|_a \geq \bar{\lambda}\hat{x} - f(\bar{\lambda})$.

(b) *If* $\bar{x}$ *is optimal for* (M) *and* $\bar{\lambda}$ *is optimal for* (D) *then* $\|\bar{x} - \hat{x}\|_a = \bar{\lambda}\hat{x} - f(\bar{\lambda})$.

(c) *If* $\bar{x}$ *is optimal for* (M) *and* $\bar{\lambda}$ *is optimal for* (D) *then* $f(\bar{\lambda}) = \bar{\lambda}\bar{x}$ *and* $-\bar{\lambda}$ *is a subgradient of* $\|x - \hat{x}\|_a$ *at* $\bar{x}$.

A proof of Proposition 2.3 can be found in [15]. It is easily verified using Proposition 2.3 that when $\Lambda = \{\lambda : \|\lambda\|_a^* \leq 1\}$ the Fenchel cut $\bar{\lambda}x \leq f(\bar{\lambda})$ is as far as possible from $\hat{x}$ as measured by the norm $\|x\|_a$; that is, *the cut generated by solving* (D) *provides the maximum* $\|x\|_a$*-norm separation between* $\hat{x}$ *and* P. This property provides

motivation for studying *deepest norm Fenchel cuts* — Fenchel cuts associated with optimal solutions to (D) when Λ is defined by the unit sphere in an arbitrary norm.

While procedures for actually solving (D) are not of direct relevance to the present paper, it is worth mentioning that (D) can be solved to optimality using generalized programming if an oracle for optimizing a linear function on P is known, and simplex procedures can be used to solve (D) if an oracle for parametrically optimizing a linear function on P is known. Other techniques can be employed to find an approximate solution to (D), and this is often sufficient in practice since cutting planes are defined by values of λ for which the objective function in (D) is positive. For further details regarding the use of Fenchel cuts in practice, the reader is directed to [5] and [6].

If a polyhedron P is full-dimensional, bounded, and contains the origin in its strict interior then the 1-*polar* of P is the polyhedron

$$Q = \{\lambda : \ \lambda x^k \leq 1, \ \ x^k \in E(P)\},$$

where $E(P)$ is the set of extreme points of P. A well-known relation between P and Q that is not difficult to verify directly is the following.

PROPOSITION 2.4. *The constraint $\lambda^k x \leq 1$ is a facet of P if and only if $\lambda^k \in E(Q)$, and the constraint $\lambda x^k \leq 1$ is a facet of Q if and only if $x^k \in E(P)$.*

A proof of Proposition 2.4 can be found in [16].

3. The convergence of Algorithm Cut. The main result presented in this section is that when deepest norm Fenchel cuts are used in Algorithm Cut the sequence of x^t generated by the algorithm not only converge to a point in $\bigcap_{i=1}^m P_i$ but that the algorithm terminates finitely. The fundamental idea underlying the desired finiteness proof is that deepest norm Fenchel cuts cannot define a given face F of a polyhedron P_i more than a finite number of times. The specific finite number is a property of the face F and the norm under consideration. The following lemma proves the existence of a fundamental angle associated with any norm that in turn is used to demonstrate the existence of a uniform bound on the minimum angular separation between the gradients of any two constraints defining a face F of P_i.

LEMMA 3.1. *For any norm $\|\cdot\|_a$ there exists an angle $\theta_a < \pi/2$ such that for any point $\bar{x} \neq \hat{x}$ and any subgradient $\bar{\lambda}$ of the function $\|x - \hat{x}\|_a$ at $\bar{x}$,*

$$\angle(\bar{\lambda}, \bar{x} - \hat{x}) \leq \theta_a$$

Proof. By the definition of a subgradient,

$$\|x - \hat{x}\|_a \geq \|\bar{x} - \hat{x}\|_a + \bar{\lambda}(x - \bar{x}),$$

and since $\bar{x} \neq \hat{x}$ and $\|\bar{\lambda}\|_2 > 0$ by Proposition 2.2(c) we can write

$$\frac{\|x - \hat{x}\|_a}{\|\bar{\lambda}\|_2 \|\bar{x} - \hat{x}\|_2} \geq \frac{\|\bar{x} - \hat{x}\|_a}{\|\bar{\lambda}\|_2 \|\bar{x} - \hat{x}\|_2} + \frac{\bar{\lambda}(x - \bar{x})}{\|\bar{\lambda}\|_2 \|\bar{x} - \hat{x}\|_2}.$$

At $x = \hat{x}$ this reduces to

$$\frac{\bar{\lambda}(\bar{x} - \hat{x})}{\|\bar{\lambda}\|_2 \|\bar{x} - \hat{x}\|_2} \geq \frac{\|\bar{x} - \hat{x}\|_a}{\|\bar{x} - \hat{x}\|_2} \frac{1}{\|\bar{\lambda}\|_2}$$

or, alternatively stated,

$$\cos \angle(\bar{\lambda}, \bar{x} - \hat{x}) \geq \frac{\|\bar{x} - \hat{x}\|_a}{\|\bar{x} - \hat{x}\|_2} \frac{1}{\|\bar{\lambda}\|_2}.$$

With $\|\bar{x}-\hat{x}\|_a/\|\bar{x}-\hat{x}\|_2$ uniformly bounded away from zero by Proposition 2.2(a) and $\|\bar{\lambda}\|_2$ uniformly bounded away from zero by Proposition 2.2(c), the result follows. In particular, if we define

$$\eta_a = \min_{\bar{x}\neq\hat{x}} \frac{\|\bar{x}-\hat{x}\|_a}{\|\bar{x}-\hat{x}\|_2}\frac{1}{\|\bar{\lambda}\|_2},$$

then

$$\theta_a \leq \cos^{-1}\eta_a. \qquad \square$$

For future reference we let $\theta_a < \pi/2$ be defined as

$$\theta_a = \max\{\angle(\bar{\lambda}, \bar{x}-\hat{x}): \ \bar{x}\neq\hat{x}, \ \bar{\lambda} \text{ a subgradient of } \|x-\hat{x}\|_a \text{ at } \bar{x}\}.$$

With the aid of Lemma 3.1 we are now in a position to complete the following theorem.

THEOREM 3.2. *Suppose that at iteration t of Algorithm Cut, the Fenchel cut $\bar{\lambda}x \leq f(\bar{\lambda})$ is generated for P_i by the optimal solution to (D_i) with $\Lambda = \{\lambda : \|\lambda\|_a^* \leq 1\}$. Let F be the face of P_i defined by this cut and let $a^j x \leq b_j$, $j = 1,\ldots,K$ be the set of inequalities in the system $A^t x \leq b^t$ of Algorithm Cut that define F. Finally, let θ_a be as defined in Lemma 3.1. Then*

$$\angle(\bar{\lambda}, a^j) \geq \frac{\pi}{2} - \theta_a \quad \text{for all } j = 1,\ldots,K.$$

Proof. Let $\bar{x}$ be optimal for (M_i), let $k \in \{1,\ldots,K\}$ be some fixed index, and for notational convenience let $\bar{\theta} = \angle(\bar{\lambda}, a^k)$ and $\hat{\theta} = \angle(\bar{\lambda}, x^t - \bar{x})$. We note that since $-\bar{\lambda}$ is a subgradient of the function $\|x - x^t\|_a$ at $\bar{x}$ by Proposition 2.3(c) it follows by Lemma 3.1 that $\hat{\theta} = \angle(-\bar{\lambda}, \bar{x} - x^t) \leq \theta_a$. Also, $\bar{x} \in F$ by Proposition 2.3(c) so that $a^k\bar{x} = b_k$, and since x^t satisfies $a^k x^t \leq b_k$ by the operation of Algorithm Cut it follows that $a^k(x^t - \bar{x}) \leq 0$. We proceed to show that assuming $\bar{\theta} < \pi/2 - \theta_a$ leads to the contradiction $a^k(x^t - \bar{x}) > 0$.

Decomposing a^k into two components, one residing in the space spanned by $\bar{\lambda}$ and the other in the orthogonal complement of this space, we have

$$\frac{a^k}{\|a^k\|_2} = (\cos\bar{\theta})\frac{\bar{\lambda}}{\|\bar{\lambda}\|_2} + (\sin\bar{\theta})\frac{\gamma}{\|\gamma\|_2},$$

where

$$\gamma = \frac{a^k}{\|a^k\|_2} - (\cos\bar{\theta})\frac{\bar{\lambda}}{\|\bar{\lambda}\|_2}.$$

We can thus write

$$\frac{a^k}{\|a^k\|_2}\frac{(x^t-\bar{x})}{\|x^t-\bar{x}\|_2} = (\cos\bar{\theta})\frac{\bar{\lambda}}{\|\bar{\lambda}\|_2}\frac{(x^t-\bar{x})}{\|x^t-\bar{x}\|_2} + (\sin\bar{\theta})\frac{\gamma}{\|\gamma\|_2}\frac{(x^t-\bar{x})}{\|x^t-\bar{x}\|_2}$$

$$= (\cos\bar{\theta})(\cos\hat{\theta}) + (\sin\bar{\theta})\frac{\gamma}{\|\gamma\|_2}\frac{(x^t-\bar{x})}{\|x^t-\bar{x}\|_2} \tag{1}$$

Similarly, decomposing $x^t - \bar{x}$ into two components, one residing in the space spanned by $\bar{\lambda}$ and the other in the orthogonal complement of this space, we have

$$\frac{x^t-\bar{x}}{\|x^t-\bar{x}\|_2} = (\cos\hat{\theta})\frac{\bar{\lambda}}{\|\bar{\lambda}\|_2} + (\sin\hat{\theta})\frac{\omega}{\|\omega\|_2},$$

where

$$\omega = \frac{x^t - \bar{x}}{\|x^t - \bar{x}\|_2} - (\cos\hat{\theta})\frac{\bar{\lambda}}{\|\bar{\lambda}\|_2}.$$

Thus,

$$\begin{aligned}\frac{\gamma}{\|\gamma\|_2}\frac{(x^t - \bar{x})}{\|x^t - \bar{x}\|_2} &= \frac{\gamma}{\|\gamma\|_2}\left((\cos\hat{\theta})\frac{\bar{\lambda}}{\|\bar{\lambda}\|_2} + (\sin\hat{\theta})\frac{\omega}{\|\omega\|_2}\right)\\ &= \frac{\gamma}{\|\gamma\|_2}\frac{\omega}{\|\omega\|_2}(\sin\hat{\theta})\end{aligned}$$

so that (1) reduces to

$$\begin{aligned}\frac{a^k}{\|a^k\|_2}\frac{(x^t - \bar{x})}{\|x^t - \bar{x}\|_2} &= (\cos\bar{\theta})(\cos\hat{\theta}) + (\sin\bar{\theta})(\sin\hat{\theta})\frac{\gamma}{\|\gamma\|_2}\frac{\omega}{\|\omega\|_2}\\ &\geq (\cos\bar{\theta})(\cos\hat{\theta}) - (\sin\bar{\theta})(\sin\hat{\theta})\left|\frac{\gamma}{\|\gamma\|_2}\frac{\omega}{\|\omega\|_2}\right|\\ &\geq (\cos\bar{\theta})(\cos\hat{\theta}) - (\sin\bar{\theta})(\sin\hat{\theta})\\ &= \cos(\bar{\theta} + \hat{\theta}),\end{aligned} \tag{2}$$

where the final equality follows from a standard trigonometric identity.

If $\bar{\theta} < \pi/2 - \theta_a$ then since $\hat{\theta} \leq \theta_a$ it follows that $\bar{\theta} + \hat{\theta} < \pi/2$, implying $\cos(\bar{\theta} + \hat{\theta}) > 0$. Together with (2) this would imply

$$a^k(x^t - \bar{x}) \geq \|a^k\|_2\|x^t - \bar{x}\|_2\cos(\bar{\theta} + \hat{\theta}) > 0,$$

which is the desired contradiction. □

Theorem 3.2 guarantees that when deepest norm Fenchel cuts are used in Algorithm Cut then any two cuts defining the same face of P_i are separated by some minimum angle. The finite number of faces of a polyhedron leads immediately to the desired finiteness result.

COROLLARY 3.3. *When Fenchel cuts are used in Algorithm Cut with the domain* Λ *of* (D_i) *defined by the unit sphere of an arbitrary norm then the algorithm terminates after a finite number of iterations.*

4. The speed of convergence of Algorithm Cut. The proof of Theorem 3.2 provides considerable insight into deepest norm Fenchel cuts beyond finiteness. In this section we elaborate upon the measure of finiteness and how it is affected by the polyhedra P_i and the choice of norm.

We first elaborate upon the effect of the shape of the polyhedra P_i on the speed of convergence of Algorithm Cut, showing in the process that a measure of finiteness, implicit in Theorem 3.2, is tight. Let $C(F)$ denote the set of gradients of constraints that define the face F of P_i; that is,

$$C(F) = \{\lambda \in R^n : \lambda x \leq \lambda_0 \text{ defines } F \text{ for some } \lambda_0\}.$$

Theorem 3.2 demonstrates that F can be generated at most $T_a(F)$ times, where $T_a(F)$ is the maximum number of vectors that can be chosen from $C(F)$ such that these vectors all have angular separation of at least $\pi/2 - \theta_a$. For example, when F

is a facet of a full dimensional polyhedron P_i then $C(F)$ is a ray and $T_a(F) = 1$, and as a result F will be defined by at most one deepest norm Fenchel cut in the course of Algorithm Cut.

In general, the bound $T_a(F)$ is not tight in the sense that it is possible to construct examples of polyhedra P_i with faces F for which $T_a(F)$ cannot be achieved. Consider, for example, the integer program defined by the following data.

$$A = \begin{bmatrix} -1 & 1 \\ 1 & -1 \\ 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} 9/10 \\ 9/10 \\ 7/4 \\ 5/4 \\ 0 \\ 0 \end{bmatrix}, \quad c = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \hat{x} = \begin{bmatrix} 7/4 \\ 5/4 \end{bmatrix}.$$

Let $P_1 = \text{conv}\{(0,0),\ (1,1)\}$, which in this case coincides with the convex hull of all feasible integer points, let $F = \{(1,1)\}$, and consider the norm $\|\cdot\|_a = \|\cdot\|_\infty$. It is easily verified that $\theta_\infty = \pi/4$, and with $C(F) = \{\lambda : \lambda_1 + \lambda_2 > 0\}$ it follows that $T_\infty(F) = 4$. On the other hand, the deepest L^∞ cut (generated by solving (D_i) with $\Lambda = \{\lambda : \|\lambda\|_1 \leq 1\}$) is $x_1 \leq 1$. Appending this constraint to the constraints $Ax \leq b$ it is easily verified that any point $\hat{x} \in \{x \notin P_1 : Ax \leq b\}$ for which the deepest L^∞ cut defines F has gradient $\tilde{\lambda}$ satisfying $\angle(\tilde{\lambda}, (1,0)) \geq \pi/2 > \pi/4$. It is left to the reader to verify that this implies it is not possible for Algorithm Cut to generate F more than twice.

The inability to generate the face F more than twice in the previous example follows from the fact that there does not always exist an $\hat{x}$ satisfying $A\hat{x} \leq b$ such that the gradient of the deepest L^∞ cut achieves an angle sufficiently close to $\pi/2 - \theta_\infty$ relative to constraint gradients already defining F. In general, this small angle condition is necessary if F is to be defined $T_a(F)$ times. However, while a sufficiently small angle cannot always be achieved in a fixed coordinate system, if coordinate system rotation is allowed after each cutting plane is appended to the system $Ax \leq b$ then it can be shown that the bound $T_a(F)$ is tight. In the above example it is not difficult to construct four deepest L^∞ cuts if sequential coordinate rotations of $\pi/4$ are performed and $\hat{x}$ is restricted to satisfy $Ax \leq b$ but not required to maximize c. The bound $T_a(F)$ is not arbitrary but tight in a well-defined sense.

It should be stressed, however, that in real settings coordinate rotations will never occur so that a face F should rarely if ever be generated $T_a(F)$ times. General quantitative statements in this regard are difficult to make, but examples are very easily constructed that emphasize the difference between $T_a(F)$ and the actual number of times a face F of P_i can actually be defined in the course of Algorithm Cut.

While the size of $C(F)$ provides an important measure of how many times a face F may be generated in the course of Algorithm Cut, the angle θ_a associated with the norm $\|\cdot\|_a$ is also an extremely important measure. In particular, the smaller θ_a is the larger the minimum separation angle $\pi/2 - \theta_a$ must be between gradients defining a face F of P_i, and thus the fewer times F can be generated by Algorithm Cut. Practically, θ_a is perhaps a more important measure since it can be dictated by the choice of domain Λ in (D_i), whereas $C(F)$ is a property of the face F of P_i.

For any given norm $\|\cdot\|_a$ the value θ_a can be calculated on an ad hoc basis. One norm that proves to be of particular interest is the L^2 norm. Using the bound for θ_a provided at the end of Lemma 3.1, we have

$$\theta_2 \leq \cos^{-1} \eta_2,$$

where

$$\eta_2 = \min_{\bar{x} \neq \hat{x}} \frac{\|\bar{x} - \hat{x}\|_2}{\|\bar{x} - \hat{x}\|_2} \frac{1}{\|\bar{\lambda}\|_2}.$$

As $\|\bar{\lambda}\|_2 = 1$ for every subgradient of the L^2 norm at points $\bar{x} \neq \hat{x}$, it follows that $\theta_2 = 0$ and thus the minimum separation angle guaranteed by the L^2 norm is $\pi/2$, which is the best achievable. This bound makes it possible to state some very strong results regarding deepest L^2 cutting planes.

THEOREM 4.1. *Let F be a proper face of $P_i \subseteq R^n$ of dimension d. Suppose that at iteration t of Algorithm Cut the Fenchel cut generated for P_i by the optimal solution to* (D_i) *defines F and that $\Lambda = \{\lambda : \|\lambda\|_2 \leq 1\}$ (deepest L^2 cuts are generated). Then F can be defined by at most $n - d - 1$ constraints $a^j x \leq b_j$ in the set of constraints $A^t x \leq b^t$ (F will be defined by at most $n - d$ cuts in the course of Algorithm Cut).*

Proof. Since F is a proper face of P_i there exists a point $\tilde{x} \in P_i$ such that $\tilde{x} \notin F$. By the definition of $C(F)$, for every $a^k \in C(F)$ and its corresponding F-defining constraint $a^k x \leq b_k$ it follows that $a^k \tilde{x} < b_k$. Letting $\bar{x}$ be a point in F so that $a^k \bar{x} = b_k$, it follows that $C(F)$ is contained in the open half-space $\{x : (\bar{x} - \tilde{x})x > 0\}$. With F a face of dimension d, it is easily argued, using Proposition 2.1, that dim $C(F) = n - d$. As the maximum number of vectors with angular separation of at least $\pi/2$ in an open half-space of dimension $n - d$ is $n - d$, the proof is complete. □

Theorem 4.1 makes a very strong case for the use of deepest L^2 Fenchel cuts since they provide a remarkably strong measure of finiteness for Algorithm Cut, independent of the properties of the underlying polyhedra P_i. In some instances, the underlying polyhedra P_i have properties that allow even stronger results to be demonstrated for the L^2 norm.

DEFINITION 4.2. *Let F be a face of a polyhedron $P = \{x : Ax \leq b\}$ and let $i = 1, \ldots, K$ be the indices of the constraints whose equality sets contain F. We say P is* cone acute *if $a^i a^j \geq 0$ for any $i, j \in \{1, \ldots, K\}$ and any face F.*

Cone acuteness arises naturally in the study of the extremely important class of polyhedra associated with independence systems that include many fundamental combinatorial optimization problems. The following result was proved in Boyd [5], and demonstrates that deepest L^2 cuts together with well-shaped polyhedra can lead to very strong convergence for Algorithm Cut.

THEOREM 4.3. *Suppose that P_i is cone acute and let F be a face of P_i. Furthermore, suppose that at iteration t of Algorithm Cut the Fenchel cut generated for P_i by the optimal solution to (D_i) defines F and that $\Lambda = \{\lambda : \|\lambda\|_2 \leq 1\}$ (deepest L^2 cuts are generated). Then there exists no constraint $a^k x \leq b_k$ defining F in the set of constraints $A^t x \leq b^t$ (a face can be defined at most once in the course of Algorithm Cut). Furthermore, if a face F' of P_i is contained in F then there exists no constraint $a^k x \leq b_k$ defining F' in the set of constraints $A^t x \leq b^t$.*

One final note on deepest norm Fenchel cuts regards how the choice of a coordinate system for the space in which P_i resides can affect the cut that is generated by (D_i). While it can be shown that translations of the coordinate system do not change the generated cut, it is not difficult to construct examples for which the generated cut is affected by rotations of the coordinate system. While it is difficult to provide a useful quantitative measure of the extent to which coordinate rotation can change the cut, the potential effect is both limited and highly dependent upon the particular norm $\|\cdot\|_a$ under consideration. Coordinate rotation would have an unlimited effect on cutting plane generation if any cut in the set $\mathcal{F}$ of all supporting hyperplanes of P_i that

strictly separate $\hat{x}$ and P_i could be generated by an appropriate coordinate rotation. This would be a highly undesirable property since it would make cut generation completely dependent upon the choice of coordinate system. In general, coordinate rotation has only a limited effect on deepest norm Fenchel cuts (we return to discuss the potentiality of an unlimited effect for a different class of cutting planes in §6). In fact, since the L^2 norm is invariant under coordinate rotations and translations we have the following result.

PROPOSITION 4.4. *If* $\Lambda = \{\lambda : \|\lambda\|_2 \leq 1\}$ *(deepest* L^2 *cuts are generated) then the set of Fenchel cuts corresponding to optimal solutions of* (D) *remains invariant (relative to* P *and* $\hat{x}$*) under coordinate rotations and translations of the space in which* P *resides.*

This proposition again emphasizes the important role of the L^2 norm.

5. The dimension of faces defined by Fenchel cuts. The previous sections focused on the worst-case performance of deepest norm Fenchel cutting planes. In particular, it was demonstrated that the number of times a face F of a polyhedron P_i could be generated in the course of Algorithm Cut was related to the size of the set $C(F)$ of gradients of constraints defining F. One implication of this result is that there is a greater opportunity for deepest norm Fenchel cuts to generate faces of P_i of low dimension since higher dimensional faces of a polyhedron tend to have correspondingly smaller sets $C(F)$. However, while the size of $C(F)$ is a fundamental factor in determining the number of times an F-defining constraint *can* be generated, the dimension of F strongly affects the *likelihood* that it will be generated. As Algorithm Cut progresses, deepest norm Fenchel cuts have a natural tendency to define faces of the polyhedra P_i of higher and higher dimension, ultimately generating facets. The reason for this behavior comes from the fact that as the point x^t to be separated gets closer to these polyhedra as more cuts are added it becomes more probable for a high dimensional face to be generated. Formally, we have the following proposition.

PROPOSITION 5.1. *Let* P_i *be a full dimensional polytope in* R^n *and let* $\Omega_r = \{x : r = \min \|x - y\|_2 \text{ s.t. } y \in P_i\}$; *that is, the set of points that are a distance* $r > 0$ *from* P_i. *Furthermore, let* S_j *denote the event that the Fenchel cut generated by solving* (D_i) *with* $\Lambda = \{\lambda : \|\lambda\|_2 \leq 1\}$ *defines a face of* P_i *of dimension* j *and let* $P_r(S_j)$ *denote the probability of this event assuming* $\hat{x}$ *is chosen from a uniform distribution on* Ω_r. *Then*

$$\lim_{r \to 0} \frac{P_r(S_j)}{P_r(S_k)} = 0 \qquad j < k$$

and, in particular, the probability that the Fenchel cut is a facet tends to 1 *as* r *tends to* 0.

Letting $X(F)$ be the set of points $\hat{x} \in \Omega_r$ that gives rise to Fenchel cuts defining the face F, the proof of Proposition 5.1 follows from elementary results showing that the surface area of $X(F)$ is a function of $r^{(n-1)-j}$, where j is the dimension of F. Care should be taken in interpreting the practical implications of this proposition. While it is all but impossible to quantify, the iterates x^t generated in the course of Algorithm Cut tend to lie near the region of $\bigcap_{i=1}^m P_i$ where cx is optimized on this set, and this reflects upon the uniformity assumption of Proposition 5.1. In point of fact, the uniformity assumption is not so essential as certain relatively weak boundedness conditions on the sequence of probability measures P_r as r tends to zero. However, we do not dwell upon a more lengthy analysis than that provided here since it would not provide any further insight than that provided by Proposition 5.1.

6. 1-polar cuts. When a full-dimensional, bounded polyhedron P contains the origin in its strict interior, an alternative method for solving the separation problem for a point $\hat{x}$ is to solve the problem

$$\text{(N)} \qquad \begin{array}{ll} \max & \lambda\hat{x} - 1 \\ \text{s.t.} & \lambda x^k \leq 1 \quad \text{for all } x^k \in E(P), \end{array}$$

where $E(P)$ denotes the set of extreme points of P. Clearly, if $\bar{\lambda}$ is feasible for (N) and has a positive objective function value, then the constraint $\bar{\lambda}x \leq 1$ is valid for P and strictly separates $\hat{x}$ and P. With the assumption that the origin is in the strict interior of P, it is equally easy to see that if there exists a constraint strictly separating $\hat{x}$ and P then it can be written as $\bar{\lambda}x \leq 1$, implying $\bar{\lambda}$ is feasible for (N) and has a positive objective function value. We call a cutting plane $\bar{\lambda}x \leq 1$ generated by an optimal solution $\bar{\lambda}$ to (N) a 1-*polar* cut. One particular advantage of 1-polar cuts is that optimal extreme point solutions of (N) define facets of P so that finiteness of Algorithm Cut is not an issue when 1-polar cuts are used. Furthermore, although (N) is defined implicitly by a potentially exponential number of constraints, solution techniques for (N) and the Fenchel cutting plane problem (D) are very similar.

While 1-polar cuts have the advantage of defining facets of P, they also have inherent weaknesses that can best be elaborated upon by interpreting 1-polar cuts in the framework of Fenchel cuts. To see this relationship let Q be the polyhedron defined by the feasible region of (N). Clearly, the origin is contained in the strict interior of Q so that the domain $\Lambda = Q$ can be used in (D) to generate a cutting plane if it exists. Furthermore, if $\bar{\lambda}$ is on the surface of Q, that is, satisfies $\bar{\lambda}x^k = 1$ for some $x^k \in E(P)$, then $f(\bar{\lambda}) = 1$, where

$$f(\lambda) = \begin{array}{ll} \max & \lambda x \\ \text{s.t.} & x \in P. \end{array}$$

Since $v(\lambda) = \lambda\hat{x} - f(\lambda)$ satisfies $v(\alpha\lambda) = \alpha v(\lambda)$ for any scalar $\alpha \geq 0$, it follows that when Λ is convex all optimal solutions to (D) reside on the boundary of Λ or at the origin, and we have argued that solving (D) with $\Lambda = Q$ reduces to solving

$$\begin{array}{ll} \max & \lambda\hat{x} - 1 \\ \text{s.t.} & \lambda \in \Lambda, \end{array}$$

which is exactly the problem (N). Fenchel cuts and 1-polar cuts are equivalent in the sense that the set of cuts corresponding to optimal solutions to (D) with the domain $\Lambda = Q$ and the set of cuts corresponding to optimal solutions to (N) are identical.

The immediate observation with respect to the strength of 1-polar cuts is that since the domain Q is a polyhedron defined by the extreme points of P it does not generally define a norm and, as such, the "depth" of 1-polar cuts is quite unpredictable. This point is underlined by the fact that 1-polar cuts are extremely sensitive to the relative location of the origin within the polyhedron P. This sensitivity to coordinate translations is as bad as can be conceived and is formalized in the following theorem.

THEOREM 6.1. *Let P be a full-dimensional polyhedron and $\hat{x}$ a point not contained in P. Furthermore, let $\mathcal{F}$ denote the set of all facets of P such that the inequality defining $F \in \mathcal{F}$ strictly separates P from $\hat{x}$. Then by an appropriate translation of the coordinate system* any *$F \in \mathcal{F}$ can be made the optimal solution to* (N).

Proof. Assume for simplicity of exposition that P is initially situated so that the origin is contained in its strict interior. We begin by observing that if x^0 is contained

in the strict interior of P then solving

$$(\mathrm{N}(x^0)) \qquad \begin{array}{ll} \max & \lambda(\hat{x}+x^0)-1 \\ \text{s.t.} & \lambda(x^k+x^0)\leq 1 \quad \text{for all } x^k \in E(P) \end{array}$$

is equivalent to solving (N) with the coordinate system of the space in which P resides translated by x^0. For notational convenience we denote the translation of P by $P(x^0)$ and the feasible region of $\mathrm{N}(x^0)$ by $Q(x^0)$. Furthermore, if $\lambda^G x \leq 1$ defines a facet of P in the original coordinate system then this same constraint in the translated coordinate system is

$$\left(\frac{\lambda^G}{1-\lambda^G x^0}\right) x \leq 1,$$

where we recognize that $1-\lambda^G x^0 > 0$ follows from the fact that x^0 is contained in the strict interior of P.

Let F be a facet of P and let x^F be a point in the relative interior of F so that the only facet defining inequality $\lambda x \leq 1$ satisfied at equality by x^F is the inequality $\lambda^F x \leq 1$ defining F. The point x^F is not an acceptable choice for the virtual origin x^0 in $\mathrm{Q}(x^0)$ since it is not contained in the strict interior of P. However, for $\alpha > 0$ sufficiently small the point $x^F - \alpha\lambda^F$ is in the strict interior of P. We proceed to show that by choosing $\alpha > 0$ small enough

$$\frac{\lambda^F}{1-\lambda^F(x^F-\alpha\lambda^F)}$$

is the unique optimal solution to $\mathrm{N}(x^F-\alpha\lambda^F)$; that is, when the origin of the space in which P resides is translated by $x^F-\alpha\lambda^F$ the generated 1-polar cut defines the facet F. As the only property we assume about F is that $\lambda^F\hat{x} > 1$, the proof follows.

The problem we wish to consider, then, is

$$(\mathrm{N}(x^F-\alpha\lambda^F)) \qquad \begin{array}{ll} \max & \lambda(\hat{x}+(x^F-\alpha\lambda^F))-1 \\ \text{s.t.} & \lambda(x^k+(x^F-\alpha\lambda^F))\leq 1 \quad \text{for all } x^k \in E(P), \end{array}$$

where we restrict attention to the case where $\alpha > 0$ is sufficiently small so that $x^F-\alpha\lambda^F$ is in the strict interior of P. We observe that the extreme points of $Q(x^F-\alpha\lambda^F)$ are in one-to-one correspondence with the facets of $P(x^F-\alpha\lambda^F)$ by Proposition 2.4. Also, we observe that the facets of $P(x^F-\alpha\lambda^F)$ are in one-to-one correspondence with the facets of P, with $\lambda^G x \leq 1$ defining a facet G of P if and only if

$$\left(\frac{\lambda^G}{1-\lambda^G(x^F-\alpha\lambda^F)}\right) x \leq 1$$

defines the translated facet G. It follows that in order to determine a maximizing value of λ for $\mathrm{N}(x^F-\alpha\lambda^F)$ it is sufficient to consider only the finite set of points

$$\frac{\lambda^G}{1-\lambda^G(x^F-\alpha\lambda^F)}$$

corresponding to facets G of P.

Thus, consider an arbitrary extreme point $\lambda^G/(1-\lambda^G(x^F-\alpha\lambda^F))$ of $Q(x^F-\alpha\lambda^F)$ corresponding to a facet defining inequality of $\lambda^G x \leq 1$ of P. From the definition of the cosine we have that the objective function value of $\lambda^G/(1-\lambda^G(x^F-\alpha\lambda^F))$ satisfies

$$\left(\frac{\lambda^G}{1-\lambda^G(x^F-\alpha\lambda^F)}\right)(\hat{x}-(x^F-\alpha\lambda^F)) \tag{3}$$
$$=\left\|\frac{\lambda^G}{1-\lambda^G(x^F-\alpha\lambda^F)}\right\|_2 \|\hat{x}-(x^F-\alpha\lambda^F)\|_2 \cos\theta(\alpha),$$

where

$$\theta(\alpha)=\angle\left(\frac{\lambda^G}{1-\lambda^G(x^F-\alpha\lambda^F)},\ \hat{x}-(x^F-\alpha\lambda^F)\right). \tag{4}$$

It remains only to consider the right-hand side of (3) in the limit as $\alpha \to 0^+$. We immediately note that independent of G

$$\lim_{\alpha\to 0^+}\|\hat{x}-(x^F-\alpha\lambda^F)\|_2=\|\hat{x}-x^F\|_2>0$$

since $x^F \in P$ and $\hat{x} \notin P$.

When $G \neq F$ we have $\lambda^G x^F < 1$ since $x^F \in P$ was chosen so that it is not contained in any facet $G \neq F$. It follows that

$$\lim_{\alpha\to 0^+}\left\|\frac{\lambda^G}{1-\lambda^G(x^F-\alpha\lambda^F)}\right\|_2$$

is finite and thus the right-hand side of (3) approaches a finite value in the limit as $\alpha \to 0^+$.

When $G = F$ the value $\lambda^F x^F = 1$ by the choice of x^F so that

$$\lim_{\alpha\to 0^+}\left\|\frac{\lambda^F}{1-\lambda^F(x^F-\alpha\lambda^F)}\right\|_2=+\infty.$$

Furthermore, the limiting value of the angle $\theta(\alpha)$ reduces to

$$\lim_{\alpha\to 0^+}\angle\left(\frac{\lambda^F}{1-\lambda^F(x^F-\alpha\lambda^F)},\ \hat{x}-(x^F-\alpha\lambda^F)\right)=\lim_{\alpha\to 0^+}\angle\left(\frac{\lambda^F}{\alpha\lambda^F\lambda^F},\ \hat{x}-x^F\right)$$
$$=\angle\left(\lambda^F,\ \hat{x}-x^F\right).$$

Since $\lambda^F\hat{x} > 1$ by the choice of F and $\lambda^F x^F = 1$ by the choice of x^F it follows that $\lambda^F(\hat{x}-x^F) > 0$ or, equivalently, $\angle\left(\lambda^F,\ \hat{x}-x^F\right) < \pi/2$, and thus $\lim_{\alpha\to 0^+}\cos\theta(\alpha) > 0$. It follows that the right-hand side of (3) approaches $+\infty$ in the limit as $\alpha \to 0^+$, completing the proof. □

While a 1-polar cut is sometimes considered the "deepest facet" separating $\hat{x}$ and P since it maximizes $\lambda\hat{x}-1$, Theorem 6.1 makes it clear that this characterization of "deepest" is completely dependent upon where P resides in space. The proof of Theorem 6.1 demonstrates that facets close to the origin are favored. Alternatively, Fenchel cuts are biased toward cuts close to $\hat{x}$, where "close" is clearly defined by the dual of the norm used in generation of the Fenchel cut.

7. Conclusions. The purpose of this paper has been to examine the convergence properties of deepest norm Fenchel cutting planes in an effort to better understand the effectiveness of these cutting planes in solving mixed-integer programs. The main result was that finite convergence of Algorithm Cut could be guaranteed using Fenchel

cuts generated from solving (D_i) with the domain Λ defined by the unit sphere of an arbitrary norm, and more specific results were given for special norms. In §5 a case was made that deepest norm Fenchel cuts tend to define faces of increasingly higher dimension as Algorithm Cut progresses, and in §6 the strengths and weaknesses of 1-polar cuts were discussed.

The results of §§3 and 4 make a strong case for the use of deepest L^2 Fenchel cuts. Theorem 4.1 demonstrates that deepest L^2 Fenchel cuts provide an excellent bound on the number of times a particular face can be generated, and Theorem 4.3 shows that this bound is even stronger when the polyhedra P_i are nicely shaped. If a single class of Fenchel cutting planes had to be chosen based solely upon theoretical attributes it would be difficult to argue with the use of Fenchel cuts generated by solving (D_i) with domain $\Lambda = \{\lambda : \|\lambda\|_2 \leq 1\}$.

However, theoretical attributes alone are not sufficient to determine the choice of domain Λ since the difficulty of actually solving (D_i) is profoundly influenced by Λ. In particular, it is generally easier to solve (D_i) when Λ is defined by a collection of linear constraints rather than the nonlinear constraint required to generate a deepest L^2 cut. The need to solve (D_i) quickly is fundamental since many cutting planes will usually have to be generated in the course of Algorithm Cut.

The obvious choice for a linearly constrained domain is the L^∞ unit sphere, $\Lambda = \{\lambda : -1 \leq \lambda \leq 1\}$, which generates deepest L^1 cutting planes. Generating deepest L^∞ cuts by defining Λ as the L^1 unit sphere cannot be accomplished directly since the L^1 unit sphere has a number of facets exponential in the dimension n of the space in which it resides. However, the L^1 unit sphere can be expressed as a convex combination of its $2n$ extreme points after introducing $2n$ auxiliary variables so that generating deepest L^∞ cuts can be accomplished without too much difficulty. While the L^1 and L^∞ norms do not have the properties of the L^2 norm, they both provide reasonable approximations to the L^2 norm and thus represent a good practical alternative to the use of the L^2 unit sphere for Λ. Another advantage to using both the L^1 and L^∞ unit spheres for Λ comes from the fact that the convergence of Algorithm Cut is influenced by the specific shape of the polyhedra P_i and, while one domain Λ may lead to good convergence, in practice another may not. The L^1 and L^∞ norms complement each other nicely from a geometric perspective. Of course, one of the main results of this paper is that finite convergence is guaranteed under *any* norm.

While 1-polar cuts have the intuitively appealing property that they are facets of the underlying polyhedron from which they are generated, the results of §6 strongly call into question the ability to generate good 1-polar cuts. The results of §6 emphasize the fact that 1-polar cuts are not guaranteed to be deep in the sense of any norm and thus are prone to unpredicatable behavior. While the use of 1-polar cuts in certain applications should not be ruled out, care must be taken when they are used, and as a rule it appears unwise to use such cutting planes without adequate attention to the specific characteristics of the problem at hand.

Acknowledgments. The author wishes to thank Steve Cox for many helpful discussions in the course of preparing this paper, and to gratefully acknowledge the support of IMSL.

REFERENCES

[1] E. BALAS, *Facets of the knapsack polytope*, Math. Programming, 8 (1975), pp. 146–164.

[2] E. BALAS, S. CERIA, AND G. CORNUEJOLS, *A lift-and-project cutting plane algorithm for mixed* $0-1$ *programs*, Management Science Research Report 576, Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, PA, 1991.

[3] E. BALAS AND E. ZEMEL, *Facets of the knapsack problem from minimal covers*, SIAM J. Appl. Math., 34 (1978), pp. 119–148.

[4] ———, *Lifting and complementing yields all the facets of positive zero–one programming polytopes*, U. Derigs, ed., Math. Programming, Rio de Janiero, 1981, North-Holland, Amsterdam, New York, 1984, pp. 13–24.

[5] E. A. BOYD, *Fenchel cutting planes for integer programs*, Oper. Res., 42 (1994), pp. 53–64.

[6] ———, *Generating Fenchel cutting planes for knapsack polyhedra*, SIAM J. Optim., 3 (1993), pp. 734–750.

[7] ———, *Solving integer programs with enumeration cutting planes*, Working paper, Department of Industrial Engineering, Texas A&M University, College Station; Ann. Oper. Res., to appear.

[8] E. W. CHENEY AND A. A. GOLDSTEIN, *Newton's method for convex programming and Tchebycheff approximation*, Numer. Math., 1 (1959), pp. 253–268.

[9] H. CROWDER, E. L. JOHNSON, AND M. W. PADBERG, *Solving large-scale zero-one linear programming problems*, Oper. Res, 31 (1983), pp. 803–834.

[10] R. E. GOMORY, *Outline of an algorithm for integer solutions to linear programs*, Bull. Amer. Math. Soc., 64 (1958), pp. 275–278.

[11] M. GRÖTSCHEL AND O. HOLLAND, *Solution of large-scale symmetric travelling salesman problems*, Math. Programming, 51 (1991), pp. 141–202.

[12] M. GRÖTSCHEL, L. LOVÁSZ, AND A. SCHRIJVER, *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, New York, 1988.

[13] P. L. HAMMER, E. L. JOHNSON, AND U. N. PELED, *Facets of Regular* $0-1$ *Polytopes*, Math. Programming 8 (1975), pp. 179–206.

[14] J. E. KELLEY, *The cutting-plane method for solving convex programs*, J. Soc. Indust. Appl. Math., 8 (1960), pp. 703–712.

[15] D. G. LUENBERGER, *Optimization by Vector Space Methods*, Wiley and Sons, New York, 1969.

[16] G. L. NEMHAUSER AND L. A. WOLSEY, *Integer and Combinatorial Optimization*, Wiley and Sons, New York, 1988.

[17] M. PADBERG, *Covering, packing, and knapsack polytopes*, Ann. Discrete Math., 4 (1979), pp. 265–287.

[18] M. PADBERG AND G. RINALDI, *Optimization of a* 532*-city travelling salesman problem by branch-and-cut*, Oper. Res. Lett., 6 (1986), pp. 1–7.

[19] ———, *A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems*, SIAM Rev., 33 (1991), pp. 60–100.

[20] R. T. ROCKAFELLAR, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[21] J. F. SHAPIRO, *Mathematical Programming: Structures and Algorithms*, Wiley and Sons, New York, 1979.

[22] A. F. VEINOTT, *The supporting hyperplane method for unimodal programming*, Oper. Res., 15 (1967), pp. 147–152.

[23] L. A. WOLSEY, *Faces for a linear inequality in* $0-1$ *variables*, Math. Programming, 8 (1975), pp. 165–178.

SIAM J. OPTIMIZATION
Vol. 5, No. 2, pp. 436–453, May 1995

011

SUBDIFFERENTIAL CONVERGENCE IN STOCHASTIC PROGRAMS*

JOHN R. BIRGE† AND LIQUN QI‡

Abstract. In this paper, we discuss convergence behavior of subdifferentials in approximation schemes for stochastic programs. This information is useful for solving stochastic programs by nonlinear programming techniques. Wets [*Variational Inequalities and Complementarity Problems*, John Wiley, New York, 1980, pp. 375–404] showed that epiconvergence of closed convex functions implies the set convergence of the graph of the subdifferentials of these functions. This conclusion is not true in general by a counterexample of Higle and Sen [*Math. Oper. Res.*, 17 (1992), pp. 112–311]. We show that epiconvergence of closed convex functions implies set convergence of subdifferentials of these functions at points where the limit function is differentiable and apply this result to convex stochastic programs. We also show that similar results can be achieved in three other cases of expectational functionals: piecewise smooth integrands, continuous probability distributions, and loss functions. In the case of loss functions, we extend the existing results of Marti [*Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 31 (1975), pp. 203–233] to more general situations. Some basic methods using the approximate derivative information are also discussed.

Key words. approximation, subdifferential, stochastic programming

AMS subject classifications. 26B25, 90C15

1. Introduction.

In general, a *stochastic programming problem* can be formulated as [37]

$$\begin{aligned} &\text{minimize } E\{f_0(x,\xi)\} \\ &\text{subject to} E\{f_i(x,\xi)\} \le 0, \quad i = 1,\dots,s, \\ &\qquad\qquad E\{f_i(x,\xi)\} = 0, \quad i = s+1,\dots,m, \\ &\qquad\qquad x \in X \subseteq \Re^n, \end{aligned} \tag{1.1}$$

where

(i) ξ is a random vector with support $\Xi \subseteq \Re^N$, and a probability distribution function P on $\Re^N$,

(ii) $f_0 : \Re^n \times \Xi \to \Re \cup \{+\infty\}$,

(iii) $f_i : \Re^n \times \Xi \to \Re$, $i = 1,\dots,m$,

(iv) X is closed,

(v) for all $x \in X$, and $i = 0, 1, \dots, m$, the *expectational functional*

$$(Ef_i)(x) := E\{f_i(x,\xi)\} = \int_\Xi f_i(x,\xi)dP(\xi)$$

is finite. The *stochastic program with recourse* and the *stochastic program with chance constraints* are two special cases of this model [37]. The numerical solution for (1.1)

*Received by the editors April 15, 1991; accepted for publication (in revised form) July 14, 1992. This work is supported in part by the Australian Research Council.

†Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, Michigan 48109 (jrbirge@srvr5.engin.umich.edu). The work of this author was supported in part by Office of Naval Research Grant N0014-86-K-0628 and the National Science Foundation Grant EECS-885101.

‡School of Mathematics, The University of New South Wales, Sydney, New South Wales 2052, Australia (LQi@unsw.edu.au).

is not an easy task [11], [27]. One popular method is to use a discrete distribution to approximate P. In the case of the stochastic program with recourse, some approximations for the recourse function have also been proposed [5], [6], [16]. This results in approximation of $f_0(x, \xi)$. The resulting problem is a large-scale program, which can be solved by either decomposition methods [35] or perhaps variants of the interior point algorithm [2]. One may also propose methods to approximate $f_i(x, \xi)$, for $i = 1, ..., m$, and solve the resulting relatively tractable problem.

In the stochastic programming literature, most of the literature focuses on the convergence of the solution vector as a by-product of the epiconvergence of the approximating expectational functionals. Some recent papers on this topic include [5], [9], [12], [14], [15], [17], [18], and [31]. Can we also get some approximation information on the subdifferentials of the objective function and the constraints of (1.1)? This information will be useful for solving stochastic programming by nonlinear programming techniques that use subgradients [11], [20], [21], [24], [25]. In [32], Wang suggested to solve stochastic programs by an *approximate nonlinear programming method*, which uses the kth approximation function value and its derivative or subgradients at the kth step to solve the original problem by a nonlinear programming method. For this approach, the approximation information of subdifferentials is especially useful. Wang used the term *differential stability* to describe this information.

It seems that Marti [20] first addressed this problem. He considered a special class of expectational functionals: convex loss functions. This class is the most common expectational functional in the recourse problem. Wets [34] proved that epiconvergence of closed convex functions implies the set convergence of the graph of the subdifferentials of these functions. In §2, we show that epiconvergence of closed convex functions implies set convergence of subdifferentials of these functions at points where the limit function is differentiable and apply this result to convex stochastic programs. In particular, combining with recent epiconsistency result for convex stochastic programs of King and Wets [17], we establish general conditions for subdifferential convergence of approximation schemes for those problems in the sense of probability one. According to this result, a stochastic quasigradient-type method (see [10]) is suggested at the end of that section.

Recently, Higle and Sen [14] gave examples that epiconvergence does not imply subdifferential convergence in the nonconvex case. Their notion of subdifferential convergence is weaker than the one used here. Therefore, we cannot expect that nice subdifferential approximation behavior can be achieved as a bonus of epiconvergence in general. Since in practice we always use some approximation functions, it is thus important to ask in which cases nice subdifferential approximation behavior can be achieved. This behavior may be especially critical when subdifferential information is required by an algorithm or sensitivity analysis.

We show that nice approximation behavior of subdifferentials of expectational functionals can be achieved in three other cases, namely, piecewise smooth integrands, continuous distributions and loss functions. We study these three cases in §§3–5, respectively.

The significance of piecewise smooth integrands is clear. The continuous distribution approximation probably was first suggested by Wets [33]. In that paper, he suggested a piecewise linear distribution approximation. In [8], Dexter, Yu, and Ziemba suggested using the linear combination of lognormal univariate distribution approximations. Wets also discussed the possibility of continuous distribution approximations in [35] and [36]. Gassmann [13] discussed applications of normal distributions in stochastic programming.

For loss functions, we study more general cases than Marti [20] studied. We show that everywhere strictly differentiable results can be obtained in reasonable conditions. Thus, in these special cases, differentiable methods may be used instead of nondifferentiable methods that are otherwise necessary.

The main uses for our results are in the ability to employ general optimization techniques that do not require complete optimization for a single approximation and that allow differentiable techniques to be used in intermediate approximation iterations. We discuss possible applications in algorithms in §6. We give general methods and show how approximations may be used without requiring optimization for each approximation ν. Certainly, this section only gives general ideas for possible uses of subdifferential approximation results. For more sophisticated nonlinear programming approaches in stochastic programming, the readers may refer to Marti [21], Marti and Fuchs [22], and Nazareth and Wets [24], [25].

While we study the issue of subdifferential convergence in this paper, we do not wish to depreciate other approaches, such as the stochastic quasigradient method [10], [11] that depends upon sampling. Each method has its advantages in particular situations. Moreover, sometimes they can be employed together, as in the case discussed in §2.

2. Convex stochastic programs. Consider the *expectational functional* $E_f = E\{f(.,\xi))\}$, where ξ is a random vector with support $\Xi \subseteq \Re^N$ and f is an extended real-valued function on $\Re^n \times \Xi$. One has

$$E_f(x) = \int f(x,\xi)P(d\xi), \tag{2.1}$$

where P is a probability measure defined on $\Re^N$. It is usually difficult to calculate E_f and its derivative or subdifferential. A popular approach is to approximate (2.1) by

$$E_f^\nu(x) = \int f(x,\xi)P_\nu(d\xi), \tag{2.2}$$

where $\{P_\nu, \nu = 1,\ldots\}$ is a sequence of probability measures converging in distribution to the probability measure P.

In the following, use E_f^0 and P_0 instead of E_f and P for convenience. Denote the Lebesgue measure by m. For a closed convex set $C \subseteq \Re^n$, let Ψ_C^* be the support function of C, defined by $\Psi^*(h|C) = \sup\{< x,h >: x \in C\}$. A sequence of closed convex sets $\{C_\nu : \nu = 1,\ldots\}$ in $\Re^n$ is said to converge to C if for any $h \in \Re^n$,

$$\lim_{\nu\to+\infty} \Psi^*(h|C_\nu) = \Psi^*(h|C).$$

One may easily prove the following proposition.

PROPOSITION 2.1. *Suppose that C and C_ν, for $\nu = 1,\ldots$, are closed convex sets in $\Re^n$. The following two statements are equivalent:*

(a) *C_ν converges to C as $\nu \to +\infty$;*

(b) *a point $x \in C$ if and only if there are $x_\nu \in C_\nu$ such that $x_\nu \to x$.*

Suppose an extended real-valued function $g : \Re^n \to \Re \cup \{+\infty\}$ is locally Lipschitz at x. The *Clarke directional derivative* of g at x with respect to $h \in \Re^n$ is

$$g^\circ(x;h) := \limsup_{y\to x, t\downarrow 0} [g(y+th) - g(y)]/t.$$

The subdifferential of g at x in the sense of Clarke [7] is then

$$\partial g(x) := \{u \in \Re^n : < u, h > \leq g^\circ(x; h), \forall h \in \Re^n\}.$$

When g is convex, the above definition coincides with the definition in convex analysis

$$\partial g(x) := \{u \in \Re^n : \ g(x+h) \geq g(x) + \langle u, h\rangle, \ \forall h \in \Re^n\}.$$

The conditions of the following theorem are the same as the conditions of Theorem 2.8 of [5]. We only slightly strengthen its conclusions for our further use.

THEOREM 2.2. *Suppose that*

(i) $\{P_\nu, \nu = 1, \ldots\}$ *converges in distribution to* P;

(ii) $f(x, \cdot)$ *is continuous on* Ξ *for each* $x \in D$, *where*

$$D = \{x : E_f(x) < +\infty\} = \{x : f(x, \xi) < +\infty, a.s.\};$$

(iii) $f(\cdot, \xi)$ *is locally Lipschitz on* D *with Lipschitz constant independent of* ξ;

(iv) *for any* $x \in D$ *and* $\epsilon > 0$, *there exists a compact set* S_ϵ *and* ν_ϵ *such that for all* $\nu \geq \nu_\epsilon$,

$$\int_{\Xi \setminus S_\epsilon} |f(x, \xi)| P_\nu(d\xi) < \epsilon,$$

and with $V_x = \{\xi : f(x, \xi) = +\infty\}, P(V_x) > 0$ *if and only if* $P_\nu(V_x) > 0$ *for* $\nu = 0, 1, \ldots$. *Then*

(a) E_f^ν *epi- and pointwise converges to* E_f; *if* $x, x^\nu \in D$ *for* $\nu = 1, 2, \ldots$ *and* $x^\nu \to x$, *then*

$$\lim_{\nu \to \infty} E_f^\nu(x^\nu) = E_f(x); \tag{2.3}$$

(b) E_f^ν, *where* $\nu = 0, 1, \ldots$, *is locally Lipschitz on* D; *furthermore, for each* $x \in D$, $\{\partial E_f^\nu(x) : \nu = 0, 1, \ldots\}$ *is bounded*;

(c) *if* $x^\nu \in D$ *minimizes* E_f^ν *for each* ν *and* x *is a limiting point of* $\{x^\nu\}$, *then* x *minimizes* E_f.

Proof. The epi- and point convergence were established in [5]. Let $x, x^\nu \in D, x^\nu \to x$. Then

$$\begin{aligned}
&\lim_{\nu \to \infty} |E_f^\nu(x^\nu) - E_f^\nu(x)| \\
&\leq \lim_{\nu \to \infty} \int |f(x^\nu, \xi) - f(x, \xi)| P_\nu(d\xi) \\
&\leq \lim_{\nu \to \infty} \int L_x \|x^\nu - x\| P_\nu(d\xi) \\
&= \lim_{\nu \to \infty} L_x \|x^\nu - x\| \\
&= 0,
\end{aligned}$$

where L_x is the Lipschitz constant of $f(\cdot, \xi)$ near x, which is independent of ξ by (iii). By point convergence of E_f^ν at x,

$$\lim_{\nu \to \infty} E_f^\nu(x) = E_f(x).$$

Putting these two results together, we have (2.3). This proves (a).

For any $x \in D$, y and z close to x, $\nu = 0, 1, \ldots$,

$$\begin{aligned} &|E_f^\nu(y) - E_f^\nu(z)| \\ \leq &\int |f(y,\xi) - f(z,\xi)| P_\nu(d\xi) \\ \leq &\int L_x \|y - z\| P_\nu(d\xi) \\ = &L_x \|y - z\|. \end{aligned}$$

By [7], $\partial E_f^\nu(x)$ is a nonempty, compact convex set, for each ν; and the 2-norms of subgradients in these subdifferentials are bounded by L_x. This proves (b).

By (b), E_f^ν are lower semicontinuous functions. By (a), E_f^ν epiconverges to E_f. By Theorem 3.7 of [37], we get the conclusion of (c). This completes the proof. □

For each $x \in D$, will $\partial E_f^\nu(x)$ converge to $\partial E_f(x)$? By the Rademacher Theorem, E_f^ν is differentiable a.s. on D. Will $\nabla E_f^\nu(x)$ converge to $\nabla E_f(x)$ for each $x \in D \setminus D_1$, where $m(D_1) = 0$? These are the topics of this paper.

In the case of closed convexity, we may invoke Theorem 3 of Wets [34]. He proved that if $g, g^\nu : \Re^n \to \Re \cup \{+\infty\}, \nu = 1, 2, \ldots$, are closed convex functions and $\{g^\nu\}$ epiconverges to g, then the graphs of the subdifferentials of g^ν converge to the graph of the subdifferential of g, i.e., for any convergent sequence $\{(x^\nu, u^\nu) : u^\nu \in \partial g^\nu(x^\nu)\}$ with (x, u) as its limit, one has $u \in \partial g(x)$; for any (x, u) with $u \in \partial g(x)$, there exists at least one such sequence $\{(x^\nu, u^\nu) : u^\nu \in \partial g^\nu(x^\nu)\}$ converging to it.

However, in general it is not true that

$$\partial g(x) = \lim_{\nu \to \infty} \partial g^\nu(x) \tag{2.4}$$

even if $x \in \text{int}(\text{dom}(g))$. For example, let $n = 1, g(x) = |x|, g^\nu(x) = |x|$ if $|x| \geq \frac{1}{\nu}, g^\nu(x) = \frac{1}{\nu}$ if $|x| < \frac{1}{\nu}$, for $\nu = 1, 2, \ldots$. Then g and g^ν are closed convex functions, $\{g^\nu\}$ epiconverges to g. For $\nu = 1, 2, \ldots$, g^ν is differentiable at $x = 0$ with $\nabla g^\nu(0) = 0$. But $\partial g(0) = [-1, 1]$. However, if g is differentiable at x, (2.4) is true. We prove this fact here.

THEOREM 2.3. *Suppose that $g, g^\nu : \Re^n \to \Re \cup \{+\infty\}, \nu = 1, 2, \ldots$, are closed convex functions and $\{g^\nu\}$ epiconverges to g. Suppose further that g is differentiable at x. Then*

$$\nabla g(x) = \lim_{\nu \to \infty} \partial g^\nu(x). \tag{2.5}$$

Proof. By Theorem 3 of [34], any convergent subsequence of $\{u^\nu \in \partial g^\nu(x)\}$ converges to $\nabla g(x)$. What we need to prove is that there exists ν_x such that for any $\nu \geq \nu_x$, $\partial g^\nu(x)$ is nonempty, and $\{\partial g^\nu(x) : \nu \geq \nu_x\}$ is bounded. We now prove these. Since g is differentiable at x, $x \in \text{int}(\text{dom}(g))$. By Corollary 5 of [34], $g^\nu(x)$ converges to $g(x)$. Thus, for ν big enough, $x \in \text{dom}(g^\nu)$, i.e., $\partial g^\nu(x)$ is nonempty since g^ν is closed convex. For any $\epsilon > 0$, since g is continuous at x, there exists a $t > 0$ such that

$$g(x + te_i) \leq g(x) + \epsilon,$$

$$g(x - te_i) \leq g(x) + \epsilon,$$

for $i = 1, 2, \ldots, n$, where $e_i, i = 1, 2, \ldots, n$, are the unit vectors in $\Re^n$. Since g^ν epiconverges to g, by the properties of epiconvergence [34], [35], [37], there exist $y_i^\nu \to x + te_i$ and $z_i^\nu \to x - te_i$ such that

$$\limsup_{\nu \to \infty} g^\nu(y_i^\nu) \leq g(x + te_i) \leq g(x) + \epsilon$$

and

$$\limsup_{\nu \to \infty} g^\nu(z_i^\nu) \leq g(x - te_i) \leq g(x) + \epsilon.$$

Then, there exists ν_x such that for all $\nu \geq \nu_x$, $x \in \mathrm{dom}(g^\nu)$,

$$g^\nu(y_i^\nu) \leq g(x) + 2\epsilon, \tag{2.6}$$

$$g^\nu(z_i^\nu) \leq g(x) + 2\epsilon, \tag{2.7}$$

$$g^\nu(x) \geq g(x) - \epsilon, \tag{2.8}$$

$$\|y_i^\nu - (x + te_i)\| \leq \frac{t}{n}, \tag{2.9}$$

and

$$\|z_i^\nu - (x - te_i)\| \leq \frac{t}{n}, \tag{2.10}$$

where (2.8) holds because $g^\nu(x)$ converges to $g(x)$. By (2.9) and (2.10), for any y satisfying $\|y - x\| \leq \frac{t}{n}$,

$$y \in \mathrm{conv}\,\{y_1^\nu, \ldots, y_n^\nu, z_1^\nu, \ldots, z_n^\nu\}.$$

By (2.6), (2.7), and convexity of g^ν, for any y satisfying $\|y - x\| \leq \frac{t}{n}$,

$$g^\nu(y) \leq g(x) + 2\epsilon. \tag{2.11}$$

Now, for any $u \in \partial g^\nu(x)$ where $\nu \geq \nu_x$, if $u \neq 0$, let

$$y = x + \frac{tu}{n\|u\|}.$$

By convexity of g^ν,

$$g^\nu(y) \geq g^\nu(x) + < u, y - x > = g^\nu(x) + \frac{t}{n}\|u\|. \tag{2.12}$$

By (2.12), (2.11), and (2.8),

$$\|u\| \leq [g^\nu(y) - g^\nu(x)]/t \leq \frac{3n\epsilon}{t}.$$

This proves that $\{\partial g^\nu(x) : \nu \geq \nu_x\}$ is bounded. Proof of this theorem is complete. □

We now apply Theorem 2.3 to expectational functionals.

COROLLARY 2.4. *In* (2.1) *and* (2.2), *suppose that* $f(\cdot,\xi)$ *is closed convex for each* $\xi \in \Xi$ *and that* E_f^ν *epiconverges to* E_f. *Then for* $D = \mathrm{dom}(E_f)$,

(d) *there is a Lebesgue zero-measure set* $D_1 \subseteq D$ *such that* E_f *is differentiable on* $D \setminus D_1$, *but not differentiable on* D_1, *and for each* $x \in D \setminus D_1$

$$\lim_{\nu\to\infty} \partial E_f^\nu(x) = \nabla E_f(x);$$

(e) *for each* $x \in D$,

$$\partial E_f(x) = \{\lim_{\nu\to\infty} u^\nu : u^\nu \in \partial E_f^\nu(x^\nu), x^\nu \to x\}.$$

Proof. By closed convexity of $f(\cdot,\xi)$, E_f^ν are also closed convex for all ν. Now (d) follows Theorem 2.3 and the differentiability property of convex functions, and (e) follows Theorem 3 of [34]. □

One can construct other results by combining Corollary 2.4 with a particular approximation, E_f^ν, to E_f. For example, combine Corollary 2.4 with Theorem 2.2.

COROLLARY 2.5. *In the setting of Theorem* 2.2, *if* $f(\cdot,\xi)$ *is convex for each* $\xi \in \Xi$, *then* (d) *and* (e) *hold.*

Proof. By Theorem 2.2, E_f^ν epiconverges to E_f. By convexity of $f(\cdot,\xi)$, E_f^ν are also convex for all ν. By (b), E_f^ν are lower semicontinuous, thus closed convex. Now the conclusions follow Corollary 2.4. □

One may also combine Corollary 2.4 with some recent results of epiconvergence, such as the results of Lepp [18]. Perhaps an interesting combination is with the results of King and Wets [17]. Let P_ν be an empirical measure derived from an independent series of random observations $\{\xi_1,\ldots,\xi_\nu\}$ each with common distribution P. Then for all x,

$$E_f^\nu(x) = \frac{1}{\nu}\sum_{i=1}^{\nu} f(x,\xi_i). \tag{2.13}$$

Let $(\Xi,\mathcal{A},P)$ be a probability space complete with respect to P. A closed-valued multifunction G mapping Ξ to $\Re^n$ is called *measurable* if for all closed subsets $C \subseteq \Re^n$, one has

$$G^{-1}(C) := \{\xi \in \Xi : G(\xi) \cap C \neq \emptyset\} \in \mathcal{A}.$$

In the following, "with probability one" refers to the sampling probability measure on $\{\xi_1,\ldots,\xi_\nu,\ldots\}$ that is consistent with P (see [17] for details). Applying Theorem 2.3 of [17] and Corollary 2.4 of this paper, we have the following corollary.

COROLLARY 2.6. *Suppose for each* $\xi \in \Xi$, $f(\cdot,\xi)$ *is closed convex and the epigraphical multifunction* $\xi \mapsto$ *epi* $f(\cdot,\xi)$ *is measurable. Let* E_f^ν *be calculated by* (2.13). *If there exist one point* $\bar{x} \in dom(E_f)$ *and a measurable selection* $\bar{u}(\xi) \in \partial f(\bar{x},\xi)$ *with* $\int \|\bar{u}(\xi)\| P(d\xi)$ *finite, then the conclusions of Corollary* 2.4 *hold with probability one.*

King and Wets [17] applied their results to the *stochastic program with fixed recourse*

$$\begin{aligned} &\text{minimize } cx + \int Q(x,\xi)P(d\xi) \\ &\text{subject to } Ax = b, \\ &\qquad x \geq 0, \end{aligned} \tag{2.14}$$

where $x \in \Re^n$ and

$$Q(x,\xi) = \inf\{q(\xi)y : Wy = T(\xi)x - \zeta(\xi), y \in \Re^m_+\}. \tag{2.15}$$

It is a fixed recourse problem since W is deterministic. Combining their Theorem 3.1 with our Corollary 2.4, we have the following corollary.

COROLLARY 2.7. *Suppose that the stochastic program* (2.14) *has fixed recourse* (2.15) *and that for all* i, j, k, *the random variables* $q_i\zeta_j$ *and* q_iT_{jk} *have finite first moments. If there exists a feasible point* $\bar{x}$ *of* (2.14) *with the objective function of* (2.14) *finite, then the conclusions of Corollary* 2.4 *hold with probability one for*

$$f(x,\xi) = cx + Q(x,\xi) + \delta(x),$$

where $\delta(x) = 0$ *if* $Ax = b, x \geq 0$, $\delta(x) = +\infty$ *otherwise.*

By Theorem 3.1 of [17], one may solve the approximation problem

$$\begin{aligned} &\text{minimize } cx + \frac{1}{\nu}\sum_{i=1}^{\nu} Q(x,\xi_i) \\ &\text{subject to } Ax = b, \\ &\qquad x \geq 0, \end{aligned} \tag{2.16}$$

instead of solving (2.14). If the solution of (2.16) converges as ν tends to infinity, then the limiting point is a solution of (2.14). Alternatively, by Corollary 2.7, one may directly solve (2.14) with a nonlinear programming method and use

$$cx + \frac{1}{\nu}\sum_{i=1}^{\nu} Q(x,\xi_i)$$

and

$$c + \frac{1}{\nu}\sum_{i=1}^{\nu} \partial_x Q(x,\xi_i)$$

as approximate objective function values and subdifferentials of (2.14) with $\nu = \nu(k)$ at the kth step. Notice that $u \in \partial_x Q(x,\xi_i)$ if and only if u is an optimal dual solution of (2.15) with $\xi = \xi_i$. Certainly, this approach needs further investigation for its convergence and practical performance. Section 6 gives some basic convergence properties for methods of this type based simply on subgradient information.

3. Piecewise smooth integrands. In the nonconvex case, if f is a "piecewise smooth" function in x, then we may obtain similar results to those in §2. We say a mapping $(x,\xi) \mapsto f(x,\xi) : \Re^n \times \Xi \to \Re \cup \{+\infty\}$ is a piecewise smooth function in x if $\Re^n \times \Xi$ can be partitioned into countable pieces of convex polyhedra such that for any (x,ξ) in the interior of a piece, $\nabla_x f(x,\xi)$ exists and is continuous in ξ,

$$m(K(\xi)) = 0,$$

for each $\xi \in \Xi$, where $K(\xi) = \{x : (x,\xi)$ is on the boundary of a piece of $f\}$. We assume that any polyhedral set in $\Re^n \times \Re^N$ is measurable under the product measure $m \times P_\nu$ for $\nu = 0, 1, \ldots$.

THEOREM 3.1. *In the setting of Theorem* 2.2, *we only assume* (i) *and* (iii) *hold. If f is a piecewise smooth function in x as defined above, then the conclusion* (b) *holds and*

(f) *there is a Lebesgue zero-measure set $D_1 \subseteq D$ such that E_f^ν, for $\nu = 0, 1, 2, \ldots$ are differentiable on $D \setminus D_1$ and for each $x \in D_2 = D \setminus D_1$*

$$\lim_{\nu \to \infty} \nabla E_f^\nu(x) = \nabla E_f(x). \tag{3.1}$$

Proof. The conclusion (b) holds since in the proof of Theorem 2.2 (b) only (iii) is invoked. Let J be the set of all the boundary (facet) points of polyhedron pieces of f. Then J is measurable under the product measure $m \times P_\nu$, for $\nu = 0, 1, \ldots$.

Let $J(x) = \{\xi \in \Xi : (x, \xi) \in J\}$ for each $x \in D$. Let $D^\nu = \{x \in D : J(x)$ has positive P_ν measure$\}$. By the Fubini Theorem, $m(D^\nu) = 0$. Let

$$D_1 = \cup_{\nu=0,1,\ldots} D^\nu$$

and $D_2 = D \setminus D_1$. Then the Lebesgue measure of D_1 is zero. We now prove (f) is true for the sets D_1 and D_2.

Let $x \in D_2$. By the property of D_2, the set $J(x)$ has zero measure in P_ν and m, $\nabla_x f(x, \xi)$ exists and is continuous in ξ for $\xi \in \Xi \setminus J(x)$. Let $h \in \Re^n$. Denote $f_\xi(x) = f(x, \xi)$. By the dominated convergence theorem, we have the existence of $(E_f^\nu)'(x; h)$ and the equality

$$(E_f^\nu)'(x; h) = \int f_\xi'(x; h) P_\nu(d\xi).$$

But

$$f_\xi'(x; h) = \langle \nabla_x f(x, \xi), h \rangle,$$

for $\xi \in \Xi \setminus J(x)$. The existence of $\nabla E_f^\nu(x)$ follows.

Let $\epsilon > 0$ be given. Let $h \in \Re^n$. By (b) and the property of D_2, $f_\xi'(x; h)$ is bounded by L_x and continuous in $\Xi \setminus J(x)$, where L_x is the Lipschitz constant in the proof of Theorem 2.2 (b). Therefore, since $f_\xi'(x; h)$ is a bounded, P-a.e. continuous function, from the theory of convergence in distribution (see, for example, Theorem 12.1.A of Loève [19]),

$$\lim_{\nu \to \infty} (E_f^\nu)'(x; h) = (E_f)'(x; h).$$

Since this is true for all $h \in \Re^n$, (3.1) holds. This completes the proof. □

4. Continuous distributions. A similar extension to Theorem 2.2 is possible if we restrict the probability measures instead of the function f. We say that a function $p : \Re^N \to \Re$ is piecewise continuous if $\Re^N$ can be partitioned into countable pieces of convex polyhedra such that p is continuous in the interior of each piece. We say a probability measure P on $\Re^N$ is *continuous* if there is a piecewise continuous function $p : \Re^N \to \Re_+$ such that

$$P(d\xi) = p(\xi) d\xi.$$

One may anticipate that many practical probability measures are continuous. Piecewise linear and piecewise constant probability measures may be regarded as examples of approximating measures P_ν. In the following, we investigate the case that P and P_ν are continuous.

In the proofs of the following theorem and Theorem 5.2, we use an alternative notion of subdifferential to establish differentiability results. We let $\partial^\diamond$ be the symbol for the generalized subdifferential in the sense of Michel and Penot [23]. Again, suppose an extended real-valued function $g : \Re^n \to \Re \cup \{+\infty\}$ is locally Lipschitz at x. The *Michel–Penot directional derivative* of g at x with respect to $h \in \Re^n$ is

$$g^\diamond(x;h) := \sup_{k\in\Re^n} \{\limsup_{t\downarrow 0}[g(x+tk+th) - g(x+tk)]/t\}.$$

The Michel–Penot subdifferential of g at x is then

$$\partial^\diamond g(x) := \{u \in \Re^n : \ \langle u, h\rangle \ \leq g^\diamond(x;h) \forall h \in \Re^n\}.$$

An advantages of the Michel–Penot subdifferential is that $\partial^\diamond g(x)$ is singleton if and only if g is differentiable at x. Thus, g is differentiable at x if and only if

$$g^\diamond(x;e_i) + g^\diamond(x;-e_i) = 0,$$

for $i = 1, 2, \ldots, n$, where e_i is the ith unit vector. This fact is used in the proof of the following theorem.

THEOREM 4.1. *In the setting of Theorem* 2.2, *we only assume* (iii) *holds. If* $\{P_\nu, \nu = 0, 1, \ldots\}$ *are continuous probability measures with*

$$P_\nu(d\xi) = p_\nu(\xi)d\xi,$$

where $p_\nu : \Xi \to \Re_+$, $\nu = 0, 1, 2, \ldots$ *are piecewise continuous functions,*

(v)

$$\lim_{\nu\to\infty} \int |p_\nu(\xi) - p(\xi)|d\xi = 0;$$

(vi) *the map* $(x, \xi) \mapsto f(x, \xi)$ *is Lebesgue measurable in* $D \times \Re^N$; *then conclusions* (f) *and* (g) *hold*:

(f) *the same as in Theorem* 3.1;

(g) *for any* $x \in$ int (D),

$$\lim_{\nu\to+\infty} \partial E_f^\nu(x) = \partial E_f(x). \tag{4.1}$$

Proof. (f) By (iii), $f(., \xi)$ is locally Lipschitz for each $\xi \in \Xi$. By the Rademacher Theorem, $f(., \xi)$ is differentiable almost everywhere in D for each $\xi \in \Xi$. Let

$$\bar{J} = \{(x, \xi) \in D \times \Xi : \nabla_x f(x, \xi) \text{ exists}\}.$$

We will show that $\bar{J}$ is Lebesgue measurable.

Let e_i be the ith unit vector in $\Re^n$. Denote $f(x, \xi)$ by $f_\xi(x)$. Consider

$$f_\xi^\diamond(x;e_i) = \sup_{k\in\Re^n} \{\limsup_{t\downarrow 0}[f(x+te_i+tk,\xi) - f(x+tk,\xi)]/t\}, \tag{4.2}$$

the Michel–Penot directional derivative of f_ξ at x with respect to e_i. Since $f(., \xi)$ is continuous, we may let $t \downarrow 0$ and k only take rational values in (4.2). By (vi), $(x, \xi) \mapsto f(x+te_i+tk, \xi)$ and $(x, \xi) \mapsto f(x+tk, \xi)$ are Lebesgue measurable. Therefore,

$f_\xi^\diamond(x;e_i)$, as the "countable sup limsup" of Lebesgue measurable functions of (x,ξ), is Lebesgue measurable. Similarly $f_\xi^\diamond(x;-e_i)$ is also Lebesgue measurable. So the set

$$\bar{J}_i = \{(x,\xi) \in D \times (\Re^N \setminus V) : f_\xi^\diamond(x;e_i) + f_\xi^\diamond(x;-e_i) = 0\}$$

is also Lebesgue measurable. However, by the discussion on the Michel–Penot subdifferential before this theorem, $\bar{J} = \cap_{i=1}^n \bar{J}_i$. Thus, $\bar{J}$ is also Lebesgue measurable.

Let $J(x) = \{\xi \in \Xi : (x,\xi) \notin \bar{J}\}$ for each $x \in D$. Let $D_1 = \{x \in D : m(J(x)) > 0\}$. By the Fubini Theorem, $m(D_1) = 0$. Let $D_2 = D \setminus D_1$. We now prove (f) is true for the sets D_1 and D_2.

First we show that the derivative $\nabla E_f^\nu(x)$ exists for $x \in D_2$. Let $x \in D_2$. Then $\nabla_x f(x,\xi)$ exists a.s. Let $h \in \Re^n$. Similarly, we may show that the map $\xi \mapsto f_\xi'(x;h)$ is Lebesgue measurable and bounded. By the dominated convergence theorem, we have the existence of $(E_f^\nu)'(x;h)$ and the equality

$$(E_f^\nu)'(x;h) = \int f_\xi'(x;h) p_\nu(\xi) d\xi = \int f_\xi'(x;h) P_\nu(d\xi). \tag{4.3}$$

But

$$f_\xi'(x;h) = \langle \nabla_x f(x,\xi), h\rangle,$$

for $\xi \notin J(x)$. The existence of $\nabla E_f^\nu(x)$ follows. By (4.3),

$$\begin{aligned} |(E_f^\nu)'(x;h) - E_f'(x;h)| &\le \int |f_\xi'(x;h)| \cdot |p_\nu(\xi) - p(\xi)| d\xi \\ &\le L_x \|h\| \int |p_\nu(\xi) - p(\xi)| d\xi, \end{aligned} \tag{4.4}$$

where L_x is the Lipschitz constant of $f(\cdot,\xi)$ near x, which is independent of ξ by (iii). By (4.4) and (v), (3.1) holds. This proves (f).

(g) Let $x \in$ int (D). Notice that (b) still holds since it only relies on (iii). By (b), $\partial E_f^\nu(x)$ are nonempty, compact, and convex sets for $\nu = 0,1,\ldots$. Let h be an arbitrary vector in $\Re^n$ and ϵ be a small positive number. Let $U(x)$ be a very small neighborhood of x. For any y in $U(x) \cap D_2$, by (4.4), $(E_f^\nu)'(y;h)$ tends to $E_f'(y;h)$ uniformly. Then there is a $\nu(\epsilon)$ such that for any $\nu \ge \nu(\epsilon)$ and $y \in U(x) \cap D_2$,

$$|(E_f^\nu)'(y;h) - E_f'(y;h)| \le \epsilon. \tag{4.5}$$

By 2.5.1 of [7],

$$\partial E_f^\nu(x) = \operatorname{conv}\{\lim \nabla E_f^\nu(y) : y \to x, y \in D_2\}, \tag{4.6}$$

for $\nu = 0,1,\ldots$. Then there is a sequence $y_j \to x, y_j \in U(x) \cap D_2$ such that

$$\Psi^*(h|\partial E_f(x)) = \lim_{j\to\infty} E_f'(y_j;h). \tag{4.7}$$

By (4.6),

$$\limsup_{j\to\infty} (E_f^\nu)'(y_j;h) \le \Psi^*(h|\partial E_f^\nu(x)), \tag{4.8}$$

for $\nu = 1, \ldots$. By (4.5), (4.7), and (4.8), for any $\nu \geq \nu(\epsilon)$,

$$\Psi^*(h|\partial E_f(x)) \leq \Psi^*(h|\partial E_f^\nu(x)) + \epsilon. \tag{4.9}$$

Similarly, one may show that for any $\nu \geq \nu(\epsilon)$,

$$\Psi^*(h|\partial E_f^\nu(x)) \leq \Psi^*(h|\partial E_f(x)) + \epsilon. \tag{4.10}$$

By (4.9) and (4.10),

$$\lim_{\nu \to +\infty} \Psi^*(h|\partial E_f^\nu(x)) = \Psi^*(h|\partial E_f(x)).$$

Then (4.1) follows. This completes the proof of this theorem. □

Notice that (f) does not imply (g) in general. See the example prior to Theorem 2.3.

5. Loss functions. We now discuss a special case of (2.1), in which

$$f(x, \xi) = u(T(\xi)x - \zeta(\xi)),$$

where $u : \Re^m \to \Re \cup \{+\infty\}$ is a loss function, $\zeta(\xi) \in \Re^m$ and $T(\xi) \in \Re^{m \times n}$. Consider

$$E_u(x) = \int u(T(\xi)x - \zeta(\xi))P(d\xi), \tag{5.1}$$

where P is a probability measure on $\Re^N$. The expectational functional (5.1) arises in stochastic program with recourse. See [5], [6]. It also appears in other applications such as error optimization and optimal design. See [22]. In stochastic linear programming with recourse, u is piecewise linear and convex. In more general problems, however, u may not even be convex. This situation occurs, for example, when u is a loss if inventory, $T(\xi)x$, is less than demand, $\zeta(\xi)$. This penalty will generally become flat for very low values of $T(\xi)x - \zeta(\xi)$, voiding convexity. Nevertheless, it is useful to characterize the convergence to critical points as we do here.

Again, approximate (5.1) by

$$E_u^\nu(x) = \int u(T(\xi)x - \zeta(\xi))P_\nu(d\xi),$$

and denote E_u by E_u^0 sometimes. In particular, we consider continuous approximations as in Theorem 4.1. In the following, we consider conditions under which the approximating expectation functionals have Clarke subdifferentials that correspond with the gradient when it exists. This occurs when a locally Lipschitz function f is *strictly differentiable* (see [7]) at x, i.e., there exists $\nabla f(x)$, such that

$$\lim_{x' \to x, t \downarrow 0} \frac{f(x' + tv) - f(x')}{t} = \nabla f(x)^T v,$$

for any v. In this case, we require an open domain and limit D to be an open set within the domain of every approximating expectation functional, E_u^ν.

THEOREM 5.1. *Suppose that*

(i) *for any $x \in D$, where D is an open set in $\Re^n$, $E_u^\nu(x) < +\infty$, for $\nu = 0, 1, \ldots$;*

(ii) *u is locally Lipschitz in its support Ω, and strictly differentiable in $\Omega_1 \subseteq \Omega$ such that $\Omega_2 = \Omega \setminus \Omega_1$ is a zero Lebesgue measure set;*

(iii) *for any* $\xi \in \Re^N$, *there is a constant* L *such that* $\|T(\xi)\| \leq L$;

(iv) *for any* $x \in D$, *the map* $\xi \mapsto T(\xi)x - \zeta(\xi)$ *maps a set in* $\Re^N$ *to a Lebesgue zero measure set in* $\Re^m$ *only if this set has Lebesgue measure zero in* $\Re^N$;

(v) $\{P_\nu, \nu = 0, 1, \ldots\}$ *are continuous probability measures with*

$$P_\nu(d\xi) = p_\nu(\xi)d\xi,$$

where $p_\nu : \Xi \to \Re_+$, $\nu = 0, 1, 2, \ldots$ *are piecewise continuous functions*;

(vi) *there is a positive number* δ *and a function* $\eta : \Omega \to \Re$, *such that*

$$\eta(w) \geq \sup\{\|d\| : d \in \partial u(\tilde{w}),\ \|\tilde{w} - w\| \leq \delta\},$$

and for any $x \in D$, $\eta(T(.)x.- \zeta(.))p_\nu(.)$ *is Lebesgue integrable, for* $\nu = 0, 1, \ldots$;

(vii) *assumption* (iv) *of Theorem* 2.2 *holds*;

(viii) *either* u *is convex or piecewise smooth, or Theorem* 4.1 (v) *holds. Then*

(a) E_u^ν *are strictly differentiable in* D *for* $\nu = 0, 1, \ldots$;

(b) *for any* $x \in D$ *and* $x^\nu \to x$,

$$\lim_{\nu \to +\infty} E_u^\nu(x^\nu) = E_u(x);$$

(c) *if* $x^\nu \in D$ *minimizes* E_u^ν *for each* ν *and* x *is a limiting point of* $\{x^\nu\}$, *then* x *minimizes* E_u;

(d) *for any* $x \in D$,

$$\lim_{\nu \to +\infty} \nabla E_u^\nu(x) = \nabla E_u(x). \tag{5.2}$$

Proof. By (ii), (iii), (v), and (vii), the conditions of Theorem 2.2 hold. We have conclusions (b) and (c). It suffices now to prove (a) and (d). Suppose $x \in D$. Let

$$U = \{\bar{x} \in D : \|\bar{x} - x\| < \delta/L\}$$

and $g_\nu(\xi) = L\eta(T(\xi)x - \zeta(\xi))p_\nu(\xi)$. Then for $\bar{x}, \hat{x} \in U$, by (ii) and 2.3.7 of [7],

$$u(T(\xi)\bar{x} - \zeta(\xi)) - u(T(\xi)\hat{x} - \zeta(\xi)) \in \langle \partial u(\tilde{w}), T(\xi)(\bar{x} - \hat{x})\rangle,$$

where $\tilde{w}$ is a point in $[T(\xi)\bar{x} - \zeta(\xi), T(\xi)\hat{x} - \zeta(\xi)]$. Let $w = T(\xi)x - \zeta(\xi)$. Then,

$$\|T(\xi)\bar{x} - \zeta(\xi) - w\| \leq \|T(\xi)(\hat{x} - x)\| \leq \delta,$$

$$\|T(\xi)\hat{x} - \zeta(\xi) - w\| \leq \|T(\xi)(\hat{x} - x)\| \leq \delta.$$

Thus, $\|\tilde{w} - w\| \leq \delta$. By (iii), (v), and (vi),

$$|u(T(\xi)\bar{x} - \zeta(\xi))p_\nu(\xi) - u(T(\xi)\hat{x} - \zeta(\xi))p_\nu(\xi)| \leq g_\nu(\xi)\|\bar{x} - \hat{x}\|,$$

where g_ν is Lebesgue integrable on $\Re^N$. However,

$$E_u^\nu(x) = \int u(T(\xi)x - \zeta(\xi))p_\nu(\xi)d\xi.$$

Hence, by 2.7.2 of [7], for $x \in D$, $\nu = 0, 1, \ldots$, E_u^ν is locally Lipschitz and

$$\partial E_u^\nu(x) \subseteq \int \partial u(T(\xi)x - \zeta(\xi))T(\xi)p_\nu(\xi)d\xi. \tag{5.3}$$

But the right-hand side of (5.3) is a singleton by (ii), (iii), and (iv). Thus $\partial E_u^\nu(x)$ is also a singleton and equality holds for (5.3). By 2.2.4 of [7], E_u^ν is strictly differentiable at x for $\nu = 0, 1, \ldots$. This proves (a).

Now, by Corollary 2.5, Theorems 3.1, and 4.1, (5.2) follows (viii). This completes the proof of this theorem. □

We note that Marti [20] establishes similar results for u continuous, positively homogeneous, and subadditive. We extend his results to this more general loss function. We now justify the conditions of Theorem 5.1.

If T is deterministic, then conditions (iii) and (iv) of this theorem hold naturally. Condition (ii) is equivalent to saying that u is locally Lipschitz and that the Clarke subdifferential of u is single-valued almost everywhere. Such functions are called *primal functions* in [28]–[30]. Convex functions, concave functions, differences of convex functions, regular functions, and semismooth functions are all examples of primal functions. All primal functions defined on an open set form a linear space, i.e., the class of primal functions is closed in addition and scalar multiplication.

As said before, in stochastic linear programming with recourse, u is a convex piecewise linear function. Thus, conditions (ii) and (viii) hold. Furthermore, condition (vi) also holds as long as the p_ν are integrable for $\nu = 0, 1, \ldots$.

For condition (v), as we said before, the piecewise linear distribution approximation suggested by Wets [33] and the linear combination of lognormal univariate distribution suggested by Dexter, Yu, and Ziemba [8] are good examples of continuous approximations. In [4], we discuss this topic more.

THEOREM 5.2. *If we delete the almost everywhere strict differentiability requirement on u in Theorem* 5.1, *then all the conclusions still hold except conclusion* (a) *is changed to*

(a) *E_u^ν are differentiable in D for $\nu = 0, 1, \ldots$.*

Proof. We use the Michel–Penot subdifferential instead of the Clarke subdifferential in the proof of Theorem 5.1. Instead of invoking 2.7.2 of [7], we invoke Theorem 5.2 of [3] now. This implies that the left-hand side of (5.3), now the Michel–Penot subdifferential of E_u^ν at x, is a singleton. By the properties of the Michel–Penot subdifferential [3], E_u^ν is differentiable at $x \in D$. This completes the proof. □

Another approach to approximate (2.1) is by using

$$E_{f^\nu}(x) = \int f^\nu(x, \xi)P(d\xi),$$

where $\{f^\nu, \nu = 1, \ldots\}$ is a sequence of extended real-valued functions converging pointwise to f. The convergence of E_{f^ν} to E_f was also discussed in [5] in the context of epiconvergence. See Theorem 2.7 of [5]. One may also derive results similar to Corollary 2.5, Theorems 3.1, 4.1, 5.1, and 5.2 by considering these four cases.

6. Application in algorithms. The major motivation in the development of continuous approximation schemes is for uses in algorithms. In this section, we demonstrate that some basic algorithms based on gradient and subgradient information can incorporate continuous approximations into convergent procedures. We first consider methods based on the result in Theorem 4.1. The benefit of the continuous distribution approximation is that subgradient convergence means the algorithm need not find

a solution for each ν. In this case, we only suppose that subgradients are available. The major result is that we obtain convergence to a value below some goal or else demonstrate that no value below the goal exists.

The algorithm is a direct modification of the subgradient method given by Polyak [26]. It can also be strengthened to allow for other step sizes as in Allen et al. [1]. We state the algorithm as follows. We assume that f is a convex function of x. The objective is to minimize $E_f(x)$ over $x \in D$.

SUBGRADIENT ALGORITHM.

Step 0. Suppose sequences $\epsilon_\nu \to 0$, $\gamma_k \to 0$ such that $\sum_{k=0}^{\infty} \gamma_k = +\infty$, a goal value G, a maximum iteration count per approximation of $I_{\max}$, and an initial point x^0. Let $k = 0, i = 0$, and $\nu = 0$.

Step 1. Pick $\zeta \in \partial E_f^\nu(x^k)$. If $\|\zeta\| < \epsilon_\nu$, let $x^k = x^{k+1}$ and go to Step 2. Otherwise, let

$$x^{k+1} = x^k - \gamma_k(\zeta/\|\zeta\|). \tag{6.1}$$

If $E_f^\nu(x^{k+1}) < G$, go to Step 2. Otherwise, go to Step 3.

Step 2. Let $\nu = \nu + 1$, $i = 0$, and $k = k + 1$. Go to Step 1.

Step 3. Let $i = i + 1$. If $i > I_{\max}$, go to Step 2. Otherwise, $k = k + 1$, go to Step 1.

We note that alternative choices for the step length parameter (here $\gamma_k/\|\zeta\|$) are possible but we will show convergence just for this case. The basic idea of the algorithm is to follow the subgradient algorithm steps unless a small subgradient is found, an approximate function value is less than the goal value, or neither has been found within $I_{\max}$ iterations with a single approximation. In these cases, the approximation is refined. From Theorem 4.1, we have conditions for the subdifferential sets to converge. We need an additional condition about the boundedness of the sequence of iterates.

(vii) The iterates x^k of the subgradient algorithm all belong to D and the diameter of D is a finite value, $M = \sup_{y,z \in D}\{\|y - z\|\}$.

This last condition can be guaranteed by adding a barrier or penalty function near the boundary of D. We assume it here for simplicity. Together the conditions lead to convergence in this algorithm as in the following theorem.

THEOREM 6.1. *Suppose the conditions of Theorem* 4.1 *and* (vii) *above, then the subgradient algorithm given above produces a sequence* $\{x^k\}$ *such that* $E_f(\bar{x}) \le G$ *for some limit point* $\bar{x}$ *of* $\{x^k\}$ *or* $E_f(x) \ge G$ *for all* $x \in D$.

Proof. The proof follows the same form as in a proof of the subgradient method's convergence. Suppose that there exists x^* such that $E_f(x^*) < G$ but $E_f(\bar{x}) > G$ for any limit point $\bar{x}$ of $\{x^k\}$. Since D is bounded, there exists $\bar{K}$ such that for all $k > \bar{K}$, $E_f(x^k) > G$. Otherwise, there exists $\{x^{k_i}\}$ with $E_f(x^{k_i}) \le G$ and some $\bar{x} = \lim_i x^{k_i}$ with $E_f(\bar{x}) \le G$. By the continuity of E_f, for all sufficiently small ϵ, there exists some δ such that for all $\|x - x^*\| < \delta$, $E_f(x) \le G - \epsilon M$. Suppose K is such that for all $k \ge K$, there exists $\eta^\nu \in \partial E_f^\nu(x^k)$ (where ν is the corresponding approximation index used by the algorithm at iteration k) with $\|\eta^\nu - \eta\| < \epsilon M$ and any $\eta \in \partial E_f(x^k)$. This is always possible for any $\epsilon > 0$ by Theorem 4.1 and noting that the algorithm always increases ν if a value below G, a small subgradient, or neither condition has been found in $I_{\max}$ iterations.

Next, we suppose that $y = x^* + \delta(\eta^\nu/\|\eta^\nu\|)$. From this for $k > \max\{\bar{K}, K\}$, we

obtain

$$
\begin{aligned}
E_f(y) &\geq E_f(x^k) + \langle \eta, y - x^k \rangle \\
&\geq G + \langle \eta - \eta^\nu, y - x^k \rangle + \langle \eta^\nu, x^* - x^k \rangle + \langle \eta^\nu, y - x^* \rangle \\
&\geq G - \|\eta - \eta^\nu\| \|y - x^k\| + \langle \eta^\nu, x^* - x^k \rangle + \delta \|\eta^\nu\| \\
&\geq G - \epsilon M + \langle \eta^\nu, x^* - x^k \rangle + \delta \|\eta^\nu\|,
\end{aligned} \tag{6.2}
$$

which implies that $\langle \eta^\nu, x^* - x^k \rangle / \|\eta^\nu\| \leq \delta$. But, we also have that $\|x^{k+1} - x^*\|^2 = \|x^k - x^*\|^2 - 2\gamma_k \langle \eta^\nu / \|\eta^\nu\|, x^k - x^* \rangle + \gamma_k^2$. Hence,

$$
\gamma^k (2\delta - \gamma^k) \leq -\|x^{k+1} - x^*\|^2 + \|x^k - x^*\|^2, \tag{6.3}
$$

but summing yields infinity on the left-hand side of (6.3) and $\|x^k - x^*\|^2$ on the right, a contradiction. The result follows. □

This result shows that subgradient information may be sufficient to produce an algorithm that achieves an optimal value. The method is similar to stochastic quasigradient algorithms that use a sampled quasigradient, $\bar{\xi}^k$, with an assumed error that vanishes asymptotically, in place of the approximate subgradient, η^ν, used in the proof above. In the stochastic quasigradient case [10], a result similar to the conclusion of (6.2) holds in *expectation* and leads to convergence with probability one. We could also apply our results from Theorem 4.1 using samples to obtain similar results. Our aim is, however, more toward approximations in which bounds can be determined. For example, if ν such that $\|\eta^\nu - \eta\| < \epsilon M$ and δ can be determined (as we explore in [4]), then we can use (6.3) to find stopping conditions for determining whether a value lower than G exists. In general stochastic quasigradient methods, this inequality only holds in expectation. Confidence intervals based on second order information are required for stopping criteria.

If continuous distribution approximations are used, we may obtain the results of Theorem 5.1 and convergence of derivatives. This result allows the use of general derivative-based methods. It is contained in the following general method. Now, in the context of Theorem 5.1, we suppose that E_u has a finite minimum and that the solution set is compact.

GRADIENT METHOD.

Step 0. Suppose a sequence $\epsilon^\nu \to 0$ and an initial point $x^0 \in D$.

Step 1. Follow a convergent descent algorithm for unconstrained minimization of a differentiable convex function that uses $\nabla E_u^\nu(x^k)$ to generate a new point x^{k+1} (i.e., an algorithm, such as steepest descent, that generates a sequence of points with decreasing function values that either terminates with an optimal solution or has all limit points as optimal solutions to the unconstrained minimization problem). Go to Step 2.

Step 2. If $\nabla E_u^\nu(x^{k+1}) \leq \epsilon^\nu$, let $\nu = \nu + 1$, $k = k + 1$ and go to Step 1. Otherwise, let $k = k + 1$ and go to Step 1.

THEOREM 6.2. *Suppose the conditions of Theorem* 5.1, *then the gradient method given above produces a sequence* x^k *such that all limit points* x^* *of* $\{x^k\}$ *minimize* E_u.

Proof. When ν is not updated, the method just follows whatever algorithm has been employed in Step 1. By the differentiability assumption, the algorithm will generate some $x^{k(\nu)}$ such that $\|\nabla E_u^\nu(x^{k(\nu)})\| < \epsilon^\nu$. Thus, the algorithm continues to refine the approximation. At each $x^{k(\nu)}$, the bound on the norm of the gradient decreases so we have that there exists some $\mathcal{N}_1$ such that for all $\nu > \mathcal{N}_1$,

$\|\nabla E_u^\nu(x^{k(\nu)})\| \leq \epsilon/2$. From Theorem 5.1, for any ϵ, there exists $\mathcal{N}_2$ such that for all $\nu \geq \mathcal{N}_2$, $\|\nabla E_u^\nu(x^{k(\nu)}) - \nabla E_u(x^{k(\nu)})\| \leq \epsilon/2$. Hence, for any $\nu \geq \max\{\mathcal{N}_1, \mathcal{N}_2\}$, $\|\nabla E_u(x^{k(\nu)})\| \leq \epsilon$. Thus, any limit point x^* of $\{x^{k(\nu)}\}$ minimizes E_u. Moreover, for any k, $k(\nu) \leq k \leq k(\nu+1)$, $E_u^\nu(x^{k(\nu)}) \geq E_u^\nu(x^k)$. Thus, we also have $E_u(x^{k(\nu)}) + \delta_k \geq E_u(x^k)$ for some $\delta_k \to 0$. Hence, $E_u(x^k) \to \min E_u(x)$, proving the result. □

These results are given to show that continuous approximation schemes have the advantage of allowing optimization with methods that require derivatives and that the resulting algorithms may converge under suitable assumptions. Specific forms of the approximations and computational results with these procedures will be reported in another paper [4]. We note that similar results are also reported in Wang [32] for several algorithms. His procedure relies on the epiconvergence of E_f^ν to E_f and requires descent when using subgradients. Since this may not occur at some approximation, refinement may be necessary before optimality at that approximation and before any steps are possible. Our procedure allows progress at each approximation either through differentiability or through the subgradient method.

REFERENCES

[1] E. Allen, R. Helgason, J. Kennington, and B. Shetty, *A generalization of Polyak's convergence result for subgradient optimization*, Math. Programming, 37 (1987), pp. 309–317.

[2] J. R. Birge and L. Qi, *Computing block-angular Karmarkar projections with applications to stochastic programming*, Management Sci., 34 (1988), pp. 1472–1479.

[3] ———, *Semiregularity and generalized subdifferentials with applications to optimization*, Math. Oper. Res., 18 (1993), pp. 982–1005.

[4] ———, *Continuous approximation schemes for solving stochastic programs*, Ann. Oper. Res., to appear.

[5] J. R. Birge and R. J.-B. Wets, *Designing approximation schemes for stochastic optimization problems, in particular, for stochastic programs with recourse*, Math. Programming Stud., 27 (1986), pp. 54–102.

[6] ———, *Sublinear upper bounds for stochastic programs with recourse*, Math. Programming, 43 (1989), pp. 131–149.

[7] F. H. Clarke, *Optimization and Nonsmooth Analysis*, Wiley, New York, 1983.

[8] A. S. Dexter, J. N. W. Yu, and W. T. Ziemba, *Portfolio selection in a lognormal market when the inversot has a power utility function : computationals results*, in Stochastic Programming, M.A.H. Dempster, ed., Academic Press, New York, 1980, pp. 507–523.

[9] J. Dupačová and R. J.-B. Wets, *Asymptotic behavior of statistical estimators and of optimal solutions of stochastic optimization problems*, Ann. Statist., 16 (1988), pp. 1517–1549.

[10] Y. Ermoliev, *Stochastic quasigradient methods and their applications to systems optimization*, Stochastics, 9 (1983), pp. 1–36.

[11] Y. Ermoliev and R. J.-B. Wets, *Numerical Techniques in Stochastic Programming*, Springer-Verlag, Berlin, 1988.

[12] K. Frauendorfer, *Solving* SLP *recourse problem with arbitrary multivariate distributions—the dependent case*, Math. Oper. Res., 13 (1988), pp. 377–394.

[13] H. Gassmann, *Conditional probability and conditional expectation of a random variable*, in Numerical Techniques in Stochastic Programming, Y. Ermoliev and R. Wets, eds., Springer-Verlag, Berlin, 1988, pp. 237–254.

[14] J. L. Higle and S. Sen, *On the convergence of algorithms, with applications for stochastic and nondifferentiable optimization*, Math. Oper. Res., 17 (1992), pp. 112–311.

[15] P. Kall, *On approximations and stability in stochastic programming*, in Parametric Optimization and Related Topics, eds. by J. Guddat, et al., Akademie-Verlag, Berlin, 1987, pp. 387–407.

[16] P. Kall, A. Ruszczyński, and K. Frauendorfer, *Approximation technicales in stochastic programming*, in Numerical Techniques in Stochastic Programming, Y. Ermoliev and R. Wets, eds., Springer-Verlag, Berlin, 1988, pp. 33–64.

[17] A. J. KING AND R. J.-B. WETS, *Epi-consistency of convex stochastic programs*, Stochastics and Stochastics Reports, 34 (1991), pp. 83–92.

[18] R. LEPP, *Approximations to stochastic programs with complete recourse*, SIAM J. Control Optim., 28 (1990), pp. 382–394.

[19] M. LOÈVE, *Probability Theory* I, Springer-Verlag, New York, Berlin, Heidelberg, 1977.

[20] K. MARTI, *Approximationen von Entscheidungsproblemen mit linearer Ergebnisfunktion und positiv homogener, subadditiver Verlusfunktion*, Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete, 31 (1975), pp. 203–233.

[21] ———, *Computation of descent directions and efficient points in stochastic optimization problem with invariant distributions*, ZAMM, 65 (1985), pp. 132–156.

[22] K. MARTI AND E. FUCHS, *Computation of descent directions and efficient points in stochastic optimization problems without using derivatives*, Math. Programming Stud., 28 (1986), pp. 132–156.

[23] P. MICHEL AND J.-P. PENOT, *Calcul sous-différentiel pour des fonctions lipschitziennes et non lipschitziennes*, C. R. Acad. Sci. Paris., 298 (1984), pp. 269–272.

[24] J. L. NAZARETH AND R. J.-B. WETS, *Algorithms for stochastic programs: the case of nonstochastic tenders*, Math. Programming Stud., 28 (1986), pp. 1–28.

[25] ———, *Nonlinear programming techniques applied to stochastic programs with recourse*, in Numerical Techniques in Stochastic Programming, Y. Ermoliev and R. Wets, eds., Springer-Verlag, Berlin, 1988, pp. 95–119.

[26] B. T. POLYAK, *Subgradient methods: A survey of Soviet research*, in Nonsmooth Optimization, C. Lemarechal and R. Mifflin, eds., Pergamon Press, Oxford, 1978, pp. 5–29.

[27] A. PRÉKOPA AND R. J.-B. WETS, *Stochastic Programming 84, Mathematical Programming Study*, 27 & 28 (1986).

[28] L. QI, *The maximal normal operator space and integration of subdifferentials of nonconvex functions*, Nonlinear Anal., Theory Methods Appl., 13 (1989), pp. 1003–1011.

[29] ———, *Semismoothness and decomposition of maximal normal operators*, J. Math. Anal. Appl., 146 (1990), pp. 271–279.

[30] ———, *Quasidifferentials and maximal normal operators*, Math. Programming, 49 (1991), pp. 263–271.

[31] S. M. ROBINSON AND R. J-B. WETS, *Stability in two-stage stochastic programming*, SIAM J. Control Optim., 25 (1987), pp. 1409–1416.

[32] J. WANG, *Approximate nonlinear programming algorithms for solving stochastic programs with recourse*, Ann. Oper. Res., 30 (1991), pp. 371–384.

[33] R. J.-B. WETS, *Solving stochastic programs with simple recourse*, II, in Proc. 1975 Conference on Information Sciences and Systems, John Hopkins University, Baltimore, 1975.

[34] ———, *Convergence of convex functions, variational inequalities and convex optimization problems*, in Variational Inequalities and Complementarity Problems, R.W. Cottle, F. Giannessi, and J.-L. Lions, eds., John Wiley, New York, 1980, pp. 375–404.

[35] ———, *Stochastic programming: Solution techniques and approximation schemes*, in Mathematical Programming: The State of the Art–Bonn 1982, A. Bachem, M. Grötschel, and B. Korte, eds., Springer-Verlag, Berlin, 1983, pp. 566–603.

[36] ———, *Stochastic programs with simple recourse*, Stochastics, 10 (1983), pp. 219–242.

[37] ———, *Stochastic programming*, in: Handbooks in Operations Research and Management Science, Volume 1: Optimization, G.L. Nemhauser, A.H.G. Rinnooy Kan, and M.J. Todd, eds., North-Holland, Amsterdam, 1989, pp. 573–630.

SIAM J. OPTIMIZATION
Vol. 5, No. 2, pp. 454–466, May 1995

012

PROXIMAL DECOMPOSITION ON THE GRAPH OF A MAXIMAL MONOTONE OPERATOR*

PHILIPPE MAHEY†, SAID OUALIBOUCH‡, AND PHAM DINH TAO§

Abstract. We present an algorithm to solve: Find $(x, y) \in A \times A^{\perp}$ such that $y \in Tx$, where A is a subspace and T is a maximal monotone operator. The algorithm is based on the proximal decomposition on the graph of a monotone operator and we show how to recover Spingarn's decomposition method. We give a proof of convergence that does not use the concept of partial inverse and show how to choose a scaling factor to accelerate the convergence in the strongly monotone case. Numerical results performed on quadratic problems confirm the robust behaviour of the algorithm.

Key words. proximal point algorithm, partial inverse, convex programming

AMS subject classification. 90C25

1. Introduction.

We consider in this paper the following constrained inclusion problem: let X be a finite dimensional vector space and A a subspace of X. Let us denote by B the orthogonal subspace of A, i.e., $B = A^{\perp}$. Let T be a maximal monotone operator on X and denote its graph by $\mathrm{Gr}(T)$, i.e., $\mathrm{Gr}(T) = \{(x, y) \in X \times X | y \in Tx\}$. Then, the problem is to find $x \in A$ and $y \in B$ such that $y \in Tx$, which can be written:

$$\text{(P) Find } (x, y) \in X \times X \text{ such that } (x, y) \in A \times B \cap \mathrm{Gr}(T).$$

A typical situation, which is easily shown to give the form (P), is the problem of minimizing a convex lower semicontinuous function on a subspace. The particular applications we have in mind are the decomposition methods for separable convex programming. They have recently gained some new interest with the possibility of implementing them on massively parallel architectures to solve very large problems such as the ones that appear in network optimization or stochastic programming (see [1]). There are many different ways to transform a separable convex program in the form (P), but the general idea is to represent the coupling between the subsystems by a subspace of the product space of the copies of the primal and dual variables.

We are aiming here at the application of the Proximal Point Algorithm (PPA) (cf. [11]) to problem (P). In 1983, Spingarn [12] proposed a generalization of PPA to solve (P) that was based on the notion of the Partial Inverse operator. If we denote by x_A the orthogonal projection of x on a subspace A, the graph of the partial inverse operator T_A is given by

$$\mathrm{Gr}(T_A) = \{(x_A + y_B, y_A + x_B) \mid y \in Tx\}.$$

Applying the PPA to this operator leads to the Partial Inverse Method (PIM) which we summarize here.

ALGORITHM 1 (PIM). At iteration k, $(x_k, y_k) \in A \times B$. Then, find (x'_k, y'_k) such that $x_k + y_k = x'_k + y'_k$ and $\frac{1}{c}(y'_k)_A + (y'_k)_B \in T((x'_k)_A + \frac{1}{c}(x'_k)_B)$.

* Received by the editors March 26, 1993; accepted for publication (in revised form) February 14, 1994.

† ISIMA, B.P. 125, 63173 Aubière, Cedex, France (mahey@flamengo.isima.fr).

‡ Institut d'Informatique et d'Intelligence Artificielle, Monruz 36, CH-2000 Neuchatel, Switzerland.

§ LMAI, INSA Rouen, Place Emile Blondel, B.P. 8, 76131 Mont-Saint-Aignan, France.

Then, $(x_{k+1}, y_{k+1}) = ((x'_k)_A, (y'_k)_B)$.

The main problem that arises with this algorithm is the difficulty of performing the proximal step (1) when $c \neq 1$ in most interesting situations including the decomposition methods. When $c = 1$, then the proximal step is a proximal decomposition on the graph of T and the subspaces A and B only appear in the projection step. In §3 we present the resultant algorithm, indeed equivalent to PIM with $c = 1$. The convergence is proved without the need to consider the Partial Inverse operator. The iteration is now written in the following way.

Proximal decomposition. Find the unique (x'_k, y'_k) such that $x'_k + y'_k = x_k + y_k$ and $(x'_k, y'_k) \in \mathrm{Gr}(T)$ If $(x'_k, y'_k) \in A \times B$, then stop.
Else $(x_{k+1}, y_{k+1}) = ((x'_k)_A, (y'_k)_B)$.

The unique solution of the proximal decomposition step is given by

$$x'_k = (I + T)^{-1}(x_k + y_k), \tag{1}$$
$$y'_k = (I + T^{-1})^{-1}(x_k + y_k).$$

Of course, only one proximal calculus is needed as $(I + T^{-1})^{-1} = I - (I + T)^{-1}$. We propose then a modified proximal decomposition algorithm by introducing scaling factors λ and μ. Indeed, problem (P) may be written in two ways :

$$y \in Tx \Longleftrightarrow x + \lambda y \in (I + \lambda T)x,$$
$$x \in T^{-1}y \Longleftrightarrow y + \mu x \in (I + \mu T^{-1})y,$$

which induces the following fixed point iteration, a natural scaled version of (1).

Modified proximal decomposition.

$$x'_k = (I + \lambda T)^{-1}(x_k + \lambda y_k), \tag{2}$$
$$y'_k = (I + \mu T^{-1})^{-1}(y_k + \mu x_k).$$

If $(x'_k, y'_k) \in A \times B$, then stop.
Else $(x_{k+1}, y_{k+1}) = ((x'_k)_A, (y'_k)_B)$.

It appears that the modified proximal step is uniquely determined and corresponds to a proximal decomposition on the graph of λT if $\lambda\mu = 1$. We recover then the scaled version of PIM proposed by Spingarn in [13]. It is mentioned in [6] that the performance of PIM is very sensitive to the scaling factor variations and we give an explanation of this fact, allowing its adjustment to an optimal value in the strongly monotone case.

In §4, we give some numerical results that confirm the accelerating properties of the scaling parameter.

2. The proximal decomposition on the graph of a maximal monotone operator. We recall here some known results on the "Prox" mapping $(I + T)^{-1}$ associated to a maximal monotone operator T and focus on the properties of the decomposition on the graph of T. More details on that subject can be found in [2] and [5].

Let T be a maximal monotone operator on a Hilbert space X. The graph of T, denoted by $\mathrm{Gr}(T)$, is defined by

$$\mathrm{Gr}(T) = \{(x, y) \in X \times X | y \in Tx\}.$$

Monotonicity implies that for all $x, x' \in X$ and for all $y \in Tx$, for all $y' \in Tx'$, $\langle y - y', x - x' \rangle \geq 0$. As T is maximal, its graph is not properly contained in the graph of any other monotone operator.

If T is strongly monotone, then there exists a positive ρ such that

$$\forall x, x' \in X \quad \text{and} \quad \forall y \in Tx, \quad \forall y' \in Tx', \quad \langle y - y', x - x' \rangle \geq \rho \|x - x'\|^2.$$

We say that the operator T is Lipschitz with constant L if

$$\forall x, x' \in X \quad \text{and} \quad \forall y \in Tx, \forall y' \in Tx', \|y - y'\| \leq L \|x - x'\|.$$

For monotone operators that share both properties, we get the following explicit bounds:

$$\rho \|x - x'\| \leq \|y - y'\| \leq L \|x - x'\|. \tag{3}$$

When T is a linear operator represented by a positive definite matrix $\mathcal{T}$, the best estimates for ρ and L are, respectively, the smallest and the largest eigenvalues of $\mathcal{T}$.

Of course, if T is maximal monotone, then for any $\lambda > 0$, λT is maximal monotone and if, moreover, T is strongly monotone with modulus ρ and Lipschitz with constant L, then λT is strongly monotone with modulus $\lambda\rho$ and Lipschitz with constant λL.

The resolvent associated with maximal monotone operator T is defined by $(I + T)^{-1}$. It is single-valued, defined on the whole space, and firmly nonexpansive, which means that, if we let $U = (I + T)^{-1}$ and $V = I - U$, then,

$$\forall x, x' \in X, \|Ux - Ux'\|^2 + \|Vx - Vx'\|^2 \leq \|x - x'\|^2 \tag{4}$$

or equivalently

$$\|Ux - Ux'\|^2 \leq \langle x - x', Ux - Ux' \rangle. \tag{5}$$

Related interesting facts on this characteristic property of resolvents may be found in theses by Martinet [9] and Eckstein [3] (see also [5]). Indeed, resolvents and maximal firmly nonexpansive mappings coincide and, following [7], one-to-one correspondences among these operators, maximal monotone, and maximal nonexpansive operators, may be stated. This fact is explored further in the appendix.

We introduce now the proximal decomposition on the graph of a maximal monotone operator.

Given a maximal monotone operator T and a vector $(x, y) \in X \times X$, there exists a unique pair $(u, v) \in X \times X$ called the proximal decomposition of (x, y) on the graph of T such that

$$u + v = x + y \quad \text{and} \quad (u, v) \in \mathrm{Gr}(T).$$

The unicity is a direct consequence of the maximality of T and we get

$$u = (I + T)^{-1}(x + y),$$
$$v = (I + T^{-1})^{-1}(x + y).$$

3. The proximal decomposition algorithm. We return now to problem (P), which has been analyzed by Spingarn [13]. Let T be a maximal monotone operator on X. Let A be a subspace and B its orthogonal subspace. The problem is to find

$$(x, y) \in X \times X \text{ such that } (x, y) \in A \times B \cap \mathrm{Gr}(T).$$

This problem is a particular case of the general problem of finding a zero of the sum of two maximal monotone operators. The algorithms we are aiming at are splitting methods that alternate computations on each operator separately (see [8]). Indeed, most large-scale optimization problems can be formulated as the problem of minimizing a separable convex lower semicontinuous function on a very simple subspace which represents the coupling between the subsystems.

We propose then a generic algorithm that alternates a proximal decomposition on the graph of T with a projection on $A \times B$. Before going on with the analysis of the method, we observe that the other alternatives that come to mind to find a point in the intersection of two sets are not suitable.

1. We can use the classical successive projections method on the two sets. The problem is that $\mathrm{Gr}(T)$ is generally not convex in $X \times X$.

2. We cannot use another proximal decomposition on $A \times B$ (which is indeed the graph of the maximal monotone operator $\partial\chi_A$, the subdifferential of the indicator function of the set A), because it would lead back to the original point! Indeed, if $(x, y) \in A \times B$ and (u, v) is the proximal decomposition of $x + y$ on $\mathrm{Gr}(T)$, then $x = (u+v)_A$ and $y = (u+v)_B$, which means that (x, y) is the proximal decomposition of $u + v$ on the graph of $\partial\chi_A$.

The Algorithm PDG (proximal decomposition on the graph) is stated below.

ALGORITHM 2 (PDG). Let $(x_0, y_0) \in A \times B$. $k = 0$.
If $(x_k, y_k) \in Gr(T)$, then stop: (x_k, y_k) is a solution of (P).
Else compute the proximal decomposition (u_k, v_k) of $x_k + y_k$ on the graph of T. If $(u_k, v_k) \in A \times B$, stop: (u_k, v_k) is a solution of (P).
Else, $x_{k+1} = (u_k)_A$ and $y_{k+1} = (v_k)_B$.
$k = k + 1$

An iteration of the algorithm may be formally stated as

$$(x, y) \in A \times B \mapsto \mathcal{L}(x, y) = x + y = z \in X \mapsto (u, v) = \mathcal{F}z \mapsto \mathcal{P}_{A\times B}(u, v) \in A \times B,$$

where L is isometric from $X \times X$ into X, $\mathcal{F}$ is the proximal decomposition operator from X into $X \times X$, and $P_{A\times B}$ is the projection on $A \times B$. Let us denote the composed mapping by

$$\mathcal{J} = \mathcal{P}_{A\times B} \circ \mathcal{F} \circ \mathcal{L}.$$

We verify now that any fixed point (x, y) of Algorithm PDG is a solution of (P). Indeed, $(x, y) = \mathcal{P}_{A\times B}(u, v)$ and $(u, v) = \mathcal{F}z$ with $z = x + y$. If $(u, v) \in A \times B$, then (x, y) is a solution of (P). Else, we have

$$(u - x, v - y) \in L = \{(a, b) \in X \times X | a + b = 0\}.$$

But, as $(x, y) = \mathcal{P}_{A\times B}(u, v)$, we can state

$$(u - x, v - y) \in B \times A.$$

A and B being orthogonal subspace, the unique intersection of L and $B \times A$ is $(0,0)$. Thus, $(x,y) = (u,v)$ and (x,y) solves (P).

On the other hand, if (x,y) is a solution of (P), $\mathcal{F}(x+y) = (x,y)$, and $(x,y) \in A \times B$, which means that (x,y) is a fixed point of Algorithm PDG.

The PDG Algorithm is a particular instance of Spingarn's Partial Inverse Method [12]. Indeed, when $c = 1$, the proximal step on the Partial Inverse operator T_A becomes: Find (x'_k, y'_k) such that : $x_k + y_k = x'_k + y'_k$ and $(y'_k)_A + (y'_k)_B \in T((x'_k)_A + (x'_k)_B)$, which means, of course, that (x'_k, y'_k) is the proximal decomposition of (x_k, y_k) on the graph of T. Thus, the convergence has been established by Spingarn who has used the properties of the PPA applied to the partial inverse operator. However, here we give a direct proof of this fact that does not use the concept of the Partial Inverse. The main interest is that we shall obtain as a corollary the numerical analysis of the scaled version of PDG in the strongly monotone case.

We prove first that the composed mapping $\mathcal{J}$ associated with Algorithm PDG is firmly nonexpansive. It can easily be seen that the mapping $\mathcal{U} = \mathcal{L} \circ \mathcal{J} \circ \mathcal{L}^{-1}$ is indeed the proximal operator associated to the Partial Inverse of T, i.e., $\mathcal{U} = (I + T_A)^{-1}$. But, we do not use this fact to prove that $\mathcal{J}$ is firmly nonexpansive.

THEOREM 3.1. *The mapping $\mathcal{J}$ associated to Algorithm* PDG *is firmly nonexpansive if and only if T is monotone. Moreover, it is defined on the whole space $A \times B$ if and only if T is maximal monotone.*

Proof. Let (x,y) and $(x',y') \in A \times B$, $z, z' \in \mathcal{L}(x,y), \mathcal{L}(x',y')$ respectively, i.e., $z = x + y$ and $z' = x' + y'$, $(u,v) \in \mathcal{F}(z)$ and $(u',v') \in \mathcal{F}(z')$, i.e., $u + v = z$, $u = (I+T)^{-1}z$ and $u' + v' = z'$, $u' = (I+T)^{-1}z'$. Finally, let (u_A, v_B) and $(u'_A, v'_B) \in A \times B$ be the respective projections of (u,v) and (u',v') on $A \times B$.

It is clear that, as $z \in (I+T)u$, $\mathrm{dom}(\mathcal{F}) = \mathrm{R}(I+T)$, and $\mathrm{dom}(\mathcal{J}) = \mathcal{L}^{-1}(\mathrm{dom}(\mathcal{F})) = \{(z_A, z_B) \in A \times B | z \in \mathrm{dom}(\mathcal{F})\}$.

Now, $\mathcal{J}$ is firmly nonexpansive if and only if

$$(6) \quad \begin{array}{l} \forall (x,y),(x',y') \in \mathrm{dom}(\mathcal{J}) \quad \text{and} \quad \forall (u_A,v_B) \in \mathcal{J}(x,y), (u'_A, v'_B) \in \mathcal{J}(x',y') \\ \langle (x,y) - (x',y'), (u_A,v_B) - (u'_A, v'_B) \rangle \geq \|(u_A,v_B) - (u'_A, v'_B)\|^2. \end{array}$$

But, we have

$$\langle (x,y) - (x',y'), (u_A,v_B) - (u'_A, v'_B) \rangle = \langle x - x', (u-u')_A \rangle + \langle y - y', (v-v')_B \rangle$$

and, as $x, x' \in A$ and $y, y' \in B$

$$\begin{aligned} \langle x - x', (u - u')_A \rangle &= \langle z - z', (u-u')_A \rangle \\ &= \langle (u + v - u' - v', (u-u')_A \rangle \\ &= \langle (u-u') + (v-v'), (u-u')_A \rangle \end{aligned}$$

and

$$\begin{aligned} \langle y - y', (v-v')_B \rangle &= \langle z - z', (v-v')_B \rangle \\ &= \langle (u-u') + (v-v'), (v-v')_B \rangle. \end{aligned}$$

Hence, inequality (6) becomes

$$\begin{aligned} &\langle (u-u') + (v-v'), (u-u')_A \rangle \\ &\qquad + \langle (u-u') + (v-v'), (v-v')_B \rangle \geq \|(u-u')_A\|^2 + \|(v-v')_B\|^2. \end{aligned}$$

We can now use the orthogonal decomposition of $u-u'$ and $v-v'$ on the direct sum $A \oplus B$ to get

$$\begin{aligned} &\forall (u,v),(u',v') \in \mathrm{Gr}(T), \\ &\langle (u-u')_A, (v-v')_A \rangle + \langle (u-u')_B, (v-v')_B \rangle \geq 0. \end{aligned}$$

Finally, remarking that

$$\langle u-u', v-v' \rangle = \langle (u-u')_A, (v-v')_A \rangle + \langle (u-u')_B, (v-v')_B \rangle,$$

we can conclude that $\mathcal{J}$ is firmly nonexpansive if and only if T is monotone.

Moreover, as $\mathrm{dom}(\mathcal{J}) = \{(x,y) \in A \times B | x+y \in \mathrm{dom}(\mathcal{F})\}$, we obtain

$$\left.\begin{array}{l} \mathcal{J} \text{firmly nonexpansive} \\ \mathrm{dom}(\mathcal{J}) = A \times B \end{array}\right\} \Longleftrightarrow \left\{\begin{array}{l} T \text{monotone} \\ \mathrm{dom}(\mathcal{F}) = X \end{array}\right. \Longleftrightarrow T \text{maximal monotone.} \quad \square$$

Assuming that (P) has a solution, the convergence of the algorithm follows directly from Opial's lemma (see [10]), which states that, if a fixed point exists, a firmly nonexpansive operator is asymptotically regular and generates a convergent sequence. This is the very same idea as used by Martinet in the original proof for the PPA [9] and developed further by Rockafellar who included approximate computations of the proximal steps [11] .

4. A scaled decomposition on the graph of T. We introduce now a scaled version of the decomposition on the graph of a maximal monotone operator.

DEFINITION 4.1. *Let $(x,y) \in X \times X$, T be a maximal monotone operator and λ a positive number. Then, the scaled proximal decomposition of (x,y) on the graph of T is the unique (u,v) such that*

$$\begin{aligned} u + \lambda v &= x + \lambda y, \\ (u,v) &\in \mathrm{Gr}(T). \end{aligned}$$

Again, the existence and unicity of that new decomposition is a consequence of T being maximal monotone. Indeed, if $v \in Tu$, we can write

$$\begin{aligned} u + \lambda v &\in u + \lambda T u \\ \Rightarrow u &= (I + \lambda T)^{-1}(u + \lambda v) \\ &= (I + \lambda T)^{-1}(x + \lambda y) \\ v &= \lambda^{-1}(x + \lambda y - u). \end{aligned}$$

Observe that we can also write the following inclusions using the inverse operator T^{-1} for a given positive μ:

$$\begin{aligned} u &\in T^{-1} v, \\ v + \mu u &\in v + \mu T^{-1} v, \\ \text{then } v &= (I + \mu T^{-1})^{-1}(v + \mu u). \end{aligned}$$

Now, if μ satisfies $\mu^{-1} = \lambda$, we get $v + \mu u = \mu(u + \lambda v)$ and, using the fact that $(\mu T)^{-1} z = T^{-1}(\mu^{-1} z)$, we obtain

$$\begin{aligned} v &= \lambda^{-1}(I + \mu T^{-1})^{-1}(u + \lambda v) \\ &= (I + \mu T^{-1})^{-1}(\mu x + y). \end{aligned}$$

Resuming, the scaled decomposition on the graph of T can be defined by

$$\begin{aligned} u &= (I+\lambda T)^{-1}(x+\lambda y), \\ v &= (I+\mu T^{-1})^{-1}(\mu x+y), \end{aligned} \tag{7}$$

which appears as a natural generalization of (1). But, in fact, only one scaling factor can be introduced to maintain the desired properties, this is why we must fix $\lambda\mu = 1$.

We can now describe the iteration of a scaled version of Algorithm PDG.

ALGORITHM 3 (SPDG). $(x_k, y_k) \in A \times B$.
Compute the scaled decomposition of (x_k, y_k) on the graph of T.

$$\begin{aligned} u_k &= (I+\lambda T)^{-1}(x_k+\lambda y_k), \\ v_k &= \lambda^{-1}(x_k+\lambda y_k-u_k). \end{aligned}$$

If $(u_k, v_k) \in A \times B$, then stop. Else, $x_{k+1} = (u_k)_A$ and $y_{k+1} = (v_k)_B$.

Observe that the scaled proximal decomposition can be stated in the following way.

Let $w = \lambda v$ and $r = \lambda y$. Then, if (u, v) is the scaled proximal decomposition of (x, y) on the graph of T, (u, w) is the proximal decomposition of (x, r) on the graph of λT. Hence, from the preceding section, we know that the sequence $\{(x_k, r_k)\}$ converges to a point in $A \times B \cap \mathrm{Gr}(\lambda T)$. This fact implies that the sequence $\{(x_k, y_k)\}$ converges to a solution of (P).

On the other hand, we can see that SPDG is equivalent to the scaled version of the Partial Inverse Method (with $c = 1$) described by Spingarn in [13, Algorithm 2, p. 208] for the minimization of a convex function on a subspace. It reduces, of course, to PDG, i.e., to PIM, when $\lambda = 1$. Again, as the decomposition on the graph of T is a proximal step, approximate rules for computations can be added as in [11] to get an implementable algorithm. We prefer to omit these details to focus on the accelerating properties of the scaling parameter, which constitute the main contribution of the present work.

To analyze the influence of the scaling parameter on the speed ratio of convergence of SPDG, we consider now the case where T is both strongly monotone and Lipschitz.

THEOREM 4.2. *When T is strongly monotone with modulus ρ and Lipschitz with constant L, then the convergence of the sequence $\{(x_k, r_k)\}$ generated by* SPDG *with $r_k = \lambda y_k$ is linear with speed ratio*

$$\sqrt{1-\frac{2\lambda\rho}{(1+\lambda L)^2}}.$$

Proof. If $\mathcal{J}_\lambda$ is the composed operator associated to the monotone operator λT, we define as in Theorem 3.1 $(x, r), (x', r') \in A \times B$, $z = x + r$, $z' = x' + r'$, $(u, w), (u', w') \in \mathrm{Gr}(\lambda T)$. Then, $(u_A, w_B) \in \mathcal{J}_\lambda(x, r)$ and $(u'_A, w'_B) \in \mathcal{J}_\lambda(x', r')$.

The strong monotonicity of λT implies that

$$\forall w \in Tu, \quad w' \in Tu', \quad \langle w - w', u - u' \rangle \geq \lambda\rho\|u - u'\|^2 \tag{8}$$

and, as $z \in (I + \lambda T)u$ and $z' \in (I + \lambda T)u'$

$$\|z - z'\| \leq (1 + L)\|u - u'\|. \tag{9}$$

From the composed nature of $\mathcal{J}_\lambda$ and using the relations (8) and (9), we deduce the following bounds:

$$\begin{aligned}\|(u_A, w_B) - (u'_A, w'_B)\|^2 &\leq \|u - u'\|^2 + \|w - w'\|^2 \\ &\leq \|z - z'\|^2 - 2\langle u - u', w - w'\rangle \\ &\leq \|z - z'\|^2 - 2\rho\|u - u'\|^2 \\ &\leq \left(1 - \frac{2\lambda\rho}{(1+\lambda L)^2}\right)\|z - z'\|^2 \\ &\leq \left(1 - \frac{2\lambda\rho}{(1+\lambda L)^2}\right)\|(x, r) - (x', r')\|^2.\end{aligned}$$

Let (x^*, r^*) be the limit point of the sequence $\{(x_k, r_k)\}$. It is therefore a fixed point of the mapping $\mathcal{J}_\lambda$. Applying the above inequality to the pairs (x_{k+1}, r_{k+1}) and (x^*, r^*), we obtain the desired result:

$$\|(x_{k+1}, r_{k+1}) - (x^*, r^*)\|^2 \leq \left(1 - \frac{2\lambda\rho}{(1+\lambda L)^2}\right)\|(x_k, r_k) - (x^*, r^*)\|^2. \qquad \square$$

Observe that, as

$$L \geq \rho, r(\lambda) = \sqrt{1 - \frac{2\lambda\rho}{(1+\lambda L)^2}} < 1.$$

We easily deduce the theoretical optimal value for λ:

$$\overline{\lambda} = 1/L \quad \text{and} \quad r(\overline{\lambda}) = \sqrt{1 - \frac{\rho}{2L}}. \tag{10}$$

When T is a linear positive definite operator, we observe that bad conditioning implies a slowdown of the algorithm. The optimal value of the scaling parameter must be chosen very small if $L = \mu_{\max}$, the largest eigenvalue of the associated matrix, is very large. We may observe that the speed ratio obtained in Theorem 4.2 is the same as the one given in [8] for the Douglas–Rachford splitting algorithm. Indeed, the connection between that algorithm and the Partial Inverse Method has been established by Eckstein [3] and we give its precise meaning in the Appendix.

The influence of the Lipschitz constant on the number of iterations has been analyzed for quadratic convex functions that were minimized on a simple subspace. The sensitivity to that parameter is shown on the five graphics of Fig. 1 and 2 for different values of L, ρ, and the dimension of the space. These results are shown in Table 1. The influence of the scaling parameter on the number of iterations is illustrated by comparing columns iter($\overline{\lambda}$) (number of iterations when $\lambda = \overline{\lambda}$) and iter(1) (number of iterations when $\lambda = 1$). The number of iterations corresponds to the implementation of Algorithm PDG associated with the graph of λT. We show below why it is faster than the straightforward application of SPDG even if the primal sequences $\{x_k\}$ coincide in both algorithms.

It is also interesting to analyze the behaviour of the sequence $\{(x_k, y_k)\}$ and to look for some values of the scaling parameter such that, that sequence is mapped by a contraction. To be more precise, let $\mathcal{J}_\lambda$ and $\mathcal{H}_\lambda$ be the maps associated with the sequences $\{(x_k, r_k)\}$ and $\{(x_k, y_k)\}$, respectively. Then, if D_λ is the mapping defined by

$$D_\lambda(x, y) = (x, \lambda y),$$

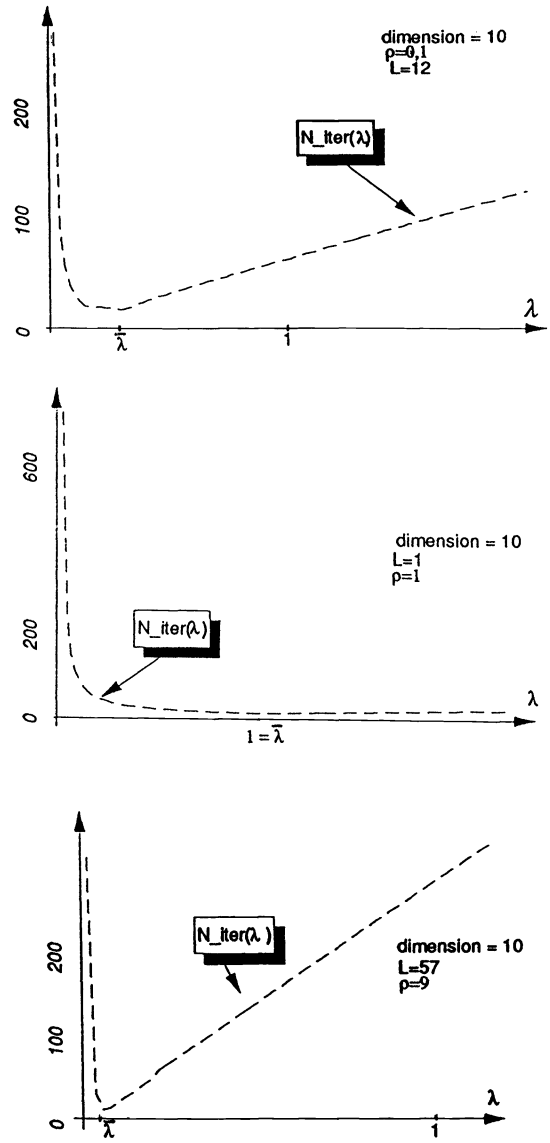


FIG. 1. *Number of iterations for* dim=10.

we can write the following correspondence:

$$\mathcal{H}_\lambda = D_\lambda^{-1} \circ \mathcal{J}_\lambda \circ D_\lambda.$$

As $(x_k, y_k) = D_\lambda^{-1}(x_k, r_k)$, we already know that the sequence $\{(x_k, y_k)\}$ converges when $\{(x_k, r_k)\}$ converges. Note that a direct proof of this fact seems rather hard to state. The reason is that $\mathcal{H}_\lambda$ is not necessarily a contractive map for any λ. We study below the conditions on λ to get a contraction in the strongly monotone case. In the strongly monotone and Lipschitz cases, we already know that $\mathcal{H}_\lambda$ is a contraction for $\lambda = 1$. The next theorem shows that this remains true if λ lies in a specific interval containing one.

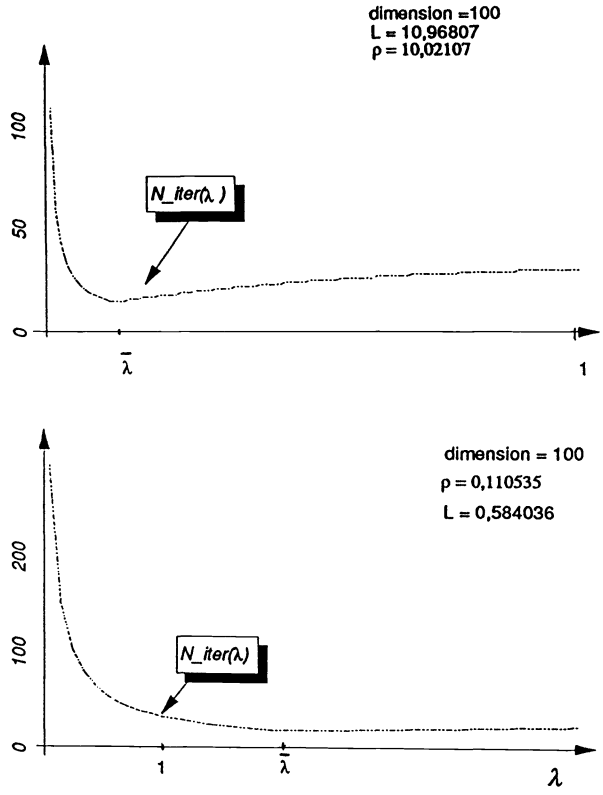


FIG. 2. *Number of iterations for* dim=100.

THEOREM 4.3. *Suppose that T is strongly monotone with modulus ρ and Lipschitz with constant L. Then, if $\lambda \in [1, \rho + \sqrt{1+\rho^2})$, the mapping $\mathcal{H}_\lambda$ is a contraction.*

Proof. Again let $u = (I + \lambda T)^{-1}(x + \lambda y)$ and $u' = (I + \lambda T)^{-1}(x' + \lambda y')$. We use successively the nonexpansiveness of the projection and the firmly nonexpansiveness of the resolvent to write

$$\begin{aligned}\|\mathcal{H}_\lambda(x,y) - \mathcal{H}_\lambda(x',y')\|_X^2 &\le \|(u-u', \lambda^{-1}(x-x' + \lambda(y-y') - u + u'))\|_X^2 \\ &\le \lambda^{-2}\|x-x'\|^2 + \|y-y'\|^2 + \frac{\lambda^2 - 2\lambda\rho - 1}{\lambda^2}\|u-u'\|^2.\end{aligned}$$

Using the Lipschitz property, we obtain

$$\|\mathcal{H}_\lambda(x,y) - \mathcal{H}_\lambda(x',y')\|_X^2 \le \lambda^{-2}\left(1 + \frac{\lambda^2 - 2\lambda\rho - 1}{(1+\lambda L)^2}\right)(\|x-x'\|^2 + \lambda^2\|y-y'\|^2). \tag{11}$$

Hence, a sufficient condition that ensures that $\mathcal{H}_\lambda$ is a contraction is $\lambda \ge 1$ and $\theta(\lambda) = \lambda^2 - 2\lambda\rho - 1 < 0$. That condition does not depend on the Lipschitz constant (indeed, this happens because $0 < \rho < L$). We observe now that $\theta(1) = -2\rho < 0$ and the desired interval must be : $\lambda \in [1, \rho + \sqrt{1+\rho^2})$. □

The different behaviour of both sequences $\{(x_k, z_k)\}$ and $\{(x_k, y_k)\}$ is illustrated in Fig. 3. For a small λ, the second sequence (which is the one that will yield a solution

TABLE 1
Numerical tests for quadratic problems.

ρ	L	$1/L$	$\bar{\lambda}$	Iter($\bar{\lambda}$)	Iter(1)	dim	tolerance
	0.584	1.712	2.05	17	34		
	1.068	0.936	1.05	17	20		
	4.940	0.202	0.26	18	29		
0.1	9.781	0.102	0.13	18	42		
	19.461	0.051	0.07	18	60		
	29.142	0.034	0.05	18	70		
	96.907	0.010	0.02	18	92	100	0.01
	1.968	0.508	0.58	17	19		
	5.840	0.171	0.12	18	35		
	10.681	0.094	0.12	17	44		
1	20.361	0.049	0.06	18	60		
	30.042	0.033	0.05	18	70		
	49.404	0.020	0.03	18	82		
	97.807	0.010	0.02	19	92		
	0.584	1.712	2.04	16	32		
	1.068	0.936	1.21	16	18		
	4.940	0.202	0.241	17	27		
0.1	9.781	0.102	0.121	17	39		
	19.461	0.051	0.061	17	54		
	29.142	0.034	0.041	17	63		
	96.907	0.010	0.021	17	75	10	0.001
	1.968	0.508	0.58	15	16		
	5.840	0.171	0.211	16	29		
	10.681	0.094	0.121	16	40		
1	20.361	0.049	0.061	17	54		
	30.042	0.033	0.041	17	63		
	49.404	0.020	0.031	17	72		
	97.807	0.010	0.021	17	75		

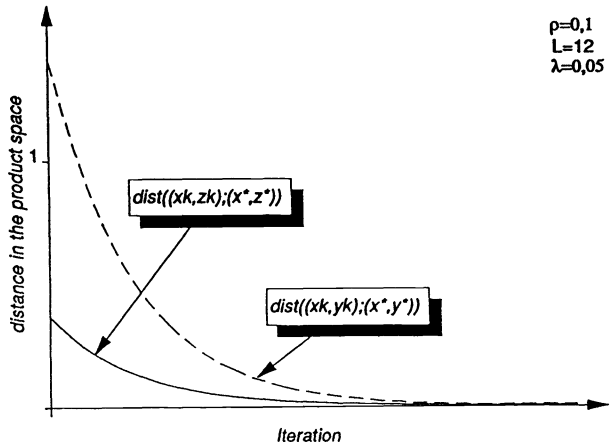


FIG. 3. *Comparison of both sequences*

for the original problem (P)) converges much slower even if it presents a monotonic decrease toward the fixed point.

We conclude with the following observations on the choice of the scaling parameter: if bad conditioning is due to a too-small ρ, then we must accelerate the convergence by choosing λ close to the optimal value $1/L$ (if it is not too far from

1!). If bad conditioning is due to a too-large L, then we may choose λ close to 1 in $[1, \rho + \sqrt{1+\rho^2})$.

Appendix. The relation between the partial inverse and the Douglas–Rachford splitting operator may be explained in the following way which is directly inspired by the work of Lawrence and Spingarn [7]. It was later derived by Eckstein and Bertsekas [4].

We recall the one-to-one correspondences among maximal monotone operators, maximal nonexpansive, and proximal operators as described in [7].

Let $\alpha : (x, y) \mapsto (x, 2y - x)$ be the one-to-one correspondence of the class of proximal operators onto the class of nonexpansive operators and let $\beta : (x, y) \mapsto (x + y, x - y)$ be the one-to-one correspondence of the class of monotone operators onto the class of nonexpansive operators. Following [7], let us define two types of composition operations.

Let $p_1 \star p_2 = \alpha^{-1}(\alpha(p_1) \circ \alpha(p_2))$ be the proximal operator obtained by composing two proximal operators p_1 and p_2 through their associated respective nonexpansive images (which give indeed another nonexpansive operator when composed). Likewise, let $T_1 \odot T_2 = \beta^{-1}(\beta(T_1) \circ \beta(T_2))$ be the monotone operator obtained by composing two monotone operators in the same way. A straightforward calculus shows that, if p_1 and p_2 are the resolvents of T_1 and T_2, respectively, then $p = p_1 \star p_2$ is the resolvent of $T = T_1 \odot T_2$. As observed in [7], we have the following interpretation of the $\star$ operation :

$$p_1 \star p_2 = p_1 \circ (2p_2 - I) + I - p_2,$$

which is the operator associated to the fixed point iteration of the Douglas–Rachford splitting method (see [8]). Observe that the nonexpansive operator $\alpha(p_1) \circ \alpha(p_2))$ is the operator associated with the Peaceman–Rachford iteration.

On the other side, it is shown in [7] that, when T_1 is the subdifferential mapping of the indicator function of a subspace A, i.e., $\mathrm{Gr}(T_1) = A \times A^{\perp}$, then $T_1 \odot T = T_A$, the Partial Inverse of T. Resuming these facts, we have the following proposition.

PROPOSITION. *Let T_1 and T_2 be two maximal monotone operators on X. The Douglas–Rachford splitting operator $p = p_1 \circ (2p_2 - I) + I - p_2$, where $p_1 = (I + \lambda T_1)^{-1}$ and $p_2 = (I + \lambda T_2)^{-1}$, is a proximal operator, indeed $p = (I + T)^{-1}$, where $T = \lambda T_1 \odot \lambda T_2$. Moreover, if $Gr(T_1) = A \times A^{\perp}$ and $T_2 = T$, then $p = (I + (\lambda T)_A)^{-1}$, the resolvent of the partial inverse of λT. Then, the Douglas–Rachford iteration applied to problem* (P) *is the partial inverse method associated to λT.* SPDG *is the corresponding algorithm defined in the product space $X \times X$.*

Observation. Clearly $(I + (\lambda T)_A)^{-1} \neq (I + \lambda T_A)^{-1}$. This point is crucial because the computation can only be performed in the first expression (this is then the SPDG Algorithm) or in the second expression with $\lambda = 1$.

REFERENCES

[1] D. BERTSEKAS AND J. TSITSIKLIS, *Parallel and Distributed Computation*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[2] H. BREZIS, *Opérateurs Maximaux Monotones*, Mathematics Studies 5, North Holland, Amsterdam, 1973.

[3] J. ECKSTEIN, *Splitting Methods for Monotone Operators with Applications to Parallel Optimization*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, 1989.

[4] J. ECKSTEIN AND D. BERTSEKAS, *On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators*, Math. Programming, 55 (1992), pp. 293–318.

[5] K. GOEBEL AND W. KIRK, *Topics in Metric Fixed Point Theory*, Studies in Advanced Mathematics 28, Cambridge University Press, 1990.

[6] H. IDRISSI, O. LEFEBVRE, AND C. MICHELOT, *Applications and numerical convergence of the partial inverse method*, in Optimization, Lecture Notes in Math. 1405, S. Dolecki, ed., Springer-Verlag, 1989, New York, pp. 39–54.

[7] J. LAWRENCE AND J. SPINGARN, *On fixed points of nonexpansive piecewise isometric mappings*, Proc. London Math. Soc., 55 (1987), pp. 605–624.

[8] P. L. LIONS AND B. MERCIER, *Splitting algorithms for the sum of two nonlinear operators*, SIAM J. Numer. Anal., 16 (1979), pp. 964–979.

[9] B. MARTINET, *Algorithmes pour la Résolution de Problèmes d' Optimisation et de Minimax*, Thèse d'Etat, University de Grenoble, 1972.

[10] Z. OPIAL, *Weak convergence of the successive approximations for nonexpansive mappings in Banach spaces*, Bull. Amer. Math. Soc., 73 (1967), pp. 591–597.

[11] R. ROCKAFELLAR, *Monotone operators and the proximal point algorithm in convex programming*, SIAM J. Control Optim., 14 (1976), pp. 877–898.

[12] J. SPINGARN, *Partial inverse of a monotone operator*, Appl. Math. Optim., 10 (1983), pp. 247–265.

[13] ———, *Applications of the method of partial inverse to convex programming: Decomposition*, Math. Programming, 32 (1985), pp. 199–223.

SIAM J. OPTIMIZATION
Vol. 5, No. 3, pp. 467–487, August 1995

NONPOLYHEDRAL RELAXATIONS OF GRAPH-BISECTION PROBLEMS*

SVATOPLUK POLJAK† AND FRANZ RENDL‡

Abstract. We study the problem of finding the minimum bisection of a graph into two parts of prescribed sizes. We formulate two lower bounds on the problem by relaxing node- and edge-incidence vectors of cuts. We prove that both relaxations provide the same bound. The main fact we prove is that the duality between the relaxed edge- and node- vectors preserves very natural cardinality constraints on cuts.

We present an analogous result also for the max-cut problem, and show a relation between the edge relaxation and some other optimality criteria studied before. Finally, we briefly mention possible applications for a practical computational approach.

Key words. max-cut problem, graph bisection, positive semidefinite relaxations

AMS subject classifications. 90C27, 90C34

1. Introduction.

We consider the problem of decomposing a weighted graph G on n nodes into two parts of prescribed sizes n_1 and n_2, $n_1+n_2=n$, so that the total sum of the edge weights between the parts is minimum. For an edge weight function c, and $k := n_1 - n_2$, we denote the minimum by $b(G,c,k)$.

We consider two lower bounds $\nu(G,c,k)$ and $\eta(G,c,k)$ on the bisection number $b(G,c,k)$, and call them the *node-* and *edge-relaxation*, respectively. We prove that these two bounds are equal by means of duality. An interesting fact that we want to emphasize is that the duality preserves certain cardinality constraints that are trivially satisfied by the integer vectors. To be specific, let $(S, V \setminus S)$ be a bipartition of the node set V, and let $x = (x_i) \in \Re^n$ and $y = (y_{ij}) \in \Re^{\binom{n}{2}}$ be the *node-* and *edge-incidence* vectors of this bipartition, respectively, defined by

$$x_i = \begin{cases} 1 & i \in S, \\ -1 & i \notin S, \end{cases} \tag{1}$$

and

$$y_{ij} = \begin{cases} 1 & i \in S, j \notin S, \\ 0 & \text{elsewhere.} \end{cases} \tag{2}$$

Then the partition classes have sizes n_1 and n_2 if and only if either

$$|\sum_{i=1}^{n} x_i| = |n_1 - n_2| \tag{3}$$

* Received by the editors January 11, 1993; accepted for publication (in revised form) February 17, 1994.

† University of Passau, Faculty of Mathematics and Informatics, Innstrasse 33, 94030 Passau, Germany until his death in 1995. A tribute to S. Poljak will appear in a special issue of *Math Programming* (Series B, 1996) dedicated to his memory. The research was done in part while the author was a visitor at the Center for Discrete Mathematics and Theoretical Computer Sciences Special Year in Combinatorial Optimization.

‡ Technische Universität Graz, Institut für Mathematik, Kopernikusgasse 24, A-8010 Graz, Austria. The work of this author was supported in part by the Christian Doppler Labor Diskrete Optimierung (rendl@fmatbds01.tu-graz.ac.at).

or

$$\sum_{1\le i<j\le n} y_{ij} = n_1 n_2 \ . \tag{4}$$

We will show in §4 that the correspondence between an x satisfying (3) and a y satisfying (4) remains preserved even if (1) is relaxed to $\sum_{i=1}^n x_i^2 = n$ and (2) is relaxed to

$$\sum_{1\le i<j\le n} b_i b_j y_{ij} \le \frac{1}{4}\left(\sum_{i=1}^n b_i\right)^2 \tag{5}$$

for each vector $b = (b_1, \ldots, b_n) \in \Re^n$. This condition will correspond to a positive semidefiniteness constraint of a matrix derived from y.

NOTATION. We fix some notation that will be used throughout the paper. Given a symmetric $n \times n$ matrix M, we denote by $\lambda_1 \le \cdots \le \lambda_n$ its eigenvalues. The minimum and maximum eigenvalue of M will also be denoted as $\lambda_{\min}(M)$ and $\lambda_{\max}(M)$, respectively. We will frequently use the Rayleigh expression for the extreme eigenvalues

$$\lambda_{\min}(M) = \min_{\|x\|=1} x^t M x \quad \text{and} \quad \lambda_{\max}(M) = \max_{\|x\|=1} x^t M x \ . \tag{6}$$

The eigenspace of an eigenvalue $\lambda(M)$ will be denoted as $\mathrm{Eig}(\lambda(M))$. We write $M \succeq 0$ to denote that M is positive semidefinite, i.e., $x^t M x \ge 0$ for all x.

Finally, we will use $\mathcal{U}$ and $\mathcal{X}_k$ to denote the following special subsets of vectors of $\Re^n$,

$$\mathcal{U} := \left\{ u \in \Re^n \mid \sum_{i=1}^n u_i = 0 \right\}, \tag{7}$$

$$\mathcal{X}_k := \left\{ x \in \Re^n \mid \|x\| = 1 \text{ and } \sum_{i=1}^n x_i = \frac{k}{\sqrt{n}} \right\} \ . \tag{8}$$

Capital letters will be used sometimes to denote the corresponding diagonal matrix, as $U = \mathrm{diag}(u)$ for a vector $u \in \Re^n$.

We use e to denote the n-vector of ones and define $J := ee^t$ to be the all-ones matrix.

Let $y = (y_{ij}) \in \Re^{\binom{n}{2}}$, $1 \le i < j \le n$, be a vector of $\binom{n}{2}$ variables. We associate to y the $n \times n$ symmetric matrix $Y = (y_{ij})$ with zero diagonal and entries ij and $ji = y_{ij}$, $i < j$.

We will feel free to consider real $n \times n$ matrices as vectors in $\Re^{n^2}$ and vice versa.

$G = (V, E)$ denotes a graph on n nodes, with node set V and edge set E. We assume $V = \{1, \ldots, n\}$. An edge-weight function is denoted by c. We consider c as a vector in $\Re^{\binom{n}{2}}$ where c_{ij} is the weight of an edge $e = ij$ for $ij \in E$, and $c_{ij} = 0$ for $ij \notin E$, $i < j$. Occasionally, we also use the symmetric weight matrix $C = (C_{ij})$, where $C_{ij} = C_{ji} = c_{ij}$ and $C_{ii} = 0$ for all i. The pair (G, c) denotes the weighted graph G with edge weights c.

The *weighted degree* d_i of a node i is defined by setting $d_i := \sum_{j=1}^{n} C_{ij}$. The vector of weighted degrees is denoted as $d = (d_i)$, and $D = \mathrm{diag}(d)$ denotes the corresponding diagonal matrix. The matrix $L = L(G,c) := D - C$ is called the *Laplacian matrix* of the weighted graph (G,c). Let us recall a useful identity satisfied by the Laplacian matrix for every vector $x \in \Re^n$

$$x^t L x = \sum_{1 \le i < j \le n} c_{ij}(x_i - x_j)^2 \ . \tag{9}$$

We now state the problems under consideration.

Graph bisection. Let (G,c) be a weighted graph on n nodes an n_1, n_2 be a pair of positive integers satisfying $n_1 + n_2 = n$. Let k be such that $n_1 = \frac{n-k}{2}$ and $n_2 = \frac{n+k}{2}$. We define the *bisection width* $b(G,c,k)$ as the minimum total weight of an edge-cut induced by a bipartition of G into two parts of sizes n_1 and n_2, respectively. That is,

$$b(G,c,k) := \min_{|S|=n_1, S \subset V} \sum_{i \in S, j \notin S} c_{ij}.$$

Clearly, the integer k must satisfy $0 \le k \le n$ and $n - k \equiv 0 \pmod 2$. We will call any such k *admissible.*

Graph bisection under inequality constraints. The previous problem can be formulated also in a slightly more general version where the sizes n_1 and n_2 are not exactly prescribed but rather constrained by a condition $\ell \le n_2 - n_1 \le k$, where the bounds ℓ and k are given in advance. Obviously, the exact bisection corresponds to the case when $\ell = k$.

Given a weighted graph (G,c) and the bounds ℓ, k, where $0 \le \ell \le k \le n$, let $b(G,c,\ell,k)$ denote the minimum total weight of an edge-cut induced by a bipartition $(S, V \setminus S)$ satisfying $\ell \le |S| - |V \setminus S| \le k$.

Max-Cut. Given a weighted graph (G,c), the *max-cut problem* asks to find a bipartition $(S, V \setminus S)$ such that the total weight of the edges between S and $V \setminus S$ is maximum.

$$mc(G,c) := \max_{S \subset V} \sum_{i \in S \ j \notin S} c_{ij} \ .$$

Thus, the max-cut problem can be viewed as a special case of the constrained bisection problem by setting the bounds $\ell = 0$ and $k = n$ and the objective function is $c' := -c$. It is advantageous however to focus on the max-cut problem independently. The max-cut problem is an *unconstrained* optimization problem; hence the main ideas of our paper also become most transparent for it.

We note the following identity relating node and edge incidence vectors of bipartitions. Let $(S, V \setminus S)$ be a bipartition with node-incidence vector x and matrix Y corresponding to the edge-incidence vector y. Then

$$\frac{1}{2}J - Y = \frac{1}{2}xx^t. \tag{10}$$

The paper is organized as follows. In §2 we develop and summarize the mathematical tools to derive our main results. These tools are based on a duality theory over convex cones. In §3 we apply these tools to show that the *node relaxation* and the *positive semidefinite edge relaxation* of max-cut form a pair of dual programs satisfying

strong duality. We show how existing optimality certificates for the node relaxation are related to the complementary slackness condition of the two programs. In §§4 and 5 we show a similar duality result for the bisection problem with and without constraints. We point out that the results of §3 are implied by the subsequent sections. We have chosen to study the max-cut problem independently, because this allows us to develop our basic proof strategy in more detail and it may help the reader become familiar with our use of duality. In the last section we propose a computational framework where the two approaches are combined.

2. Duality over cones. In this section we summarize the tools necessary to prove our main results. For a closed convex cone $C \subset \Re^n$, let $C^* = \{x \in \Re^n \mid x^t y \geq 0 \text{ for all } y \in C\}$. It is well known that $C^{**} = C$ for every closed convex cone.

Let A be a matrix, b, c vectors, and Ω and S closed convex cones. Consider the following pair of problems.

$$\min c^t y \quad \text{subject to} \quad Ay - b \in S,\ y \in \Omega, \tag{11}$$

$$\max b^t x \quad \text{subject to} \quad c - A^t x \in \Omega^*, x \in S^*. \tag{12}$$

Problem (11) is said to satisfy the *generalized Slater condition* if there exists some $y \in$ rel-int Ω such that $Ay - b \in$ rel-int S (where rel-int denotes the *relative interior*).

The following duality lemma is a special case of a more general result of [20, Thm. 4.1].

LEMMA 2.1. [20]

(i) (*weak duality*) *If y is a feasible solution of* (11) *and x is a feasible solution of* (12), *then $b^t x \leq c^t y$.*

(ii) (*strong duality*) *Assume that* (11) *satisfies the generalized Slater condition. If both* (11) *and* (12) *have a feasible solution, then* $\min c^t y = \max b^t x$.

(iii) (*complementary slackness*) *If y is an optimal solution of* (11) *and x is an optimal solution of* (12), *then both $y^t(c - A^t x) = 0$ and $x^t(Ay - b) = 0$.*

In our applications we are mostly interested in the cone of positive semidefinite matrices, which we denote by Psd.

$$\text{Psd} := \{A \in \Re^{n \times n} : A = A^t, A \succeq 0\}.$$

We will use the usual inner product in the space of real square matrices:

$$\langle A, B \rangle := \text{tr} AB^t = \sum_{i,j} a_{ij} b_{ij}.$$

(Note that this is consistent with the inner product for vectors, if we consider the matrices as vectors in $\Re^{n^2}$.) To describe the dual cone Psd* we introduce the set

$$\text{Skew} := \{A \in \Re^{n \times n} : A = -A^t\}$$

of real skewsymmetric matrices and point out that skewsymmetric matrices are orthogonal to symmetric matrices, i.e., $\langle A, B \rangle = 0$ for $A = A^t, B \in$ Skew.

PROPOSITION 2.1. *It holds that*

$$\text{Psd}^* = \text{Psd} + \text{Skew}.$$

Proof. Let $X = A + B \in \text{Psd} + \text{Skew}$. Then $Y \in \text{Psd}$ implies $\langle X, Y \rangle = \langle A, Y \rangle \geq 0$, so $X \in \text{Psd}^*$. Conversely let $X \in \text{Psd}^*$. X has a (unique) representation $X = A + B$

where $B \in$ Skew and A symmetric. We conclude that $0 \leq \langle X, Y\rangle = \langle A, Y\rangle$ for all $Y \in$ Psd, therefore $A \in$ Psd. □

Let $C_1 := \{A \in \Re^{n\times n} : \langle J, A\rangle = 0\}$. Then C_1 is a closed convex cone, and clearly $C_1^* = \{tJ : t \in \Re\}$. We will also need the following subcone $K := \text{Psd} \cap C_1$ of Psd.

LEMMA 2.2. *It holds that*

(i) K *is a closed convex cone,*

(ii) $(\frac{1}{2} - p)J + pI \in$ rel-int K *where* $p = \frac{n}{2(n-1)}$,

(iii) $K^* = \text{Psd}^* + C_1^*$.

Proof. (i) K is the intersection of two closed convex cones, hence K is a closed convex cone as well.

(ii) Let $\bar{A} := (\frac{1}{2} - p)J + pI$ where $p = \frac{n}{2(n-1)}$. The eigenvalues of $(\frac{1}{2} - p)J$ are $(0, \ldots, 0, n(\frac{1}{2} - p))$ ($n-1$ times the eigenvalue 0). Therefore the eigenvalues of $\bar{A}$ are $(p, p, \ldots, p, \frac{n}{2} - np + p)$, and they are positive except the last one which is zero due to the definition of p. Thus $\bar{A} = (\bar{a}_{ij})$ is positive semidefinite and $\bar{A} \in K$. Let $A = (a_{ij})$ be an arbitrary (symmetric) matrix such that $e^t A e = 0$ and $|\bar{a}_{ij} - a_{ij}| \leq \varepsilon$, where

$$\varepsilon := \frac{\lambda_2(\bar{A})}{n} = \frac{1}{2(n-1)}.$$

Given a vector w, satisfying $w^t e = 0$ and $\|w\| = 1$, we have

$$w^t A w = w^t(\bar{A} - (\bar{A} - A))w = w^t \bar{A} w - w^t(\bar{A} - A)w$$
$$\geq \lambda_2(\bar{A}) - \varepsilon \left(\sum_{i=1}^{n} |w_i|\right)^2 \geq \lambda_2(\bar{A}) - \varepsilon n = 0.$$

Hence, every symmetric matrix A in the ε-neighbourhood (defined by $|\bar{a}_{ij} - a_{ij}| \leq \varepsilon$) of $\bar{A}$ is positive semidefinite as well. This proves that $\bar{A} \in$ rel-int K.

(iii) First let $X \in \text{Psd}^* + C_1^*$. Then $X = A + tJ$, where $\langle A, Y\rangle \geq 0$ for all Y in Psd. Thus $\langle X, Y\rangle = \langle A, Y\rangle + t\langle J, Y\rangle \geq 0$ for all $Y \in K$, so $X \in K^*$.

Conversely, let $X \in K^*$. Then $X = S + \tilde{S}$, where S is symmetric and $\tilde{S} \in$ Skew. Thus $\langle X, Y\rangle \geq 0$ for all $Y \in K$ is equivalent to $\langle S, Y\rangle \geq 0$ for all $Y \in K$.

Let $u := \frac{e}{\|e\|}$ and U contain an orthonormal basis of $u^\perp$. Thus $U^t U = I_{n-1}, U^t u = 0$. Then the matrix $P = [u\ U]$ is orthogonal. $Y \in K$ implies $Yu = 0$, and therefore

$$P^t Y P = \begin{pmatrix} 0 & 0 \\ 0 & U^t Y U \end{pmatrix} \text{ yielding } \langle S, Y\rangle = \text{tr}(\text{P}^\text{t}\text{SP})(\text{P}^\text{t}\text{YP}) = \text{tr}(\text{U}^\text{t}\text{SU})(\text{U}^\text{t}\text{YU}).$$

Thus $\langle S, Y\rangle \geq 0$ for all $Y \in K$ only if $U^t S U \succeq 0$.

To conclude we show that $U^t S U \succeq 0$ implies $S + tJ \succeq 0$ for some $t \in \Re$. Suppose not. Then, for all $t \in \Re$, $S + tJ$ and therefore

$$R := P^t(S + tJ)P = \begin{pmatrix} u^t S u + tn & u^t S U \\ U^t S u & U^t S U \end{pmatrix} \not\succeq 0.$$

Let $x^t = (\alpha\ a^t)$ be a unit vector such that $x^t R x < 0$. Then

$$0 > x^t R x = \alpha^2(u^t S u + tn) + 2\alpha u^t S U a + a^t U^t S U a.$$

Clearly $\alpha \neq 0$, so without loss of generality $\alpha > 0$ and $\alpha^2 + a^t a = 1$. The inequality is true only if $tn + u^t S u + \frac{2}{\alpha} u^t S U a < 0$ which in turn implies $t < \frac{1}{n}(\frac{2}{\alpha}\|u^t S U\| - u^t S u)$.

So for each $\alpha > 0$ the possible values for t, such that $x^t Rx < 0$ are bounded from above, a contradiction. Therefore $U^t SU \succeq 0$ implies $S + tJ \succeq 0$ for some $t \in \Re$.

Thus $X \in K^*$ implies $X = S + \tilde{S} = (S + tJ + \tilde{S}) + (-tJ) \in \mathrm{Psd}^* + C_1^*$. □

As a final tool we derive a max-min relationship for the following optimization problem in two real variables s and t. Let M be a (given) symmetric matrix of size $n \times n$, $n \geq 2$, and let k be a constant satisfying $0 < k < n$ (k not necessarily integer).

$$\max\ ns - k^2 t, \tag{13}$$

$$M + tJ - sI \succeq 0, \tag{14}$$

$$s, t \in \Re. \tag{15}$$

LEMMA 2.3. *The maximum in* (13) *is attained for some* $\bar{s}, \bar{t} \in \Re$. *Moreover,* $\bar{s} = \lambda_{\min}(M + \bar{t}J)$.

Proof. Clearly, $s \leq \lambda_{\min}(M + tJ)$ for any feasible s and t due to (14). Hence we may restrict our choice to $s = \lambda_{\min}(M + tJ)$, since increasing s increases the value of the objective function as well. Let us denote $\rho = \rho(M)$ the spectral radius of M, and $\lambda_0 = \lambda_{\min}(M)$ the minimum eigenvalue of M. Let us set $f(t) := n\lambda_{\min}(M + tJ) - k^2 t$. We claim that $f(t) \geq f(0) = n\lambda_0$ only for $t \in [-\frac{n(\rho - \lambda_0)}{n^2 - k^2}, \frac{n(\rho - \lambda_0)}{k^2}]$.

(i) *Lower bound.* Take $e = (1, \ldots, 1)^t$. Then $\lambda_{\min}(M + tJ) \leq \frac{e^t(M+tJ)e}{e^t e} \leq \rho(M) + tn$. Assume that $f(t) \geq f(0)$, i.e., $n(\rho + tn) - k^2 t \geq n\lambda_0$, which yields $t \geq -\frac{n(\rho - \lambda_0)}{n^2 - k^2}$.

(ii) *Upper bound.* Take $x = (1, -1, 0, \ldots, 0)^t$. (Here the assumption $n \geq 2$ is used.) Then $\lambda_{\min}(M + tJ) \leq \frac{x^t(M+tJ)x}{x^t x} \leq \rho(M)$. Assume that $f(t) \geq f(0)$, i.e., $n\rho - k^2 t \geq n\lambda_0$, which yields $t \leq \frac{n(\rho - \lambda_0)}{k^2}$.

Since $f(t) \geq f(0)$ may hold only for t belonging to a closed interval containing zero, and the objective function (13) is continuous, the maximum is attained for some $\bar{t}$ and $\bar{s} = \lambda_{\min}(M + \bar{t}J)$. □

The following lemma is crucial for the proof of Theorem 2.1. We recall that $\mathcal{X}_k := \{x \in \Re^n \mid \|x\| = 1 \text{ and } \sum_{i=1}^n x_i = \frac{k}{\sqrt{n}}\}$

LEMMA 2.4. *Let* $\bar{s}$ *and* $\bar{t}$ *be the optimum for the program* (13)–(15). *Then* $\mathcal{X}_k \cap \mathrm{Eig}(\lambda_{\min}(M + \bar{t}J - \bar{s}I))$ *is nonempty.*

Proof. Let us denote $M(s,t) := M + tJ - sI$, and set

$$f(s,t) := ns - k^2 t$$

and

$$g(s,t) := \lambda_{\min}(M(s,t)).$$

Hence, the problem (13)–(15) can be rewritten as

$$\max f(s,t) \tag{16}$$

$$g(s,t) \geq 0. \tag{17}$$

Let $\bar{s}$ and $\bar{t}$ denote the optimum solution, which exists by Lemma 2.3. We distinguish three cases.

Case (i). Assume that the eigenvalue $\lambda_{\min}(M(\bar{s}, \bar{t}))$ is simple.

Using, e.g., Theorem 2 of [7], we get that the function $g(s,t)$ is differentiable at the point $(\bar{s},\bar{t})$, and its partial derivatives are given by

$$\frac{\partial g(\bar{s},\bar{t})}{\partial s} = -\sum_{i=1}^{n} \bar{x}_i^2 = -1,$$

$$\frac{\partial g(\bar{s},\bar{t})}{\partial t} = \left(\sum_{i=1}^{n} \bar{x}_i\right)^2,$$

where $\bar{x} = (\bar{x}_i)$ is the eigenvector of $\lambda_{\min} M(\bar{s},\bar{t})$ of norm one. The partial derivatives of the objective function f are

$$\frac{\partial f(\bar{s},\bar{t})}{\partial s} = n,$$

$$\frac{\partial f(\bar{s},\bar{t})}{\partial t} = -k^2.$$

Using the Kuhn–Tucker optimality condition, by which

$$\nabla f(\bar{s},\bar{t}) + \alpha \nabla g(\bar{s},\bar{t}) = 0$$

for some $\alpha \in \Re$, we get $\alpha = n$, and hence

$$\left(\sum_{i=1}^{n} \bar{x}_i\right)^2 = \frac{k^2}{n}.$$

Case (ii). Assume that the eigenvalue $\lambda_{\min} M(\bar{s},\bar{t})$ is multiple, and that there exists an eigenvector x such that $\|x\|^2 = 1$ and $\sum_{i=1}^{n} x_i > \frac{k}{\sqrt{n}}$. Clearly, $y := -x$ is an eigenvector of unit length satisfying $\sum_{i=1}^{n} y_i < -\frac{k}{\sqrt{n}}$. Since $\{x \in \Re^n \mid \|x\| = 1\} \cap \mathrm{Eig}(\lambda_{\min}(M + \bar{t}J - \bar{s}I))$ is compact and connected, we conclude that there exists an eigenvector $\bar{x}$ of norm one satisfying $\sum_{i=1}^{n} \bar{x}_i = \frac{k}{\sqrt{n}}$.

Case (iii). Assume that neither case (i) nor case (ii) occur. We will derive a contradiction. Let $\mathcal{E}$ denote the eigenspace of $\lambda_{\min} M(\bar{s},\bar{t})$, $\mathcal{S} = \{x \mid \|x\| = 1\}$, and $\mathcal{E}^1 = \mathcal{E} \cap \mathcal{S}$. Set $\alpha := \frac{k^2}{n}$, and $\tilde{\gamma} := \max_{x \in \mathcal{E}^1} x^t J x$. If neither case (i) nor case (ii) occur, then $\tilde{\gamma} < \alpha$. Let γ be chosen so that $\tilde{\gamma} < \gamma < \alpha$. Now consider

$$\nu := \min\{x^t(M + \bar{t}J - \bar{s}I)x : \ x^t J x \geq \gamma, x^t x = 1\}.$$

The feasible set is compact and nonempty so the minimum indeed exists. (The feasible set is nonempty, since $x^t J x = n > \frac{k^2}{n} = \alpha > \gamma$ for $x = \frac{e}{\sqrt{n}}$.)

The choice of $\bar{s},\bar{t}$ makes $\bar{M} := M + \bar{t}J - \bar{s}I$ positive semidefinite, and the choice of γ ensures $\nu > 0$. Let

$$\epsilon := \frac{\nu}{n - \gamma}, \ \tilde{s} := \bar{s} - \epsilon\gamma, \ \tilde{t} := \bar{t} - \epsilon.$$

We show that

$$\tilde{M} := M + \tilde{t}J - \tilde{s}I = \bar{M} + \epsilon(\gamma I - J)$$

is positive semidefinite, so that $\tilde{s}, \tilde{t}$ are feasible for problem (13)–(15). We distinguish two cases. Suppose $x^t x = 1, x^t J x \geq \gamma$. Then, using $x^t J x \leq n$, $x^t \bar{M} x \geq \nu$, we get $x^t \tilde{M} x \geq \nu - \frac{\nu}{\gamma - n}(\gamma - n) = 0$. Suppose $x^t x = 1, x^t J x \leq \gamma$. Then $x^t \tilde{M} x \geq x^t \bar{M} x \geq 0$. Finally $f(\tilde{s}, \tilde{t}) = f(\bar{s}, \bar{t}) + \epsilon(\alpha - \gamma) > f(\bar{s}, \bar{t})$, contradicting optimality of $\bar{s}, \bar{t}$. □

Let us remark that the proof of part (iii) in Lemma 2.4 is due to Ch. Helmberg, and it simplifies our previous more complicated arguments.

THEOREM 2.1. *It holds that*

$$\max\left\{s - \frac{k^2}{n}t : M + tJ - sI \succeq 0\right\} = \min\{x^t M x : x \in \mathcal{X}_k\}.$$

Proof. Let $\bar{s}, \bar{t}$ be the optimum of (13)–(15).

(i) Let $\bar{x} \in \mathcal{X}_k \cap \mathrm{Eig}(\lambda_{\min}(M + \bar{t}J - \bar{s}I))$. We have

$$\begin{aligned} 0 &= \lambda_{\min}(M + \bar{t}J - \bar{s}I) = \lambda_{\min}(M + \bar{t}J) - \bar{s} \\ &= \bar{x}^t(M + \bar{t}J)\bar{x} - \bar{s} = \bar{x}^t M \bar{x} + \bar{t}\left(\sum_{i=1}^n \bar{x}_i\right)^2 - \bar{s} \\ &= \bar{x}^t M \bar{x} + \frac{\bar{t}k^2}{n} - \bar{s} \geq \frac{\bar{t}k^2}{n} - \bar{s} + \min_{x \in \mathcal{X}_k} x^t M x. \end{aligned}$$

Hence

$$\min_{x \in \mathcal{X}_k} x^t M x \leq \bar{s} - \frac{\bar{t}k^2}{n}.$$

(ii) Let $\tilde{x} \in \mathcal{X}_k$ be such that $\tilde{x}^t M \tilde{x} = \min_{x \in \mathcal{X}_k} x^t M x$. We have

$$\tilde{x}^t M \tilde{x} + \frac{\bar{t}k^2}{n} = \tilde{x}^t(M + \bar{t}J)\tilde{x} \geq \lambda_{\min}(M + \bar{t}J) = \bar{s}.$$

Hence $\tilde{x}^t M \tilde{x} \geq \bar{s} - \frac{\bar{t}k^2}{n}$, which proves the opposite inequality. □

3. Max-cut problem. The *node-relaxation* $\varphi(G, c)$ for the max-cut problem was introduced in [5] as

$$\varphi(G, c) := \min_{u \in \mathcal{U}} \frac{n}{4}\lambda_{\max}(L + \mathrm{diag}(u)). \tag{18}$$

Let us note that (18) is nothing else but

$$\varphi(G, c) := \min_{u \in \mathcal{U}} \frac{n}{4} \max_{\|x\|=1} x^t(L + \mathrm{diag}(u))x \tag{19}$$

when using the Rayleigh quotient to express $\lambda_{\max}$.

The (positive semidefinite) *edge-relaxation* $\psi(G, c)$ of the max-cut is based on (10).

$$\psi(G, c) := \max\left\{\sum_{1 \leq i < j \leq n} c_{ij} y_{ij} \mid \frac{1}{2}J - Y \succeq 0\right\}, \tag{20}$$

where $y = (y_{ij}) \in \Re^{\binom{n}{2}}$ is a vector of variables, and Y is the corresponding $n \times n$ matrix defined as before so that

$$Y_{ij} := \begin{cases} y_{ij} & i < j, \\ y_{ji} & i > j, \\ 0 & i = j. \end{cases}$$

Using the notions of node- and edge-incidence vectors defined by (1) and (2), and the property (10) (which shows that $\frac{1}{2}J - Y \succeq 0$), it is easy to obtain Lemma 3.1.

LEMMA 3.1. *Both $\varphi(G,c)$ and $\psi(G,c)$ are upper bounds on $mc(G,c)$.*

We will now show that the two relaxations form a pair of dual programs, satisfying strong duality, thus φ and ψ define the same bound.

THEOREM 3.1. *Let (G,c) be a weighted graph. Then $\varphi(G,c) = \psi(G,c)$.*

Proof. Let us first rewrite the definition of the bound $\psi(G,c)$ in a form that allows dualization by Lemma 2.1. Let M be the $n^2 \times \binom{n}{2}$ matrix defined such that $w = My$ is the symmetric matrix corresponding to y with main diagonal zero and written as a vector in $\Re^{n^2}$. In other words, the entry $M_{(i,j),\{k,\ell\}}$ (lying in the (i,j)th row and the $\{k,\ell\}$th column, which are indexed by ordered and unordered pairs, respectively) is defined by

$$M_{(i,j),\{k,\ell\}} := \begin{cases} 1 & \text{if } \{i,j\} = \{k,\ell\}, \\ 0 & \text{otherwise.} \end{cases}$$

We denote by $j \in \Re^{n^2}$ the matrix J written as a vector. We have (identifying vectors of size n^2 with square matrices)

$$\psi(G,c) = \max c^t y, \tag{21}$$

$$\frac{1}{2}j - My \in \text{Psd}, \tag{22}$$

$$y \in \Re^{\binom{n}{2}}. \tag{23}$$

Since (21)–(23) satisfies the generalized Slater condition (consider $y = (\frac{1}{2}, \ldots, \frac{1}{2})$), the strong duality of Lemma 2.1 can be applied. The dual of (21)–(23) reads

$$\psi(G,c) = \min \frac{1}{2} j^t x, \tag{24}$$

$$M^t x - c = 0, \tag{25}$$

$$x \in \text{Psd}^*. \tag{26}$$

Without loss of generality, we may assume that x corresponds to a symmetric matrix, because $j^t z = 0$ for all $z \in \text{Skew}$. Hence (26) can be replaced by $x \in \text{Psd}$.

Equivalently, (24)–(26) can be formulated as

$$\psi(G,c) = \min \frac{1}{2} \sum_{1 \le i,j \le n} x_{ij}, \tag{27}$$

$$x_{ij} + x_{ji} - c_{ij} = 0, \quad i < j, \tag{28}$$

$$x \in \text{Psd}. \tag{29}$$

Since all entries x_{ij} but x_{ii} are determined by (28) as $x_{ij} = \frac{1}{2}c_{ij}$, the actual variables are only x_{ii}. Let us express them in the form

$$2x_{ii} = -d_i - u_i + s, \tag{30}$$

where d_i, $i = 1, \ldots, n$, is the weighted degree, and $s := \frac{1}{n}\sum_{i=1}^{n}(2x_{ii} + d_i)$ and $u_i = -d_i + s - 2x_{ii}$. Clearly, $u \in \mathcal{U}$. The constraint (29) is equivalent to

$$C - \mathrm{diag}(d) - \mathrm{diag}(u) + sI \succeq 0$$

or, equivalently, to

$$s \geq \lambda_{\max}(L + \mathrm{diag}(u)) \tag{31}$$

since $L = \mathrm{diag}(d) - C$. The objective function (27) becomes, after substituting (30)

$$\frac{1}{2}\sum_{i\neq j} x_{ij} + \frac{1}{2}\sum_{i=1}^{n} x_{ii} = \frac{1}{4}\sum_{i\neq j} c_{ij} + \frac{1}{4}\sum_{i=1}^{n}(-d_i - u_i + s) = \frac{n}{4}s \tag{32}$$

since $\sum u_i = 0$ and $\sum_{i\neq j} c_{ij} = \sum_{i=1}^{n} d_i$. Now it is easy to see that (24) and (25) are equivalent to (18). □

We point out that A. Schrijver [19] proposed formulation (20) with Y instead of y to derive tractable relaxations of the max-cut problem. This formulation of Schrijver was in fact the motivation for us to study the more general bisection problems described in the present paper.

In the rest of this section, we show the way in which the optimum solution Y of (20) is related to the optimality certificate formulated in [5].

Let (G, c) be a weighted graph and $u \in \mathcal{U}$. Let $z_1 = (z_{i1}), \ldots, z_m = (z_{im}) \in \mathrm{Eig}(\lambda_{\max}(L + \mathrm{diag}(u)))$ be a collection of m eigenvectors. We say that the collection $z_1, \ldots, z_m$ is an *optimality certificate for* u, if

$$\sum_{j=1}^{m} z_{ij}^2 = 1 \quad \text{for every } i = 1, \ldots, n. \tag{33}$$

Let us say that $u^* \in \mathcal{U}$ is an *optimum correcting vector* for (G, c) if $\varphi(G, c) = \frac{n}{4}\lambda_{\max}(L + \mathrm{diag}(u^*))$.

THEOREM 3.2. [5] *Let $u \in \mathcal{U}$. Then u is the optimum correcting vector if and only if there exists an optimality certificate for u.*

The next theorem describes the mutual relation between an optimal Y in the edge-relaxation (20) and an optimality certificate.

THEOREM 3.3. *Let (G, c) be a weighted graph, and $u \in \mathcal{U}$ be the optimum correcting vector.*

(i) *Let $z_1, \ldots, z_m$ be an optimality certificate for u. Let $Z = [z_1, \ldots, z_m]$ be the $n \times m$ matrix with columns $z_1, \ldots, z_m$. Then $Y := \frac{1}{2}(J - ZZ^t)$ is the optimum for (20).*

(ii) *Let Y be the optimum for (20). Then there exists a collection of pairwise orthogonal vectors $z_1, \ldots, z_m$ satisfying*

$$J - 2Y = z_1 z_1^t + \cdots + z_m z_m^t, \tag{34}$$

which form an optimality certificate for u.

Proof. (i) The matrix Y has zero diagonal, since ZZ^t has all ones on the main diagonal due to (33). The matrix $\frac{1}{2}J - Y$ is positive semidefinite, since $\frac{1}{2}J - Y = ZZ^t$. Thus, Y is feasible for (20). It remains to show the optimality of Y.

Let $U = \mathrm{diag}(u)$. We have

$$\mathrm{tr}((\mathrm{L} + \mathrm{U})\mathrm{ZZ}^{\mathrm{t}}) = \mathrm{n}\lambda_{\max} \tag{35}$$

since the columns of Z are eigenvectors of $\lambda_{\max}(L+U)$, and ZZ^t has all ones on the main diagonal. Furthermore, we have

$$\text{tr}(\text{CJ}) - \text{tr}((\text{D}+\text{U})\text{ZZ}^{\text{t}}) = \sum_{\text{i}\neq\text{j}} \text{c}_{\text{ij}} - \sum_{\text{i}=1}^{\text{n}} \text{d}_{\text{i}} = 0\ . \tag{36}$$

Using (35), (36), and the identities $L = D - C$ and $C = (C - D - U) + (D + U)$, we get

$$\begin{aligned}\text{tr}(2\text{CY}) &= \text{tr}(\text{C}(\text{J} - \text{ZZ}^{\text{t}})) = \text{tr}(\text{CJ}) - \text{tr}(\text{CZZ}^{\text{t}}) \\ &= \text{tr}(\text{CJ}) - \text{tr}((\text{C} - \text{D} - \text{U})\text{ZZ}^{\text{t}}) - \text{tr}((\text{D} + \text{U})\text{ZZ}^{\text{t}}) \\ &= \text{tr}((\text{L} + \text{U})\text{ZZ}^{\text{t}}) = \text{n}\lambda_{\max}\ .\end{aligned}$$

(ii) Using the complementary slackness (cf. part (iii) of Lemma 2.1) for (22) and (29), we have

$$x^t\left(\frac{1}{2}j - My\right) = 0 \tag{37}$$

or equivalently

$$\text{tr}\left(\text{X}^{\text{t}}\left(\frac{1}{2}\text{J} - \text{Y}\right)\right) = 0 \tag{38}$$

when rewritten back in the notation using matrices. Since $J - 2Y$ is positive semidefinite by (22), there exist $z_1, \ldots, z_m$ such that (34) holds.

Claim. The vectors $z_1, \ldots, z_m$ are eigenvectors of $\lambda_{\min}(X)$, and $\lambda_{\min}(X) = 0$.

Let x_{ij} denote the entries of X, and $z_{i\ell}$ the entries of z_ℓ, $\ell = 1, \ldots, m$, respectively. When substituting (34) into (38), we get

$$0 = \text{tr}\left(\text{X}^{\text{t}}\left(\frac{1}{2}\text{J} - \text{Y}\right)\right) = \sum_{\text{i,j}} \text{x}_{\text{ij}}\left(\sum_{\ell=1}^{\text{m}} \text{z}_{\text{i}\ell}\text{z}_{\text{j}\ell}\right) = \sum_{\ell=1}^{\text{m}} \text{z}_\ell^{\text{t}}\text{Xz}_\ell. \tag{39}$$

Since $X \succeq 0$, we have $z_\ell^t X z_\ell \geq 0$ for every ℓ, and hence $z_\ell^t X z_\ell = 0$ for every ℓ, since the sum (39) of these nonnegative terms is zero. This proves that $z_1, \ldots, z_m$ are eigenvectors of X corresponding to eigenvalue zero, which is the minimum eigenvalue of X. This proves the claim.

Since the constraints (29) and (31) are equivalent (they differ only by a diagonal shift), we conclude that $z_1, \ldots, z_m$ are eigenvectors of $\lambda_{\max}(L+\text{diag}(u))$. The property (33) follows immediately from (34). □

COROLLARY 3.1. *There always exists an optimality certificate $z_1, \ldots, z_m$, $m \leq t$, where the vectors z_i are pairwise orthogonal and t is the dimension of the eigenspace* $\text{Eig}(\lambda_{\max}(L+U))$.

Let us remark that m can be substantially smaller than t. In particular, $m = 1$ if $\text{Eig}(\lambda_{\max}(L+U)) \cap \{-1, 1\}^n \neq \emptyset$, i.e., the eigenspace contains a ± 1-vector x. Then x determines a cut of size $\varphi(G, c)$, and hence $mc(G, c) = \varphi(G, c)$. However, it is NP-hard to determine the minimum size of an optimality certificate [6].

The optimality certificate from [5] recalled in Theorem 3.2 is related to an optimality criterion of Overton, formulated in [12] for a more general problem. We will rephrase this criterion.

Let (G,c) be a weighted graph, and $u \in \mathcal{U}$. Let $q_1,\ldots,q_t$ be an orthonormal basis of the eigenspace $\mathrm{Eig}(\lambda_{\max}(L+U))$, where t denotes the dimension of the eigenspace. Let $r_1,\ldots,r_n$ denote the rows of the matrix $Q=[q_1,\ldots,q_t]$.

THEOREM 3.4. (Overton's criterion) *The vector u is an optimum correcting vector for (G,c) if and only if there exists a $t\times t$ positive semidefinite matrix W, such that $r_i^t W r_i = 1$ for every $i=1,\ldots,n$.*

The connection between Overton's criterion and the optimality certificate of Theorem 3.2 is the following.

Let $\mathcal{E}=\mathrm{Eig}(\lambda_{\max}(L+U))$. Let $z_1,\ldots,z_m \in \mathcal{E}$ be an optimality certificate, and $q_1,\ldots,q_t \in \mathcal{E}$ be an orthonormal basis of $\mathcal{E}$. Let Δ denote the $m\times t$ matrix $\Delta=(\delta_{ij})$ of coefficients such that

$$z_\ell = \sum_{j=1}^{t} \delta_{\ell j} q_j \tag{40}$$

for $\ell=1,\ldots,m$. Let $W=(w_{ij})$ be the $t\times t$ matrix defined by

$$w_{ij} = \sum_{\ell=1}^{m} \delta_{\ell i}\delta_{\ell j}. \tag{41}$$

THEOREM 3.5. *The matrix W defined by* (41) *provides Overton's criterion formulated in Theorem* 3.4. *Conversely, let W be a positive semidefinite matrix given by Theorem* 3.4, *and $\Delta=(\delta_{ij})$ be a $t\times t$ matrix such that*

$$W = \Delta^t\Delta\ . \tag{42}$$

(*The existence of Δ follows from the fact that W is positive semidefinite.*) *Then $z_1,\ldots,z_t$ defined by* (40) *constitute an optimality certificate.*

Proof. Let $q_1,\ldots,q_t$ be an orthonormal basis of the eigenspace $\mathcal{E}$. Given an optimality certificate $z_1,\ldots,z_m$, let W be defined by (41). We have, for every $i=1,\ldots,n$,

$$\begin{aligned} r_i^t W r_i &= \sum_{j,k} w_{jk} q_{ij} q_{ik} = \sum_{j,k}\left(\sum_{\ell=1}^{m}\delta_{\ell j}\delta_{\ell k}\right) q_{ij}q_{ik} \\ &= \sum_{\ell=1}^{m}\sum_{j,k}\delta_{\ell j}\delta_{\ell k}q_{ij}q_{ik} = \sum_{\ell=1}^{m}\left(\sum_{j=1}^{t}\delta_{\ell j}q_{ij}\right)\left(\sum_{k=1}^{t}\delta_{\ell k}q_{ik}\right) = \sum_{\ell=1}^{m} z_{i\ell}^2 = 1, \end{aligned}$$

which proves that W provides Overton's optimality criterion. Reversing the arguments proves the converse. □

Remark 3.1. Let Q and W be as in Theorem 3.4. Then $Y=\frac{1}{2}(J-QWQ^t)$ is the optimum Y for the definition of $\psi(G,c)$ in (20).

4. Graph bisection into fixed sizes. Rendl and Wolkowicz [17] introduced in an equivalent form (see the remark at the end of this section) the following lower bound $\nu(G,c,k)$ on the bisection width, which we call the *node relaxation*. Let

$$\nu(G,c,k) := \max_{u\in\mathcal{U}} \min_{x\in\mathcal{X}_k} \frac{n}{4} x^t(L+\mathrm{diag}(u))x, \tag{43}$$

where $\mathcal{U}$, $\mathcal{X}_k$, and the Laplacian matrix $L=L(G,c)$ were defined above.

LEMMA 4.1. [17] *Let (G,c) be a weighted graph. Then $\nu(G,c,k) \leq b(G,c,k)$ for every admissible k.*

Proof. Let $S \subset V, |S| = \frac{1}{2}(n-k)$, be such that $b(G,c,k) = \sum_{i \in S, j \notin S} c_{ij}$. Let $\overline{x}$ be the node-incidence vector of the bipartition $(S, V \setminus S)$ given by (1). Using (9), we have

$$\sum_{i \in S, j \notin S} c_{ij} = \frac{1}{4} \sum_{i \in S, j \notin S} c_{ij}(\overline{x}_i - \overline{x}_j)^2 = \frac{1}{4} \overline{x}^t L \overline{x} \ .$$

Given $u \in \mathcal{U}$, we have $\sum_{i=1}^n u_i \overline{x}_i^2 = \sum_{i=1}^n u_i = 0$ since $\overline{x}$ is a ± 1-vector, and $\|\overline{x}\| = \sqrt{n}$. Hence

$$b(G,c,k) = \frac{1}{4} \overline{x}^t (L + \mathrm{diag}(u)) \overline{x} \geq \nu(G,c,k) \ . \qquad \square$$

Our main result is again that $\nu(G,c,k)$ can alternatively be obtained as the optimum of another optimization problem. The importance of the result consists in the fact that our new formulations allow adding further constraints that have already proved to be useful in the approximation of graph partition problems. We postpone the more detailed discussion to the last section.

We introduce the positive semidefinite *edge relaxation* $\eta(G,c,k)$ as the optimum value of the following semidefinite linear program.

$$\eta(G,c,k) := \quad \min c^t y, \tag{44}$$

$$\frac{1}{2} J - Y \succeq 0, \tag{45}$$

$$e^t y = m, \tag{46}$$

where

$$m := n_1 n_2 = \frac{1}{4}(n^2 - k^2). \tag{47}$$

The following lemma shows that $\eta(G,c,k)$ is indeed a lower bound on the bisection width.

LEMMA 4.2. *Let (G,c) be a weighted graph. Then $\eta(G,c,k) \leq b(G,c,k)$ for every admissible k.*

Proof. It is useful to present a direct proof although the statement is a consequence of Lemma 4.1 and Theorem 4.1.

Let $S \subset V, |S| = \frac{1}{2}(n-k)$, be such that $b(G,c,k) = \sum_{i \in S, j \notin S} c_{ij}$. Let y be the edge-incidence vector of the bipartition $(S, V \setminus S)$ given by (2), and Y be the corresponding matrix. Then

$$\frac{1}{2} J - Y \succeq 0$$

by (10) and y clearly satisfies (46). Thus, y is a feasible solution of the above program, and hence $\eta(G,c,k) \leq b(G,c,k)$. $\square$

In §6.4 the condition (45), which is equivalent to (5), is further strengthened. It is also useful to mention that the constraint (45) implies that $0 \leq y \leq 1$ by considering 2×2 submatrices of Y.

We now formulate our main theorem.

THEOREM 4.1. *Let (G,c) be a weighted graph. Then $\nu(G,c,k) = \eta(G,c,k)$ for any admissible k.*

Proof. We first formulate the program (44)–(46) which defines the bound $\eta(G,c,k)$ as a program of the form (11). We set $\Omega := \Re^{\binom{n}{2}}$ and $S := \{(w,0) \in \Re^{n^2+1} \mid w \in \text{Psd}\}$. Then clearly $\Omega^* = \{0\} \subset \Re^{\binom{n}{2}}$ and $S^* = \{(x,t) \mid x \in \text{Psd}^*, t \in \Re\}$. We recall the definition of the operator M from the proof of Theorem 3.1, that maps $y \in \Re^{\binom{n}{2}}$ to the corresponding symmetric matrix with main diagonal zero, written as vector.

The program (44)–(46) is equivalent to the following problem.

$$\eta(G,c,k) = \min c^t y, \tag{48}$$

$$-My + \frac{1}{2}j \in \text{Psd}, \tag{49}$$

$$e^t y - m = 0, \tag{50}$$

$$y \in \Omega. \tag{51}$$

We must distinguish two cases according to whether $k > 0$ or $k = 0$. It will be shown that these two cases correspond to whether or not problem (49)–(51) satisfies the generalized Slater condition.

Case (i). $k > 0$. We claim that (49)–(51) satisfy the generalized Slater condition.

In order to prove the claim, let us define the matrix $\bar{Y}$ as a matrix with the zero diagonal, and all off-diagonal entries equal $p := \frac{n^2-k^2}{2n(n-1)}$. With this choice of $\bar{Y}$, we have $\sum_{i<j} y_{ij} = m$, as required by (50). Let us consider $\frac{1}{2}J - \bar{Y} = (\frac{1}{2} - p)J + pI$. The eigenvalues of this matrix are (cf. part (ii) of Lemma 2.2) $(p, p, \ldots, p, \frac{n}{2} - np + p)$, and they are all positive due to the definition of p. Thus the matrix $\frac{1}{2}J - \bar{Y}$ is positive definite, and hence it lies in the relative interior of the cone of positive semidefinite matrices.

The dual of (48)–(51) now reads as follows.

$$\max\ mt - \frac{1}{2}\sum_{1 \le i,j \le n} x_{ij}, \tag{52}$$

$$c_{ij} + x_{ij} + x_{ji} - t = 0 \quad i < j, \tag{53}$$

$$x \in \text{Psd}^*, \tag{54}$$

$$t \in \Re. \tag{55}$$

Here x is the vector of the variables dual to the constraint (45), and t is the variable dual to the constraint (46).

Case (ii). $k = 0$. In this case, we have $e^t(\frac{1}{2}J - Y)e = 0$ if and only if Y satisfying (46). Hence no $\frac{1}{2}J - Y$ is positive definite, and the generalized Slater condition is not satisfied. Let K denote the cone of positive semidefinite matrices A satisfying $e^t A e = 0$, which is studied in Lemma 2.2. The program (48)–(51) can be replaced by

$$\eta(G,c,0) = \min c^t y, \tag{56}$$

$$-My + \frac{1}{2}j \in K, \tag{57}$$

$$y \in \Omega. \tag{58}$$

Now we will apply Lemma 2.2. By part (i), K is a closed convex cone. By part (ii), the problem (56)–(58) satisfies the generalized Slater condition.

Using Lemma 2.1, the dual problem of (56)–(58) reads

$$\max \; -\frac{1}{2}\sum_{1\le i,j\le n} w_{ij}, \tag{59}$$

$$c_{ij} + w_{ij} + w_{ji} = 0 \; i < j, \tag{60}$$

$$W \in K^*, \tag{61}$$

where $W = (w_{ij})$ is a symmetric matrix of dual variables. Using the characterization of K^* given in part (iii) of Lemma 2.2, we can write

$$W = X - \frac{1}{2}tJ \tag{62}$$

for some t. After this substitution, (59)–(61) turn into the form (52)–(55) with $m = \frac{1}{4}n^2$.

Thus, we have shown that independently of whether $k = 0$ or $k > 0$, the dual of the problem (49)–(51) can be reduced to the form (52)–(55).

The dual program (52)–(55) can be further simplified by replacing the constraint (54) by

$$x \in \text{Psd} \tag{63}$$

in the program (52)–(55), because, as in the proof of Theorem 3.1, the cost matrix is symmetric, and so Skew does not contribute to the objective function. $x \in$ Psd implies $x_{ij} = x_{ji}$ for all $i \neq j$, thus $x_{ij} = \frac{1}{2}(t - C_{ij})$ for every $i, j, i \neq j$. (Matrix $C = (C_{ij})$ based on c was defined in the introduction.) Let us introduce new variables $z_i, i = 1, \ldots, n$, by

$$z_i := d_i - 2x_{ii} + t, \tag{64}$$

where $d_i := \sum_{j=1}^n C_{ij}$, i.e., d_i is the "weighted degree" of node i. Hence (63) can be replaced by (66). The objective function (52) becomes (using (64), (53), and (47))

$$\begin{aligned}
&mt - \frac{1}{2}\sum_{1\le i,j\le n} x_{ij} \\
&= mt - \sum_{1\le i<j\le n} x_{ij} - \frac{1}{2}\sum_{i=1}^n x_{ii} \\
&= mt - \frac{1}{2}\sum_{1\le i<j\le n} (t - c_{ij}) - \frac{1}{4}\sum_{i=1}^n (d_i - z_i + t) \\
&= mt - \frac{1}{2}t\binom{n}{2} + \frac{1}{2}\sum_{1\le i<j\le n} c_{ij} - \frac{1}{4}\sum_{i=1}^n d_i + \frac{1}{4}\sum_{i=1}^n z_i - \frac{1}{4}tn \\
&= -\frac{1}{4}tk^2 + \frac{1}{4}\sum_{i=1}^n z_i.
\end{aligned}$$

Thus, we can write the dual program as

$$\eta(G, c, k) = \max \frac{1}{4}\sum_{i=1}^n z_i - \frac{1}{4}k^2 t, \tag{65}$$

$$tJ + L - \operatorname{diag}(z) \succeq 0, \tag{66}$$

$$t \in \Re,\; z \in \Re^n, \tag{67}$$

where $L = L(G,c)$ is the Laplacian matrix of (G,c).

We prove that the programs (65)–(67) and (43) defining $\nu(G,c,k)$ provide the same optimum.

Given $z = (z_i)$, set $s := \frac{1}{n}\sum_{i=1}^n z_i$, and $u_i := -z_i + s, i = 1,\dots,n$. Clearly, $u \in \mathcal{U}$ since $\sum u_i = 0$. Program (65)–(67) is equivalent to

$$\max_{u\in\mathcal{U}} \frac{1}{4} f(u),$$

where

$$\begin{aligned} f(u) : = \max\ & ns - k^2 t, \\ & M(u) + tJ - sI \succeq 0, \\ & t, s \in \Re. \end{aligned} \tag{68}$$

Here $M(u) := L + \operatorname{diag}(u)$. Since

$$\nu(G,c,k) := \max_{u\in\mathcal{U}} \min_{x\in\mathcal{X}_k} \frac{n}{4} x^t (L + \operatorname{diag}(u))x,$$

we can apply Theorem 2.1 with matrix $M := M(u)$. The equality

$$\nu(G,c,k) = \eta(G,c,k)$$

follows by taking $M(\overline{u})$ where $\overline{u}$ is the optimum of (43). □

COROLLARY 4.1. *The relaxation $\eta(G,c,k)$ of the bisection width is monotone with respect to k, i.e., $\eta(G,c,k_1) \geq \eta(G,c,k_2)$ for $k_1 < k_2$.*

Proof. Let $\overline{z}$ and $\overline{t}$ be optimum for the program (65) – (67) defining $\eta(G,c,k_2)$. We have

$$\eta(G,c,k_2) = \frac{1}{4}\sum_{i=1}^n \overline{z}_i - \frac{1}{4}k_2^2\overline{t} \leq \frac{1}{4}\sum_{i=1}^n \overline{z}_i - \frac{1}{4}k_1^2\overline{t} \leq \eta(G,c,k_1)\ . \qquad \square$$

Remark 4.1. In the special case of $n_1 = n_2 = n/2$, the relaxation (69) has been first given by R. Boppana [4] (in an equivalent form). Since $k := n_1 - n_2 = 0$, the node relaxation (43) takes the form

$$\nu(G,c,0) := \max_{u\in\mathcal{U}} \min_{x\in\mathcal{X}_0} \frac{n}{4} x^t (L + \operatorname{diag}(u))x. \tag{69}$$

Let us remark that the case with $k = 0$ is somewhat easier than the general case with $k > 0$. One reason is that $\{x \mid e^t x = k\}$ becomes a linear subspace for $k = 0$, and hence $\min_{x\in\mathcal{X}_0} \frac{n}{4} x^t (L + \operatorname{diag}(u))x$ can be reduced to an eigenvalue problem by projecting on this subspace. Another reason is that the proof of Theorem 4.1 can be made much simpler for this case. We sketch some details. The dual definition (65)–(67) of $\eta(G,c,0)$ reads

$$\begin{aligned} \eta(G,c,0) : = \max\ & \frac{1}{4}\sum_{i=1}^n z_i, \\ & tJ + L - \operatorname{diag}(z) \succeq 0, \\ & t \in \Re,\ z \in \Re^n. \end{aligned}$$

The crucial part in the proof of Theorem 4.1 consists of the following claim.

If $\bar{t}$ and $\bar{z}$ are optimal for the above program, then

$$\mathrm{Eig}(\lambda_{\min}(\bar{t}J + L - \mathrm{diag}(\bar{z}))) \cap \mathcal{X}_0 \neq \emptyset\ .$$

Since $k = 0$, case (iii) of Lemma 2.4 is immediately excluded. On the other hand, we needed Lemma 2.2 in order to establish the correctness of the dual problem in case $k = 0$.

Remark 4.2. In [17] the general partitioning problem into subsets of specified sizes is investigated. In the special case of partitioning into just two sets the approach chosen in [17] is equivalent to the relaxation (43). To see the equivalence we note that in [17] a partition (S_1, S_2) with $|S_1| = n_1$ is represented by an $n \times 2$ 0-1-matrix $X = (x_{ij})$ where $x_{ij} = 1$ if and only if $i \in S_j$. The relaxation in [17] for bisection is then obtained by optimizing over all matrices $X = (y\ z)$, satisfying the constraints

$$y^t y = n_1, \quad z^t z = n - n_1, \quad y^t z = 0,$$

$$y + z = e, \quad e^t y = n_1, \quad e^t z = n - n_1.$$

(The first set of constraints describes X as having orthogonal columns of specified lengths, while the second set describes the bisection into specified sizes $n_1, n - n_1$.) Defining $x := y - z$ shows that x normalized to unit length lies in $\mathcal{X}_k$. Conversely taking $x \in \mathcal{X}_k$ we can set $w := \sqrt{n}x$, and $y := (e - w)/2$, $z = (e + w)/2$. Then the matrix $X = (y\ z)$ can easily be shown to satisfy the constraints above. This shows the equivalence of the two approaches.

Remark 4.3. One of the early relaxations of graph partitioning was proposed in [9]. In the case of bisections, this approach amounts to optimizing over all matrices X satisfying just the orthogonality constraints described above. Therefore this relaxation is never better than the relaxation from [17] or (43).

5. Graph bisection under inequality constraints. In this section we will point out how the results of the previous section carry over to the constrained case. We introduce lower bounds $\nu(G, c, \ell, k)$ and $\eta(G, c, \ell, k)$ on $b(G, c, \ell, k)$, which are obtained by relaxation of node and edge incidence vectors, respectively. Let us introduce

$$\nu(G, c, \ell, k) := \max_{u \in \mathcal{U}} \min_{x \in \mathcal{X}_{\ell k}} \frac{n}{4} x^t (L + \mathrm{diag}(u)) x, \tag{70}$$

where

$$\mathcal{U} := \left\{ u \in \Re^n \mid \sum_{i=1}^{n} u_i = 0 \right\}, \tag{71}$$

$$\mathcal{X}_{\ell k} := \left\{ x \in \Re^n \mid \|x\| = 1 \text{ and } \frac{\ell}{\sqrt{n}} \leq \sum_{i=1}^{n} x_i \leq \frac{k}{\sqrt{n}} \right\}, \tag{72}$$

and $L = L(G, c)$ is the Laplacian matrix of the weighted graph (G, c), and

$$\eta(G,c,\ell,k) := \min c^t y, \tag{73}$$

$$\frac{1}{2}J - Y \succeq 0, \tag{74}$$

$$\frac{1}{4}(n^2-k^2) \leq e^t y \leq \frac{1}{4}(n^2-\ell^2), \tag{75}$$

where Y denotes the symmetric matrix with zero diagonal and the off-diagonal entries y_{ij} given by the vector y.

THEOREM 5.1. *Let (G,c) be a weighted graph. Then*

$$\nu(G,c,\ell,k) = \eta(G,c,\ell,k)$$

for any bounds $\ell \leq k$.

Proof. Using the duality given by Lemma 2.1, the dual problem to the definition of $\eta(G,c,\ell,k)$ can be transformed to the following problem.

$$\max \frac{1}{4}\left(-k^2 t_1 + \ell^2 t_2 + \sum_{i=1}^{n} z_i\right), \tag{76}$$

$$(t_1 - t_2)J + L - \operatorname{diag}(z) \succeq 0, \tag{77}$$

$$t_1, t_2 \geq 0, z \in \Re^n. \tag{78}$$

We omit further details of the proof, since it is quite analogous to that of Theorem 4.1. □

6. Computational aspects. Two different practical approaches to the graph partition problems have been pursued so far: (i) "polyhedral" approach, based on solving a linear relaxation, and (ii) "eigenvalue" approach based on optimizing a convex function involving the maximum (or minimum) eigenvalue of a matrix. The theory developed in this paper suggests how to naturally merge these two approaches into a more powerful computational scheme. Let us first recall some details about the individual techniques.

6.1. Polyhedral approach. Let (G,c) be a weighted graph and $x = (x_{ij}) \in \Re^{\binom{n}{2}}$ a vector of variables. Let $\pi(G,c)$ (π for "polyhedral") denote the following bound.

$$\pi(G,c) := \max c^t x, \tag{79}$$

$$x_{ij} + x_{ik} + x_{jk} \leq 2 \text{ for every } i < j < k, \tag{80}$$

$$x_{ij} - x_{ik} - x_{jk} \leq 0 \text{ for every triple } i,j,k. \tag{81}$$

Clearly, $\pi(G,c) \geq mc(G,c)$ because every incidence vector $x = (x_{ij})$ of an edge-cut satisfies (80) and (81). Computational experiments with this bound are reported, e.g., in [3].

Since $\pi(G,c) = mc(G,c)$ for planar graphs G (see [2]), the bound can be expected to behave well on nearly planar instances. On the other hand, it was proved in [15] that the ratio $\frac{\pi(G,c)}{mc(G,c)}$ tends to 2 for a certain class of random graphs (with $c_e = 1$ for $e \in E(G)$).

6.2. Eigenvalue approach. The earliest experiments based on the eigenvalue approach are reported in [9]. However, the bound considered there was not the best possible for the approach. The computational experiments with the bound $\nu(G,c,0)$ (graph bisection into equal sizes) are reported in [10], and the bound $\varphi(G,c)$ on the max-cut problem is computed in [14]. Lower bounds are obtained by rounding a suitable eigenvector to a $\pm 1-$vector, and a consecutive local improvement of the cuts. The approach provides solutions with about 5% relative error between the upper bound and a cut found.

The eigenvalue bound $\varphi(G,c)$ has several interesting properties that resemble the behaviour of the actual value $mc(G,c)$ ([5], [6]). In particular, the ratio $\frac{\varphi(G,c)}{mc(G,c)}$ tends to 1 for random graphs G, i.e., the bound is asymptotically optimal. However, the worst case ratio is not yet known.

The bounds $\nu(G,c,k)$ and $\varphi(G,c)$ can be computed, for arbitrary required precision, in polynomial time by using the ellipsoid method. However, for practical experiments we have used the Bundle Trust algorithm [18] in combination with a Lanczos routine for computing the maximum eigenvalue. Recently, several other methods have been proposed for minimization of the maximum eigenvalue of a parametrized matrix [1], [13], but their practical efficiency has not yet been investigated thoroughly.

6.3. Semi-infinite programs. Our main theorem opens new ways to derive even tighter relaxations of the graph bisection problem, by combining the polyhedral approach relying on a (partial) description of the cut polytope with the semi-infinite edge relaxation introduced in this paper. Specifically, we propose to merge the above approaches in the following semi-infinite program, which presents a lower bound on the bisection $b(G,c,k)$.

$$\min c^t x, \tag{82}$$

$$\sum_{1\le i<j\le n} b_i b_j x_{ij} \le \frac{1}{4}\left(\sum_{i=1}^{n} b_i\right)^2 \quad \text{for every } b=(b_1,\dots,b_n)\in \Re^n, \tag{83}$$

$$x_{ij}+x_{ik}+x_{jk}\le 2 \quad \text{for every } i,j,k, \tag{84}$$

$$x_{ij}-x_{ik}-x_{jk}\le 0 \quad \text{for every } i,j,k, \tag{85}$$

$$\sum_{1\le i<j\le n} x_{ij} = \frac{n^2-k^2}{4}. \tag{86}$$

The program (82),(83), and (86) computes $\eta(G,c,k)$ since constraints (83) are equivalent to $\frac{1}{2}J - X \succeq 0$. Let us also remark that (83) can be efficiently tested by computing the minimum eigenvalue of $\frac{1}{2}J - X$, and if the eigenvalue is negative, then its eigenvector presents a b for which the constraint is violated.

6.4. Hypermetric and gap inequalities. For a particular choice of a vector b, a class of stronger inequalities than (83) can be considered. Let $b_1,\dots,b_n$ be integers with $\sum_{i=1}^n b_i$ odd. Let us consider the following inequality

$$\sum_{1\le i<j\le n} b_i b_j x_{ij} \le \frac{1}{4}\left(\left(\sum_{i=1}^{n} b_i\right)^2 - 1\right). \tag{87}$$

It is not difficult to see that inequalities (87) are valid for the edge-cut incidence vectors. (*Proof.* Let $x=(x_{ij})$ be the edge-cut incidence vector of a partition $(S, V\backslash S)$.

Then, say,

$$\sum_{i\in S} b_i \le \frac{1}{2}\left(\sum_{i=1}^{n} b_i - 1\right), \text{ and } \sum_{i\notin S} b_i \ge \frac{1}{2}\left(\sum_{i=1}^{n} b_i + 1\right).$$

Hence

$$\sum b_i b_j x_{ij} = \sum_{i\in S\ j\notin S} b_i b_j = \left(\sum_{i\in S} b_i\right)\left(\sum_{j\notin S} b_j\right) \le \frac{1}{4}\left(\left(\sum_{i=1}^{n} b_i\right)^2 - 1\right).$$

A more general class of related inequalites, called *gap inequalities*, was proposed in [11]. Let $b_1,\dots,b_n$ be integers, and γ be the maximum integer such that, for any partition of V into S and $V\setminus S$, the difference between the sums $(\sum_{i\in S} b_i)$ and $(\sum_{j\notin S} b_j)$ is at least γ. (For example $\gamma \ge 1$ if $\sum_{i=1}^{n} b_i$ is odd.) Then the inequality

$$\sum_{1\le i<j\le n} b_i b_j x_{ij} \le \frac{1}{4}\left(\left(\sum_{i=1}^{n} b_i\right)^2 - \gamma\right) \tag{88}$$

is valid for all edge-cut incidence vectors by an argument similar to that above for the inequalities (87).

Clearly, every inequality (87) is dominated by a gap-inequality. However, the class (87) might be easier to handle, since it is NP-complete to determine the gap γ for given $b_1,\dots,b_n$.

The gap-inequalities with $\gamma = 1$ are called the *hypermetric inequalities*, and have been quite intensively studied in the literature. An important theoretical result about the hypermetric inequalities was proved in [8], telling that, for every n, the hypermetric inequalities define a polytope.

The use of hypermetric inequalities in max-cut computation, and a heuristic search for violated ones, was proposed by G. Rinaldi and C. De Simone [16] at the workshop on graph partition problems in Rome 1991.

The class of inequalities (87) contains all hypermetric inequalities. We propose to search for possible violated inequalitites (87) in the "neighbourhood" of the eigenspace of $\lambda_{\min}(\frac{1}{2}J - X)$, but details will be elaborated elsewhere.

Acknowledgments. We thank Monique Laurent, Chun-Wa Ko, and Lex Schrijver for a discussion and helpful comments. In particular, we are indebted to Henry Wolkowicz for discussing the use of duality and to Christoph Helmberg for a simple proof of Lemma 2.4. Finally we thank an anonymous referee for the careful review of the paper and the constructive comments.

Svata Poljak died on April 2, 1995. *I regret deeply the loss of a teacher and good friend to whom I owe very much.*

Franz Rendl

REFERENCES

[1] F. ALIZADEH, *Optimization over the positive semi-definite cone: Interior point methods and combinatorial applications*, Proc. IPCO 1992, pp. 385–405.

[2] F. BARAHONA, *The max-cut problem in graphs not contractible to* K_5, Oper. Res. Lett., 2 (1983), pp. 107–111.

[3] F. BARAHONA, M. GRÖTSCHEL, M. JÜNGER, AND G. REINELT, *An application of combinatorial optimization to statistical physics and circuit layout design*, Oper. Res., 36 (1988), pp. 493–513.
[4] R. B. BOPPANA, *Eigenvalues and graph bisection: an average case analysis*, in Proc. 28th Annual Symposium on Computer Science, IEEE 1987, pp. 280–285.
[5] C. DELORME AND S. POLJAK, *Laplacian eigenvalues and the max-cut problem*, Math. Programming, 63 (1993), pp. 557–574.
[6] ———, *Combinatorial properties and complexity of a max-cut approximation*, Europ. J. Combinatorics, 14 (1993), pp. 313–333.
[7] V. F. DEM'YANOV AND V. N. MALOZEMOV, *Introduction to Minimax*, J. Wiley, New York, 1974.
[8] M. DEZA, M. GRISHUCHIN, AND M. LAURENT, *The hypermetric cone is polyhedral*, Combinatorica, 13 (1993), pp. 397–411.
[9] W. E. DONATH AND A. J. HOFFMAN, *Lower bounds for the partitioning graphs*, IBM J. Research and Developement, 17 (1973), pp. 420–425.
[10] J. FALKNER, F. RENDL, AND H. WOLKOWICZ, *A computational study of graph partitioning*, Math. Programming, 66 (1994), pp. 211–239.
[11] M. LAURENT AND S. POLJAK, *Gap inequalities*, manuscript, 1995.
[12] M. L. OVERTON, *On minimizing the maximum eigenvalue of a symmetric matrix*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 256–268.
[13] ———, *Large-scale optimization of eigenvalues*, SIAM J. Optim., 2 (1992), pp. 88-120.
[14] S. POLJAK AND F. RENDL, *Solving the max-cut problem using eigenvalues*, Research Report No. 91735-OR, Institut für Diskrete Mathematik, University of Bonn, 1991; Discrete Appl. Math., to appear.
[15] S. POLJAK AND Z. TUZA, *On the relative error of the polyhedral approximation of the max-cut problem*, Oper. Res. Lett., 16 (1994), pp. 191–198.
[16] G. RINALDI AND C. DE SIMONE, *A Cutting Plane Algorithm for the Max-Cut Problem*, Technical Report 346, IASI-CNR, Rome, 1992.
[17] F. RENDL AND H. WOLKOWICZ, *A projection technique for partitioning the nodes of a graph*, preprint, 1990; Ann. Oper. Res., to appear.
[18] 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.
[19] A. SCHRIJVER, personal communication, 1992.
[20] H. WOLKOWICZ, *Some applications of optimization in matrix theory*, Linear Algebra Appl., 40 (1981), pp. 101–118.

SIAM J. OPTIMIZATION
Vol. 5, No. 3, pp. 488–505, August 1995

002

FASTER SIMULATED ANNEALING*

BENNETT L. FOX†

Abstract. By cooling slightly more slowly than the canonical schedule and simulating direct self-loop sequences implicitly, the computer time to execute simulated annealing given the number of accepted moves becomes proportional to that number in expectation and, in a certain sense, almost surely. This is generally orders of magnitude faster than naive schemes, while (in contrast to previous work) not implicitly altering the cooling schedule. Running simulated annealing on m independent parallel processors gives, in a certain sense, a further computer-time speedup asymptotically linear in m, under an attractive way of constructing (not entirely local) neighborhoods, given that the computer time is large. Roughly speaking, this happens as the set of optimal states gets hard enough to reach. A pathology of purely local neighborhoods is pointed out.

Key words. simulated annealing, combinatorial optimization, Markov chains, parallel computing

AMS subject classifications. 60J10, 60J27, 90C10

1. Introduction and main results.

This paper has two linked major themes:

1. Generate direct self-loops (of the form $x \to x \to \cdots \to x$) in $O(1)$ *computer time* while leaving the *simulated time* stochastically unchanged.

2. Give conditions that imply (in a certain sense) *linear* (computer time) speedup of simulated annealing when it is executed independently on (say) m processors.

Step 1 makes it practical, for the first time, to use cooling schedules where the temperatures approach zero. In view of step 1, for step 2 it suffices to consider the chain pruned of all direct self-loops. All results extend to hybrids with tabu search and (radically modified) genetic algorithms. Fox [7] details these hybrids. From a mathematical viewpoint, they are (an elaborate version of) simulated annealing—but on a more sophisticated (though still finite) state space, with a tailored (pseudo)objective function and neighborhood structure.

1.1. Computer time on one processor.

We implement step 1 via an algorithm QUICKER, detailed and discussed in §2. Its speed depends on the cooling schedule. The schedule $T_k = \Delta \log(k+1)$, where Δ is (slightly) larger than the maximum difference among objective-function values (or larger than an upper bound on that difference) cools at a *subcanonical* rate. This is slightly slower than the *canonical* schedule, which replaces Δ by the maximum "depth" among local, nonglobal minima (e.g., see Hajek [11]). The (subcanonical) schedule $\tilde{T}_k$ with Δ replaced by the maximum difference among objective-function values falls in between. Section 2.3 discusses these schedules and, more generally, cooling. A PC *schedule* is piecewise constant, bounded away from zero, with all jumps downward. We have *geometric cooling* when the schedule is PC and the ratio of the heights of successive flat pieces is a constant; such schedules are sometimes used in practice, though results of Hajek [11] and others about converging in probability to the set of optimal states then no longer apply. These results and counterparts for almost-sure Cesàro convergence of the proportion of time in optimal states are especially relevant in practice when the objective-function values must be estimated *dynamically*; see Heine [13].

By QUICKER(x, k), we signify that state x is entered from a different state at transition number k, here counting both accepted and (implicitly) rejected moves. This algorithm generates the (random) transition number L on which the next move to a different state is made. Let $J(x, k)$ be the number of geometric variates generated by QUICKER(x, k). Since

* Received by the editors April 26, 1993; accepted for publication January 21, 1994. This research was partially supported by Air Force Office of Scientific Research and Office of Naval Research Contract F49620-90-C-0033.

† Department of Mathematics, Campus Box 170, University of Colorado, P. O. Box 173364, Denver, Colorado 80217-3364 (bfox@castle.cudenver.edu).

each such variate can be generated in $O(1)$ time (with respect to the possible values of their respective parameters), inspection of QUICKER(x, k) shows that the work to execute it is proportional to $J(x, k)$.

Our first result deals with QUICKER's speed.

THEOREM A. *Assume a* PC *schedule or* $\{T_k\}$: *then for each* x, $J(x, k)$ *converges in quadratic mean to one as* $k \to \infty$.

All theorems are proved in subsequent sections. In §2.3.1, we show by example that Theorem A does not extend to the canonical schedule or to $\{\tilde{T}_k\}$. For Theorem A, in the definition of T_k we could replace Δ by anything (slightly) larger than the minimum distance ρ to a neighbor among local minimizers that have no neighboring local minimizers. If there are no such (isolated) local minimizers, then we can replace Δ by any positive number. Another constraint on Δ is that it be larger than d^* to get Hajek's result. Since ρ and d^* are generally hard to find, we chose Δ as above.

Remark 1. The following fact follows from Heine [13]: with the schedule $\{T_k\}$,

$$x \text{ is a } \textit{strict} \text{ (i.e., isolated) local minimizer}$$
$$\Updownarrow$$
$$J(x, k) > 1 \text{ for infinitely many } k \text{ almost surely.}$$

Heine uses an intricate argument involving the Borel–Cantelli lemma (e.g., see Chung [3], Thms. 4.2.1 and 4.2.4). He allows schedules somewhat faster than $\{T_k\}$, though the parameters in these faster schedules are hard to find. Heine's result contrasts with Theorem A, but there is no mathematical contradiction.

Using QUICKER, the expected computer time to execute simulated annealing given the number of accepted moves is proportional to that number, because, given that the next state differs from the current state, the next state obviously can be generated in $O(1)$ time. As a practical matter, however, the implicit proportionality constants here and for QUICKER depend on the neighborhood size. Thus, for example, it is impractical (and, regardless of QUICKER, nonsensical except for some contrived problems) to let every state be an explicit neighbor of every other state, but it is practical to do this implicitly as §3.4 details. With that method, there can be, perhaps surprisingly, (many) local, nonglobal minimizers (at which the speedup QUICKER gives is most striking).

The successive states that we visit *explicitly* are pairwise distinct. To recover known results about convergence in probability to the set of global optimizers, each visit to a state is weighted by the length of the corresponding implicit direct self-loop. That length is readily available from QUICKER, in (small) constant time. Under certain conditions, papers surveyed by Romeo and Sangiovanni-Vincentelli [17] find the (slow) convergence rate in terms of the number of moves, accepted and rejected; see also Remark 2 in §2.1. Using QUICKER, the convergence rate in terms of computer time is generally an order-of-magnitude faster than would be inferred directly from those results. More relevant would be counterparts of these results when self-loops are pruned from the sample path. We conjecture that, allowing asymmetric neighborhoods (e.g., as in [6] and [7]) but assuming each global minimizer has a neighboring minimizer (e.g., as with our construction of §3.4), these results extend to the chain with all self-loops deleted. Heine [13] shows that Hajek's result [11] extends this way.

Perhaps of (even) more interest than the above convergence rate, in view of our observation about expected computer time above, is the distribution of the number N of *accepted* moves to *first* visit an optimal state or, in connection with a tentative move, to scan an optimal state. The exact distribution of N is intractable, but under the following condition we can say a lot about N.

CONDITION R. The supremum λ over temperatures of the spectral radii of the one-step move matrices, conditioned on accepting all moves, with all rows and columns corresponding to the optimal states deleted, is less than one. Each such matrix has just one eigenvalue with modulus equal to its spectral radius and that eigenvalue has multiplicity one.

Section 3.4 discusses Condition R, showing that it holds under a heuristically appealing way of generating neighborhoods. On the other hand, §3.6 shows by example that Condition R does not generally hold with purely local neighborhoods. Our second result deals with the hitting-time distribution.

THEOREM B. *Condition* R *implies that*

(i) $P\{N > k\}$ *decreases exponentially in* k.

(ii) *All moments of* N *are finite.*

1.2. Speedup on parallel processors. Since the elements on the one-step move matrix, conditioned on acceptance, vary continuously with the temperature, Condition R implies the following stronger condition.

PROPERTY R*. The spectral radii of the one-step move matrices, conditioned on acceptance, with all rows and columns corresponding to the optimal states deleted, converge to $\tilde{\lambda} < 1$. (Possibly, $\tilde{\lambda} \neq \lambda$.)

This leads to a characterization of parallel speedup.

THEOREM C. *Using* QUICKER, *Condition* R *implies that, starting from a fixed state, the ratio of the (conditional) expected computer time to first visit an optimal state on one processor to the (conditional) expected time to first visit an optimal state when simulated annealing is executed independently on* m *processors, given that the number of accepted moves on each processor is greater than* k, *goes to* m *as* $\tilde{\lambda}$ *goes to one and* k *goes to infinity with* $k = o(1/(1 - \tilde{\lambda}))$.

Roughly speaking, Theorem C says that parallel processing gives asymptotically linear speedup as the set of optimal states gets hard enough to reach—given that we are in the tail of the distribution. This corresponds to a *metaprinciple* in Keilson [15, p. 92], which we translate from a reliability setting to ours as follows: the longer the search has not found an optimal state, the more exponentially distributed is the residual time to visit an optimal state. The intuition behind the theorem is then that the expectation of the minimum of m exponentially distributed independently and identically distributed (iid) variates is $1/m$ times the expectation of the first, while the condition $k = o(1/(1 - \tilde{\lambda}))$ says roughly that the current time is asymptotically negligible relative to the expected remaining time. If on some processor we hit an optimal state before reaching the (approximate) exponential tail, speedup is unimportant; likewise, if $\tilde{\lambda}$ is not near one (again ignoring possibly large implicit constants). In the (important) remaining case, we get asymptotically *linear speedup* in m on average. Keilson's results [15, Chap. 8] show that, under certain conditions, including stationarity, the distribution of the time (on one processor) to hit a rare set has an exponential tail. Here, however, stationary corresponds to constant temperature for which condition R* holds trivially, assuming (as we do) that the tentative-move matrix is irreducible. Even with constant temperature it is necessary to let k go to infinity in the statement of the theorem. With decreasing temperatures, the condition also makes only the matrices corresponding to low temperatures relevant asymptotically. For large k, the matrices are asymptotically stationary (in the sense that they converge element-wise to a limit) and so Theorem C could be anticipated from Keilson's results, though it does not seem to follow directly from them.

Fox and Simon [9] pinpoint the link between rarity, spectral radii, and corresponding left eigenvectors. Probably, they record folklore. Roughly speaking, rarity implies that the spectral radius is near one but not conversely. For rarity, in the sense of Fox and Simon [9], the first condition (sufficient by itself) is that, especially at low temperatures, each (one-step)

move from the complement $\mathcal{R}^c$ of the rare set to $\mathcal{R}$ should have small probability (made more precise in [9]). Thus, translated to our setting here, generally most moves from $\mathcal{R}^c$ to $\mathcal{R}$ are uphill and $\mathcal{R}$ should consist not only of the global minimizers but also of all states from which there is a downhill path to a global minimizer, where no move on that path has very low probability at any temperature.

With that understanding, we temporarily redefine N and conditions R and R^* accordingly. In the limit, when it becomes impossible to get from $\mathcal{R}^c$ to $\mathcal{R}$, the truncated matrices in Condition R become stochastic and so (the redefined) $\tilde{\lambda} \to 1$ from the continuity of the spectral radius as a function of the matrix elements. Strengthening a "downhill path to a global minimizer" above to a "downhill path to a global minimizer and no downhill path to a local, nonglobal minimizer" does not always work; as the example in §2.3.1 shows, that requirement sometimes would eliminate all states not in $\mathcal{S}^*$, thus making the condition above impossible to satisfy.

If on leaving $\mathcal{R}$ without first hitting $\mathcal{S}^*$ the resulting state is independently and identically (though not necessarily uniformly) distributed across successive exits from $\mathcal{R}$, then the random restarting result in Fox [7] gives linear speedup in the sense made precise there. The (more-obvious) condition for rarity is, roughly, that the equilibrium mass of $\mathcal{R}$ with respect to the limiting matrix implicit in condition R^* is small; this equilibrium mass is well defined when enriching neighborhoods as in §3.4. Fox and Simon [9] show that a small equilibrium mass is sufficient by itself (but not necessary) for the spectral radius to be near one. It follows that the expected time to hit $\mathcal{R}$ is large; in that case, speedup is important.

With the respective revised definitions of N, R, and R^*, if the phrase "an optimal state" is replaced by "$\mathcal{R}$" in the statement of Theorem C, the revised statement is also correct. Starting in $\mathcal{R}$ if the expected remaining time to visit a global minimizer is small, then for practical purposes the speedup is essentially linear. This condition is sufficient but not necessary; usually $\tilde{\lambda}$ is near one with the original definitions of N, R, and R^*, because usually $\mathcal{S}^*$ has small equilibrium mass. Thus, in Theorem C, the condition on $\tilde{\lambda}$ is reasonable.

No counterpart of Theorem C exists for deterministic methods, for example, based on combining cutting planes with branch and bound. With such algorithms, we would expect the (empirical) speedup from parallel computing to be (markedly) sublinear, at least when the number of processors exceeds a dozen say. Unlike with simulated annealing, the processors must exchange information (extensively).

An implicit assumption in Theorem C is that no overhead to parallel process occurs; this is reasonable because the processors do not have to communicate. With this understanding, (22) in §3.3 compactly transcribes Theorem C.

Examples in §§3.5 and 3.6, respectively, show that letting $\tilde{\lambda}$ converge to one *strictly* from below is a necessary condition for Theorem C. In §3.5, we show that there is essentially no speedup if the problem is too easy or if the hitting time distribution is too peaked. Scanning the proof of Theorem C shows that letting $k \to \infty$ is also a necessary condition for *linear* speedup. In view of that condition, §3.6 shows that when $\tilde{\lambda}$ equals one reaching an optimal state is asymptotically too hard, in terms of expected number of remaining moves, to get *any* speedup. Section 3.6 also suggests that $\tilde{\lambda}$ usually equals one when the neighborhoods are purely local. Section 3.4 gives a cure. Generally, the rare set $\mathcal{R}$ above consists mostly of states from which there is a downhill path to $\mathcal{S}^*$ in the original neighborhood structure, not counting (low-probability) paths made possible by the neighborhood enrichment of §3.4. From a practical viewpoint, we also get parallel speedup in the sense of Theorem C when considering first hitting to the set of optimal *and* nearly optimal states (by essentially the same argument).

Our treatment of speedup from parallel processing was stimulated by a preprint of

Shonkwiler and Van Vleck [18], but our treatment differs significantly from theirs.

2. Implicit skipping of self-loops. Theorem A implies that, if at state x at transition k, the expected time to generate the (possibly vacuous) self-loop sequence *implicitly* (via QUICKER) is *uniformly bounded* over (x, k). On the other hand, the time to generate it *explicitly*, via a sequence of time-inhomogeneous Bernoulli trials, goes to ∞ as $k \to \infty$ whenever x is a *strict* local minimizer (all neighbors strictly uphill) and the temperature converges to 0 as $k \to \infty$.

When the self-loop sequence terminates, we exit to a *different* state, say Y. The distribution of Y depends on the temperature at the end of the self-loop sequence via the corresponding transition number, say L. The time to generate Y, say by "inversion," is independent of that transition number. Just after generating Y, we reset simulated time to $L + 1$.

If one pretends that each self-loop takes exactly one transition, then the original chain is not simulated, even implicitly, when the temperature strictly decreases. Greene and Supowit [10] implicitly take this approach. With it, results of Hajek [11] and others about convergence in probability to the set of global optimizers no longer apply; they would no longer hold, when there are strict local minimizers.

Another approach to get $O(1)$ expected time to generate each direct self-loop sequence, with a slightly smaller implicit proportionality constant, uses a *random* cooling schedule that has constant temperature on each direct self-loop sequence. However, whether one then always gets convergence in probability to the set of global optimizers is an open question. When there are no isolated minima, Heine [13] answers affirmatively via an analysis of QUICKER. Theorem A shows that, as k gets large, asymptotically QUICKER always is as fast on average as the one tailored to the above adaptive schedule.

The lower the (unconditional) acceptance probability, the more savings from QUICKER. At *strict* local minimizers, as the temperature goes to zero that probability goes to zero (at a rate depending on the objective function distance to the nearest upward neighbor(s)) and so the savings at these minimizers goes to infinity. Heine [13] gives explicit bounds for this savings. The setup in Fox [7] always inhibits, via (tabu) penalties large enough to reverse local uphill-downhill relationships, short-run oscillations among local minimizers and their respective neighbors and (as a special case of that setup), the neighborhood enrichment in §3.4 of this paper, always prevents long-run oscillations among neighboring local, nonglobal minimizers whether or not the temperature goes to zero. This prevention is more powerful than that implicit in irreducibility of the tentative-move matrix or in convergence in probability to the set of global optimizers. Thus, it is enough to (implicitly) prune self-loops to dramatically increase the number of distinct states seen in fixed computer time.

2.1. The algorithm. Before stating our algorithm, we need two definitions. First, let $\alpha(x, n)$ be the (acceptance) probability that, starting from state x at transition n, the state at transition $n + 1$ differs from x. Second, let $G(x, n)$ be a geometric random variable with (success) parameter $\alpha(x, n)$. Here it is defined as the trial number of the first success. We note that $\alpha(x, n) \geq \alpha(x, n + 1)$ for all n, as is usual (but not universal) in the simulated-annealing literature; this follows from our assumption about the cooling schedule. If, contrary to our assumption, that schedule were not monotone, then the parameter of $G(x, n)$ would have to be $\sup\{\alpha(x, j) : j \geq n\}$, generally resulting in a significant loss of speed.

Given the objective-function values (not necessarily all computed from scratch) at the neighbors of x, the marginal cost of computing $\alpha(x, k)$ is modest. We view the cost of the former as *sunk*, to compute the probabilities corresponding to *intelligent* tentative moves in the sense of Fox [6], [7], which take account of these values. In particular, this is the only way to discriminate among improving moves. Sunk costs are properly ignored. Tentative

moves corresponding to a uniform distribution over neighbors or to a symmetric tentative-move matrix are not intelligent, though some papers assume such *blind* moves. On the other hand, tentative moves influenced *solely* by objective-function values at neighbors can lead to short-run oscillation, especially with respect to local minima. Such oscillation is inhibited by (tabu) penalties imposed in Fox [6], [7], via a (recent) history-remembering (Cartesian product) state space. Tabu search also computes objective-function values at neighbors; in numerous studies, it empirically beats simulated annealing with blind moves.

The algorithm QUICKER(x, k) below assumes that we start in state x at transition k. It outputs the transition number L on which the next move to a different state is made.

ALGORITHM QUICKER(x, k)
  Set $j \leftarrow k$
  Until exit, repeat
    Generate a geometric variate $G(x, j)$
    Set $L \leftarrow j + G(x, j) - 1$
    Generate a standard uniform variate V
    If $V \leq \alpha(x, L)/\alpha(x, j)$, then exit with L
    Else set $j \leftarrow L + 1$
  End

This algorithm is adapted from Fox [6], [7], where no analysis of speed is given.

Correctness. Briefly, here is the proof that the output L has the correct distribution. A naive algorithm generates iid standard uniform variates $U_1, U_2, \ldots$ and outputs the first L such that $U_L \leq \alpha(x, L)$. We do this faster by doing it implicitly. Just after passing the "V-test," L is in effect the first index after j such that $U_L \leq \alpha(x, L)$ since the geometric variate has implicitly found that $U_L \leq \alpha(x, j)$ but $U_i > \alpha(x, i)$ for $i = j, \ldots, L - 1$; the latter assertion follows from $\alpha(x, j) \geq \alpha(x, i)$ for $i > j$, which in turn is equivalent to (the assumed condition that) the temperatures are nonincreasing. But $P\{U_L \leq \alpha(x, j),\ V \leq \alpha(x, L)/\alpha(x, j) | L = \ell\}$ is $\alpha(x, \ell)$ by independence. Memorylessness after each V-test failure justifies updating j as indicated. This completes the proof.

QUICKER is a discrete-time analog of the Lewis–Shedler thinning algorithm [16] to generate nonhomogeneous Poisson processes; e.g., see Bratley, Fox, and Schrage [1, §5.3.18] or Devroye [5, §VL.1.3] for a discussion of the Lewis–Shedler algorithm.

Remark 2. Chiang and Chow [2] give an inhomogeneous continuous-time Markov chain version of simulated annealing, with a detailed analysis of the rate of convergence (in probability) to the optimal set. They do not say how they would simulate. If all neighbors are at the same height above the state in question and a canonical or subcanonical schedule is used, then generating holding times by inversion is fast. Otherwise, the only way that seems practical to generate them notes that each such time corresponds to the first arrival in an inhomogeneous Poisson process (with rate function depending on the current state) and then uses the Lewis–Shedler algorithm cited above. This has the effect of introducing self-loops corresponding to the rejected arrivals. Relative to discrete-time simulated annealing with QUICKER, it is slower (by a constant factor) and more complex; likewise, for the subsequent generation of the next state. Chiang and Chow consider the probability as a function of *continuous, simulated* time that the chain is in the set of optimal states and find the (slow) rate of convergence of that probability to one as (continuous, simulated) time gets large. A more appropriate measure would be the rate as computer time gets large; on average, the latter is proportional to the number of accepted moves (accepted "arrivals" in this case).

Remark 3. With PC schedules, our algorithm QUICKER applies without change and Theorems A, B, and C apply. The following streamlined version of QUICKER may be slightly faster in this case. Let $n(j)$ be the next breakpoint in the cooling schedule after j or,

are none, infinity.

ALGORITHM QUICKER(x, k)—tailored
    Set $j \leftarrow k$
    Until exit, repeat
        Generate a geometric variate $G(x, j)$
        Set $L \leftarrow j + G(x, j) - 1$
        If $L < n(j)$ then
                Exit with L
            Else
                Generate a standard uniform variate V
                If $V \leq \alpha(x, L)/\alpha(x, j)$ then
                        Exit with L
                    Else
                        Set $j \leftarrow L + 1$
End

We get a further speedup by treating the case $n(j)$ equal infinity separately.

2.2. Its speed. Let $T(x, k)$ be the expected number of geometric variates generated by QUICKER(x, k) until exit. We show that $T(x, k)$ is uniformly bounded in k for any cooling schedule with decreasing (positive) temperatures. (To get convergence in probability to the set of global optimizers, the probability of accepting an uphill move must go to zero. For this, the limiting temperature must be zero but here we allow the limit to be any nonnegative number.)

Denote by $p(\ell; x, k)$ the joint probability that $G(x, k)$ equals ℓ and that the corresponding (first) V-test fails. Clearly,

$$T(x, k) = 1 + \sum_{\ell=1}^{\infty} p(\ell; x, k) T(x, k + \ell) \tag{1}$$

and

$$\begin{aligned} p(\ell; x, k) &= [1 - \alpha(x, k)]^{\ell-1} \alpha(x, k) \{1 - \alpha(x, k + \ell - 1)/\alpha(x, k)\} \\ &= [1 - \alpha(x, k)]^{\ell-1} [\alpha(x, k) - \alpha(x, k + \ell - 1)]. \end{aligned} \tag{2}$$

Because $p(1; x, k) = 0$, the sum in (1) effectively starts at $\ell = 2$.

Since

$$\sum_{\ell=1}^{\infty} [1 - \alpha(x, k)]^{\ell-1} \alpha(x, k) = 1, \tag{3}$$

and

$$\sum_{\ell=1}^{\infty} [1 - \alpha(x, k)]^{\ell-1} \alpha(x, k + \ell - 1) > 0 \tag{4}$$

for each k, to show that

$$\theta = \sup_k \sum_{\ell=1}^{\infty} p(\ell; x, k) < 1, \tag{5}$$

it is enough to show that

$$\phi = \lim_{k\to\infty} \inf \sum_{\ell=1}^{\infty} [1-\alpha(x,k)]^{\ell-1}\alpha(x,k+\ell-1) > 0. \tag{6}$$

We will show more: the limit exists and $\phi = 1$.

To make the rest of the proof of Theorem A easier to follow, we divide it into two cases.

Case 1. x is not a strict local minimum or the temperatures are bounded away from zero.

Case 2. x is a strict local minimum and $\{T_k\}$ is used.

The proof for Case 1 shows that, when x is not a strict local minimum, the conclusion of Theorem A holds for *any* monotone-decreasing cooling schedule. When the cooling schedule is PC, this part suffices. For Case 1, the proof relies solely on the following fact: the acceptance probabilities are bounded away from zero. Here, the strategy is to show that the probability that the first V-test in QUICKER fails converges to zero as $k \to \infty$. This is equivalent to showing that $\phi = 1$. From this, it follows easily that $T(x,k) \to 1$. A similar argument, detailed in §2.2.3, shows that $EJ^2(x,k) \to 1$ and hence that Var $J(x,k) \to 0$, completing the proof for Case 1.

The primary task for Case 2 is to show that (again) $\phi = 1$, now harder to do because the acceptance probabilities are not bounded away from zero. Once we get $\phi = 1$, the rest of the proof follows that for Case 1.

2.2.1. Case 1. Let $\tau = \lim_{k\to\infty} \alpha(x,k) > 0$. Because $\alpha(x,k)$ decreases in k, $[1-\alpha(x,k)]^{\ell-1}\,\alpha(x,k+\ell-1) \le [1-\tau]^{\ell-1}$; the upper bound is clearly summable, because $\tau > 0$ in this case. So, by dominated convergence, we bring the limit inside the sum to get

$$\phi = \lim_{k\to\infty} \sum_{\ell=1}^{\infty} [1-\alpha(x,k)]^{\ell-1}\alpha(x,k+\ell-1) = \sum_{\ell=1}^{\infty} [1-\tau]^{\ell-1}\tau = 1. \tag{7}$$

For any fixed k, plainly $p(1;x,k) + p(2;x,k) + \cdots$ is the probability that the first V-test fails. Consider the recursion

$$R(x) = 1 + \theta R(x), \tag{8}$$

where, under an alternative scenario, the V-test fails with probability θ, independently of L and j (and, hence, of k). This scenario is inconsistent with the form of the V-test in QUICKER, but that does not matter. Clearly, $R(x)$ is the expected number of geometric variates generated under that scenario, $R(x) = 1/(1-\theta)$, and $T(x,k) \le R(x)$ for all k. The boundedness of $T(x,k)$ follows, since $\theta < 1$.

Since

$$1 \le T(x,k) \le 1 + R(x) \sum_{\ell=1}^{\infty} p(\ell;x,k), \tag{9}$$

and the sum in (9) (the probability that the V-test fails) clearly converges to zero (from (2), (3), (7) and inspection of QUICKER), we get $T(x,k) \to 1$. The recursion for the second moment of the number $J(x,k)$ of geometric variates generated by QUICKER (x,k) is like (1) except for a crossproduct term. So, the same analysis (detailed in §2.4) shows that $EJ^2(x,k) \to 1$ and hence that Var$J(x,k) \to 0$. The conclusion of Theorem A follows.

2.2.2. Case 2. In this case, from the assumed form of the cooling schedule, $\alpha(x,k)$ has the form $\sum p_i(k+1)^{-\delta_i}$ with $0 < \delta_i < 1$, $p_i \geq 0$, and $\sum p_i = 1$. Here δ_i is independent of k (but depends on x). For large enough k, only the smallest δ_i matters. (In principle, as in Hajek [11, §5], one could insert dummy states and then rescale so that each δ_i equals a positive constant less than one, but it seems inefficient to do this.) In (6), stress the dependence on the function α by writing ϕ_α. Temporarily denote by β_ξ the function α with each δ_i replaced by ξ. Now let

$$\tilde{\xi} = \operatorname{argmin}\{\phi_{\beta_\xi} : \min \delta_i \leq \xi \leq \max \delta_i\}.$$

Making this replacement, we assume without loss of generality that $\alpha(x,k)$ equals $(k+1)^{-\xi}$ with $0 < \xi < 1$.

Clearly, for each m,

$$\begin{aligned}
&\sum_{\ell=1}^{\infty}(1-\alpha(x,k))^{\ell-1}\alpha(x,k+\ell-1) \\
&\quad \geq \sum_{\ell=1}^{m}(1-\alpha(x,k))^{\ell-1}\alpha(x,k+m-1) \\
&\qquad + \sum_{\ell=m+1}^{\infty}(1-\alpha(x,k))^{\ell-1}\alpha(x,k+\ell-1),
\end{aligned} \tag{10}$$

and (multiplying and dividing the sum of the finite geometric series by $\alpha(x,k)$)

$$\begin{aligned}
&\sum_{\ell=1}^{m}(1-\alpha(x,k))^{\ell-1}\alpha(x,k+m-1) \\
&\quad = \left[\frac{\alpha(x,k+m-1)}{\alpha(x,k)}\right][1-(1-\alpha(x,k))^m].
\end{aligned} \tag{11}$$

Pick a positive integer n and choose $m = \lceil (k+1)^{\xi} \rceil n$. Thus, the second factor on the right side of (11) converges to $1 - e^{-n}$ as $k \to \infty$. Since $\xi < 1$, the first factor converges to one. Since the right side of (10) is positive (at least $1 - e^{-n}$), we get $\theta < 1$. It remains to show that $\phi = 1$.

Now letting $n \to \infty$, the first term on the right side of (10) goes to one. At the same time, since the sum on the left side of (10) is at most the sum on the left side of (3), which is one (for all k), the second term on the right side of (10) goes to zero. Thus, $\phi = 1$.

Remark 4. We get additional insight by considering, over the vector space ℓ_∞, the mapping $\mathcal{H}$ defined by

$$[\mathcal{H}v](k) = 1 + \sum_{i=1}^{\infty} p(\ell; x, k)v(k+i), \tag{12}$$

where the left side is the kth component of the vector $\mathcal{H}v$. With respect to the sup norm, $\mathcal{H}$ is clearly a contraction with modulus θ. Therefore, it has a unique fixed point v^* bounded in sup norm and $v^*(k) = T(x,k)$.

2.2.3. The second moment. We already showed that $EJ(x,k) \to 1$. Thus, it suffices to show that $EJ^2(x,k) \to 1$.

It is routine to check that

$$EJ^2(x,k) = \left[1 + 2\sum_{\ell=1}^{\infty} p(\ell;x,k)T(x,k+\ell)\right] + \sum_{\ell=1}^{\infty} p(\ell;x,k)EJ^2(x,k+\ell). \tag{13}$$

Because $T(x,j)$ is uniformly bounded, as we already showed, so is the bracketed term. Now, repeating the previous argument shows that $EJ^2(x,k+\ell)$ is uniformly bounded. Taking the limit as $k \to \infty$ in (13), we get $EJ^2(x,k) \to 1$ by arguing as before.

2.3. Cooling.

2.3.1. Canonical and subcanonical schedules. A remarkable pathology of the canonical schedule follows. It is apparently folklore.

Example. There are three states: a global minimum and a local, nonglobal minimum connected only via the third state, a local maximum. It is routine to check that, with the canonical schedule, the expected number of *explicit* moves in a direct self-loop sequence starting at either minimum at any transition number is infinite.

Denote by x the local, nonglobal minimum above. Since the sum of a finite number of geometric variates is finite almost surely, it follows that there is no i such that $J(x,\cdot) > i$ only finitely often almost surely; in turn, this implies that lim inf $EJ(x,k) > 1$.

We now sharpen this observation. As a preliminary, we note that

$$\frac{d}{dp}\sum_{\ell=0}^{k}(1-p)^{\ell} = -p^{-2} + p^{-2}(1-p)^{k+1} - p^{-1}(k+1)(1-p)^k$$

$$= -\sum_{\ell=1}^{k}\ell(1-p)^{\ell-1}.$$

Now set $p = \alpha(x,k) = 1/(k+1)$, for the canonical schedule and the example above, and note that $(1-p)^{k+1} \to e^{-1}$ and $(1-p)^k \to e^{-1}$ as $k \to \infty$. Hence,

$$\frac{1}{k}\sum_{\ell=1}^{k}\ell(1-p)^{\ell-1}p \to 1.$$

Routine calculations now show that

$$\sum_{\ell=1}^{\infty} p(\ell;x,k) = \sum_{\ell=1}^{\infty}\frac{(1-\alpha(x,k))^{\ell-1}\alpha(x,k)(\ell-1)}{(k+\ell)}$$

$$\geq \sum_{\ell=1}^{k}\frac{(1-\alpha(x,k))^{\ell-1}\alpha(x,k)(\ell-1)}{2k} \to \frac{1}{2}$$

as $k \to \infty$. Thus, it follows from (1) and $T(x,j) \geq 1$ for all j that, for each $\varepsilon > 0$ and k correspondingly large enough, $T(x,k) \geq \frac{3}{2} - \varepsilon$. Peter Glynn (personal communication) gets a great refinement. He shows that $J(x,k) \Rightarrow \text{geom}(E(1+\exp(1))^{-1})$. Convergence of the means follows from Glynn's argument and monotone convergence. Since $T(x,k)$ is therefore asymptotically the reciprocal of

$$\int_0^{\infty}\frac{e^{-x}}{1+x}\,dx = e\int_1^{\infty}\frac{e^{-u}}{u}\,du$$

and the latter definite integral is in tables, we get $T(x,k) \to 1.83$ approximately.

Beyond this disadvantage relative to subcanonical schedules (where $T(x,k) \to 1$), with the canonical schedule, when k is a random variable (say, the transition number at the beginning of the second self-loop at x), there are examples where QUICKER(x,k) can be infeasible to implement numerically because $\alpha(x,k)$ can be smaller than machine precision with a significant probability; likewise, for schedules cooling only slightly more slowly. A related drawback, heuristically if not mathematically, is that the inhibition of uphill moves can be too strong too soon. Hence, we recommend that the canonical schedule *not* be used; likewise, for the perturbed schedule where the maximum "depth" is replaced by something epsilontically larger.

With the schedule $\{\tilde{T}_k\}$, an analogous pathology can happen only at an optimal state. From a practical viewpoint, this is still bad because a user, without clairvoyance or implicit enumeration, would not have identified that state as optimal. A way out checks whether the difference between objective-function values at the current state and its farthest uphill neighbor equals the bound on the maximum difference among objective-function values used in $\tilde{T}_k$. If equality holds, the current state is optimal; otherwise, the pathology cannot occur at the current state. In Theorem A, we can replace T_k by $\tilde{T}_k$ provided that we exclude optimal states x.

With the schedule $\{T_k\}$, this pathology never happens. However, the expected number of explicit moves in a direct self-loop sequence starting at a strict local minimum is not bounded as a function of the transition number on which that sequence starts. The contrast with Theorem A is sharp.

2.3.2. Qualitative considerations. Under certain conditions (not all natural in general but satisfied for a matching problem with a particular neighborhood structure), Hajek and Sasaki [12] show (nonconstructively) that no monotone-decreasing cooling sequence is optimal. As they remark, finding an optimal sequence may well be harder than the original optimization problem. In addition, the class of optimality criteria that they allow does not include the cost of implementing the cooling schedule. As we remarked earlier, QUICKER gets slowed down (generally significantly) if the cooling sequence is not monotone decreasing. So, when the cost of implementing the schedule is accounted for, a cooling sequence that decreases monotonely, at least from some point onwards, may be nearly optimal.

If the neighborhoods were purely "local," then a heuristic argument can be made occasionally to increase the temperature, even when in the tail of the cooling sequence, to stimulate the search to leave the current region. An example in §3.6 shows that, for any cooling schedule with long intervals of low temperatures, purely local neighborhoods are generally bad in a precise sense. However, when using the "enriched" neighborhoods of §3.4, if a better region exists, the search will eventually "jump" to it without reheating. A tentative move (one-step jump) to that region eventually becomes possible and any improving move is accepted.

In addition, the hybrid algorithm of Fox [7] has an initialization phase using random restarting in tandem with a descent routine. While the former corresponds to infinite temperature, the latter implicitly corresponds to zero temperature. So, our overall scheme does not have monotone-decreasing temperature—though, once simulated annealing proper is begun, it does. Without QUICKER, random restarting coupled with descent (as a stand-alone pair) would beat simulated annealing in a certain asymptotic sense as Fox [7] details.

We view the remarks above as a partial reconciliation with the Hajek–Sasaki result, though we certainly do not claim that our (implicit) overall cooling schedule is, in any sense, optimal for any combination of problem, neighborhood topology, and tentative-move probabilities. Some favor constant temperature, at least from some point onwards; results of Catoni [4] indicate that such schedules are not (even nearly) optimal.

Ideally, one would like to tailor the cooling schedule to the *computer* time budget (as opposed to a simulated-time horizon); this looks hard.

Catoni finds a sequence of schedules which are "almost" optimal, in his sense, for each respective simulated-time (not computer-time) horizon, assuming that the tentative-move matrix is symmetric (which rules out the setup in [6] and [7]). Each such schedule is a sequence of pieces, each amounting to a perturbation of the "standard" logarithmic form. The number of pieces-depends on the objective function, whereas their respective lengths depend on the simulated-time horizon. Like Hajek and Sasaki [12], Catoni [4] ignores the cost of implementing the schedule and (because his schedules depend on parameters that seem as hard to find as global optima) does not give a (genuinely) constructive way to produce nearly optimal schedules.

To us, Catoni's results roughly indicate qualitatively that the following properties are true:

1. A logarithmic tail, without breakpoints, gives a nearly optimal cooling schedule.

2. The longer the horizon (in simulated time or in computer time), the larger should be the initial point of that tail and the longer should be the initialization phase.

Even without Catoni's assumptions, the second property above is heuristically appealing. Using QUICKER in the tail makes the logarithmic form practical; with a naive simulation, it would not be. The term "tail" above seems appropriate when the horizon is far away.

2.4. Almost-sure behavior. This section states a consequence of a result of Heine [13].

Let U (L) be an upper (lower) bound on the objective function over the feasible set. By choosing an integer j greater than one and then, in the definition of T_k, choosing Δ (slightly) larger than $(j/(j-1))(U-L)$, we have $J(x,\cdot) > j$ only finitely often almost surely for all states x. With this choice of Δ, we can modify QUICKER(x,k) to exit after at most j iterations, while still assuring that the sequence of states visited converges to the optimal set. The user specifies j.

Remark 5. This modification of QUICKER may make synchronization of processors, especially on a single-instruction multiple-data (SIMD) machine, easier: if, on any processor, QUICKER finishes in less than j (real) iterations, then *dummy* iterations can be added to that execution of QUICKER so that a total of j iterations are executed with each invocation of QUICKER on each processor.

From Theorem 2.3 of [13], it follows that no such synchronization is possible with the canonical schedule.

3. The number of accepted moves. Consider the following weakening of Condition R.

CONDITION R′. With $\mathbf{A}_j$ the (nonstationary) matrix of one-step transition probabilities, conditioned on acceptance at move j, the matrix

$$\mathbf{A}_\infty = \lim_{j\to\infty} \mathbf{A}_j \tag{14}$$

has no closed set of states not containing an optimal state. This condition does not rule out isolated, local, nonglobal minimizers because their respective nearest uphill neighbors can have other downhill paths leading from them.

Let $\tilde{A}_j$ be $\mathbf{A}_j$ with all rows and columns corresponding to optimal states deleted; likewise, for $\tilde{\mathbf{A}}_\infty$. We now show that condition R′ implies Condition R. It is well known that the irreducibility of $\mathbf{A}_j$ implies that the spectral radius μ_j of $\tilde{A}_j$ is less than one. Likewise, Condition R′ implies that the spectral radius μ_∞ of $\tilde{A}_\infty$ is less than one. Therefore, since eigenvalues are continuous functions of the matrix elements, there is a λ, *less than one*, such that $\mu_j \le \lambda$ for *all* j. We pick λ as in Condition R. The point of introducing Condition R′ is that it is easier to guarantee directly, as §3.4 shows. An example in §3.6 shows that, without Condition R′, the conclusions of Theorems B and C may not hold.

Let N_x be the number of moves to hitting the set of optimal states, starting from state x. To bound the probability that N_x exceeds k, we note that $P\{N_x > k\}$ would equal $[\mathbf{B}^k\mathbf{1}]_x$ if the transitions among the suboptimal states were governed by a fixed matrix $\mathbf{B}$. So,

$$P\{N_x > k\} \leq \max_{(j_1,\ldots,j_k)} \left[\prod_{i=1}^{k} \tilde{\mathbf{A}}_{j_i}\mathbf{1}\right]_x. \tag{15}$$

Equation (15) and Condition R together imply that

$$P\{N_x > k\} = O(\lambda^k) \tag{16}$$

for fixed λ and large k.

3.1. General Jordan forms. Perturbation of the elements of $\tilde{A}_j$ is one way to make its eigenvalues distinct. One objection to such perturbation is that it may require changing zero elements to positive elements.

So, instead, we consider the following weaker property.

PROPERTY E. Only one eigenvalue of each $\tilde{A}_j$ has modulus equal to its spectral radius.

This follows from a well-known theorem of Frobenius when each $\tilde{A}_j$ is irreducible and aperiodic (in the sense that some power of $\tilde{A}_j$ has all positive elements). Such irreducibility and aperiodicity follow from neighborhood enrichment of §3.4. Property E is included in Condition R.

3.2. Proof of Theorem B. The first assertion is now clear from (16). Summing (16) over k yields

$$EN_x = O(1/(1-\lambda)) < \infty. \tag{17}$$

Since (16) implies that $P\{N_x = k\} \leq c\lambda^k$ for some constant c and large enough k, $\sum kP\{N_x = k\} \leq \sum_k ck\lambda^k$ and so an easy calculation, differentiating the sum of a geometric series j times, gives

$$EN_x^j = O((1-\lambda)^{-j}) < \infty. \tag{18}$$

This completes the proof. Only Condition R was used.

3.3. Proof of Theorem C. Clearly,

$$P\{N_x > k+\ell | N_x > k\} \leq \max_{(j_1,\ldots,j_{k+\ell})} \left[\prod_{i=1}^{k+\ell} \tilde{\mathbf{A}}_{j_i}\mathbf{1}\right]_x \cdot \left[\prod_{i=1}^{k} \tilde{\mathbf{A}}_{j_i}\mathbf{1}\right]_x^{-1}, \tag{19}$$

where $j_1, \ldots, j_{k+\ell}$ are the transition numbers where the first $k+\ell$ acceptances occur. We get an analogous lower bound by replacing max by min. By hypothesis of the theorem, the spectral radii of the $\tilde{A}_j$s converge to $\tilde{\lambda} < 1$; the corresponding left and right eigenvectors also converge.

To get insight, we temporarily consider the time-homogeneous case with $\tilde{A}_j$ diagonalizable with eigenvalues $\beta_0, \beta_1, \ldots, \beta_n$ and $\tilde{\lambda} = \beta_0$. The right side of (19) then has the form

$$\frac{\tilde{\lambda}^{k+\ell}d_1 + \beta_1^{k+\ell}d_2 + \cdots + \beta_n^{k+\ell}d_n}{\tilde{\lambda}^k d_1 + \beta_1^k d_2 + \cdots + \beta_n^k d_n}.$$

For fixed $\tilde{\lambda}$ and ℓ, this converges to $\tilde{\lambda}^\ell$ as $k \to \infty$. Thus, the left side of (19) asymptotically becomes the geometric distribution, consistent with Keilson's results.

Now we return to the time-inhomogeneous case, no longer assuming tht $\tilde{A}_j$ is diagonalizable but using Property E. Fixing ℓ and letting $k \to \infty$ in (19), the right side has the form

$$\lim_{k\to\infty} \frac{[\mathbf{B}_k\{(\tilde{\lambda}^{\ell+1}\mathbf{c} + o(\tilde{\lambda}^{\ell+1})\mathbf{d}) + \mathbf{f}_{k,\ell}\}]_x}{[\mathbf{B}_k\{(\tilde{\lambda}\mathbf{c} + o(\tilde{\lambda})\mathbf{d}) + \mathbf{g}_k\}]_x},$$

where for some increasing subsequence $\{j_i\}$ corresponding to the maximization

$$\mathbf{B}_k = \prod_{i=1}^{k-1} \tilde{\mathbf{A}}_{j_i},$$

$$\mathbf{f}_{k,\ell} = \prod_{i=k}^{k+\ell} [\mathbf{A}_{j_i} - \mathbf{A}_\infty]\mathbf{1},$$

$$\mathbf{g}_k = [\mathbf{A}_{j_k} - \mathbf{A}_\infty]\mathbf{1},$$

and $\mathbf{f}_{k,\ell} \to 0$ and $\mathbf{g}_k \to 0$. Here $\mathbf{f}_{k,\ell}$ is the error term arising from replacing the final $\ell + 1$ terms in the matrix product by $\tilde{\mathbf{A}}_\infty$; likewise, for $\mathbf{g}_k$. The factors in parentheses come from the spectral decomposition of $\tilde{\mathbf{A}}_\infty$. Thus, as $k \to \infty$,

$$P\{N_x > k + \ell | N_x > k\} \to \tilde{\lambda}^\ell + o(\tilde{\lambda}^\ell). \tag{20}$$

Now let $N_{x,m}$ be the minimum of m iid copies of N_x. From (19) and the independence of the m copies, we get

$$P\{N_{x,m} > k + \ell | N_{x,m} > k\} \to \tilde{\lambda}^{\ell m} + o(\tilde{\lambda}^{\ell m}) \tag{21}$$

analogously to (20). It follows from the spectral decomposition that the second term on the right above is small relative to the first *uniformly* in ℓ. Therefore, summing (21) over $\ell = 1, 2, \ldots$ and using the hypothesis $k = o(1/(1 - \tilde{\lambda}))$ gives

$$E[N_{x,m} | N_{x,m} > k] \to (1 - \tilde{\lambda}^m)^{-1} + o((1 - \tilde{\lambda}^m)^{-1}).$$

Letting $k \to \infty$ and $\tilde{\lambda} \to 1$ gives

$$\frac{E[N_x | N_x > k]}{E[N_{x,m} | N_{x,m} > k]} \to m \tag{22}$$

by L'Hôpital's rule. This proves Theorem C.

3.4. Neighborhood enrichment. For each state x, we enrich its original neighborhood $\mathcal{N}(x)$, for example, consisting of all states Hamming distance k away, so that Condition R′ holds. We defer discussion of that construction. It is enough to guarantee that Condition R′ holds, because we already showed that Condition R′ implies Condition R.

Using $\mathbf{A}_\infty$, the only moves possible are downward, horizontal, or, if at a strict local minimizer, upward to the neighbor(s) with smallest objective function value, when the temperature goes to zero as the transition number goes to infinity. (If these temperatures are bounded away from zero, then $\mathbf{A}_\infty$ is clearly irreducible.) With moves restricted as above, Condition R′ holds if every state is a neighbor of every other state, since then there are no local, nonglobal

minimizers. Such neighborhoods would make QUICKER impractical, because of the work to compute $\alpha(x,k)$. A way to assure Condition R′ while leaving $\alpha(x,k)$ practical to compute is to make every state a *potential* neighbor of every other state in the following sense.

Enrich $\mathcal{N}(x)$ by generating a state Y randomly (not necessarily uniformly) from the remaining states, each state getting positive mass, and then replacing $\mathcal{N}(x)$ by its union with $\{Y\}$ on entering x from a different state.

Remark 6. For example, we could provisionally generate Y uniformly, but with a certain (high) probability replace that Y by the result of using a descent algorithm starting at Y. Analogously to Remark 5, synchronization may be easier if the user specifies a bound on the number of descent steps.

On exit from x to a different state, we delete Y from $\mathcal{N}(x)$. A new Y is added to $\mathcal{N}(x)$ on each entrance to x from a different state. However, $\mathcal{N}(x)$ does not change during direct self-loop sequences. Thus, the acceptance probability in QUICKER depends (only) on x, the current enriched neighborhood of x, the current simulated time, and the objective function. This dynamic-neighborhood scheme can be recast in standard simulated-annealing form.

3.4.1. The construction: I. We now detail that construction. Clearly,

$$P\{\text{tentative } x_j \to x_\ell \text{ move}\} = \sum_{k\neq j}[P\{\text{tentative } x_j \to x_\ell \text{ move}|E_{jk}\} \cdot P\{E_{jk}\}], \tag{23}$$

where E_{jk} is the (low-probability) event that x_k is added (explicitly) to the neighborhood of x_j. Since x_k is either in the original neighborhood of x_j or has a positive probability of being added to it (when $k \neq j$), a tentative $x_j \to x_\ell$ is possible for all $\ell \neq j$. Thus, every state is an *implicit* neighbor of every other state. On the other hand, our algorithm *never* calculates the *unconditional* probability of a tentative $x_j \to x_\ell$ move. Another (equivalent) way of looking at this considers *(macro)states* of the form $(x_j, \tilde{E}_{jk})$, whose objective function value is that of x_j and whose (permanent) neighborhood is the union of $N(x_j)$ and $\{x_k\}$ by construction, and redefines the transition probabilities accordingly. Thus,

$$\begin{aligned} &P\{\text{tentative } (x_j, \tilde{E}_{jk}) \to (x_\ell, \tilde{E}_{\ell m}) \text{ move}\} \\ &\quad = P\{\text{tentative } x_j \to x_\ell \text{ move}|E_{jk}\} \cdot P\{E_{\ell m}\}. \end{aligned} \tag{24}$$

The scheme already described streamlines that approach.

3.4.2. The construction: II. To show that our neighborhood enrichment implies not only irreducibility of each A_j but also *weak reversibility* of each A_j and that Condition R' holds, we proceed as follows. If and only if x_j is a strict local minimizer with respect to its original neighbors, we construct the macrostates so that $(x_j, \tilde{E}_{jk})$ and $(x_j, \tilde{E}_{j\ell})$ are neighbors of each other for all $k \neq \ell$; both these macrostates also have uphill neighbors. To show weak reversibility: any forward path from $(x_j, \tilde{E}_{jk})$ to $(x_n, \tilde{E}_{n\ell})$ has height (as defined by Hajek [11]) at least as large as the backward path $(x_n, \tilde{E}_{n\ell}) \to (x_m, \tilde{E}_{nj}) \to (x_j, \tilde{E}_{jk})$ with obvious shortening if j is in $\tilde{E}_{n\ell}$. Here $m = n$ if x_n is a strict local minimizer with respect to its original neighbors and m is another element of $N(x_n)$ with at most the same height as x_n otherwise. To show Condition R': from any macrostate there is a path, consisting only of downward and horizontal segments, to an optimal macrostate. If x_j is not a global minimizer and if there is no downward move from $(x_j, \tilde{E}_{jk})$, then for some n there is a horizontal move to $(x_n, \tilde{E}_{nz})$ where z is a global minimizer, and if with respect to its original neighbors x_j is a strict local minimizer, $n = j$. To streamline this procedure: on "exit" from x with temporary neighborhood $N(x) \cup \{Y\}$, with (small) positive tentative-move probability allow the next

state to be x with new temporary neighborhood $N(x) \cup \{Y'\}$, where $Y' \neq Y$. Thus, no minimizer is isolated when considering macrostates. Hajek's d^* is zero.

Remark 7. Therefore, by Remark 1, when using $\{T_k\}$ we can modify QUICKER so that it generates just one geometric variate and then terminates after its first *set* statement. Thus, neighborhood enrichment yields more efficient synchronization than (otherwise) obtainable from Remark 4. Unless the objective-function value of Y' is less than that of x, one would normally make the tentative-move probability small; any other tentative move would be to an uphill neighbor. Even though the speedup from QUICKER does not go to infinity, at low temperatures the speedup at formerly isolated local minimizers is huge.

Equation (24) shows that (generally) not every macrostate is a neighbor, explicit or implicit, of every other macrostate—in contrast to the situation for the original states. The matrices A_j and $\tilde{A}_j$ should be interpreted relative to transitions among macrostates, in the setting of neighborhood enrichment. Equation (24) shows that (usually) there are macrostates that are local, nonglobal minimizers. This is needed to make it reasonable that $\tilde{\lambda}$ is near one, because otherwise, in the limit, absorption takes place in number of moves at most equal to the (generally very large) number of rows of $\tilde{A}_\infty$, implying that $\tilde{\lambda}$ vanishes.

3.4.3. Diversification. Neighborhood enrichment can be seen as a way to *diversify* the search over the state space, so that it is not myopic. Though not required for our theorems, we let the probability of a tentative $x \to z$ move vary inversely with objective function value of z. To a large extent, this makes the work to compute $\alpha(x, k)$ a sunk cost, considering that the calculation of the objective-function values of all elements of $\mathcal{N}(x)$ needs to be done to compute the tentative move probabilities. It also makes the search unlikely to get sidetracked to Y, unless Y has a smaller objective-function value than those in the original $\mathcal{N}(x)$, especially at low temperatures. Thus, we diversify without making the search aimless.

Remark 8. Fox [7] uses this idea in a hybrid algorithm in which (tabu-searchlike) penalties inhibit heuristically bad moves, such as those leading to short-run oscillation in the sequence of states visited, and in which a radically modified genetic-algorithm is integrated. A counterpart of neighborhood enrichment makes at least one *historyless* element, a randomly generated feasible solution, part of each *population* as defined in Fox [7] following genetic-algorithm jargon. While at low temperatures neighborhood enrichment as described above has a strong flavor of random restarting, in the richer setting of Fox [7] a better balance with local searching (sometimes called *intensification*) is achieved. The (pseudo)objective function value for each population is the minimum of the (pseudo) objective function values for its respective elements. Tabu penalties are reflected in the latter values. While at first sight, remembering recent history of some population elements excludes weak reversibility, the historyless elements and a counterpart of the construction in §3.4.2 together guarantee weak reversibility.

Another advantage of diversification is that it makes irreducibility and weak reversibility of the tentative-move matrix hold trivially; even in the limit as the temperature goes to zero, A_∞ is not generally irreducible, but there is a path (not necessarily downhill) from every state to the set of global optimizers and from each global optimizer to every other. Thus, $\mathbf{A}_\infty$ has just one recurrence class and so (e.g., see Karlin and Taylor [14, p. 4] its (maximal) eigenvalue one has multiplicity one. Condition R follows. In the Introduction, the phrase "equilibrium mass of $\mathcal{R}$" should be interpreted as its equilibrium mass relative to $\mathbf{A}_k$ for large k. If the (tabu) penalties mentioned above are high enough, then diversification makes $\mathbf{A}_\infty$ irreducible too because, roughly speaking, the original neighbors of any local minimizer become farther uphill than some new neighbor. Typically, when that minimizer is global, these new neighbors are in $\mathcal{R}^c$— making it intuitively plausible that the equilibrium mass (in the sense above) of $\mathcal{R}$ is small. The latter property does not imply that the expected long-run fraction f of time the (time-inhomogeneous) chain, with direct self-loops pruned, spends in $\mathcal{R}$ is small. When the

problem has been transformed so that the difference in objective-function values between each state and its neighbors has absolute value zero or one, one might suspect that f is typically near one, because, with the direct self-loops retained and a subcanonical (or canonical) schedule, the limiting mass of the subset of $\mathcal{R}$ corresponding to global minima is one, as Hajek [11] shows. The example in §2.3.1 shows that, when there are isolated global minimizers, the problem transformation above or something akin to it is a necessary condition to assure that f is near one. Heine [13] shows that this condition along with Hajek's assumptions implies that the long-run fraction of time the pruned chain spends at global optima and their respective closest uphill neighbors converges to one; hence $f = 1$.

Remark 9. When not at a local maximum, the probability of an upward move at transition n given that the move at transition n is accepted is at least $d(n+1)^{-\xi}$ where $d > 0$ and $0 < \xi < 1$. Therefore, assuming that the objective function is not trivial, the Borel–Cantelli lemma implies that an infinite number of upward moves are accepted almost surely, with or without the problem transformation above. So, if f is near one, long-range as well as short-range attraction to $\mathcal{R}$ is indicated.

3.5. Limitations of parallel speedup. We have shown that, under certain natural conditions, parallel processing gives nearly linear speedup—asymptotically, linear. Superlinear speedup, however, seems beyond reach unless the original algorithm could be accelerated on a sequential computer. At the other extreme, the speedup can be negligible. To see that, we choose the tentative-move matrix so that the time to hit the unique optimal state is virtually constant as follows. Let all other states have the same objective-function value among themselves. Assume that $1 \to 2 \to 3 \to \cdots \to n$ is a downhill path, in the weak sense, that state n is the global minimizer, and that there are n states. Let each superdiagonal element of the tentative-move matrix be $1 - \varepsilon$. Set all other elements equal to ε^2, except that the bottom-right element is $1 - \varepsilon$. Choosing $\varepsilon = 1/(n-1)$ makes all row sums equal one. Taking n large enough forces ε arbitrarily small. The tentative-move matrix itself and that matrix with its last row and column deleted are each irreducible and aperiodic, yet parallel processing yields essentially no speedup. Since all moves are accepted, except at state n, the truncated tentative-move matrix equals the truncated one-step move matrix conditioned on acceptance. The spectrum of that matrix becomes arbitrarily close to that of the corresponding matrix with ε equal zero when the actual ε is small enough. For ε equals zero, the spectrum is the eigenvalue zero repeated $n - 1$ times. It is routine to check (by Cramer's rule) that the eigenvectors vary continuously with the spectrum and therefore so does the speedup. So, the essentially zero speedup is consistent with our earlier analysis.

More generally, speedup is negligible whenever there are no local, nonglobal minima. In such (unusual) problems, if the descent direction is deterministic, there is no speedup.

3.6. A bad example. To see what can go wrong without neighborhood enrichment or some other diversification strategy, even when weak reversibility holds, consider the following example.

Example. There are four states v, x, y, and z with heights 2, 1, 3, and 0, respectively. The (symmetric) neighborhoods are: $N(v) = \{x\}$, $N(x) = \{v, y\}$, $N(y) = \{x, z\}$, and $N(z) = \{y\}$. As the temperature $\to 0$, in the limit the only possible moves are $v \to x$, $x \to v$, $y \to x$, $y \to z$, and $z \to y$. Since $x \to y$ becomes impossible asymptotically, v and x disconnect from the sole global minimizer z. With $\tilde{A}_\infty$ corresponding to the set $\{v, x, y\}$ of strictly suboptimal states, it is easily checked that $\tilde{A}_\infty$ has spectral radius one and that

$$-k + E[N_{b,m} | N_{b,m} > k] \to \infty$$

as $k \to \infty$ where b can be v or x. Given the condition above, $N_{b,m}$ is k plus the remaining time to hit z.

Without diversification, we believe that such examples are the rule rather than the exception. With our neighborhood enrichment, such examples do not exist. Tabu penalties that are large enough to (in effect) raise v or x higher than y will cure the pathology on the example above. However, in large problems in which the only way to reach a global minimizer is to climb a hill higher than any other, tabu penalties may not cure the pathology because their effect is too local to raise the other hills high enough.

Note added in proof. A recent related paper by the author is *Simulated Annealing: Folklore, Facts, and Directions*, in Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing, H. Niederreiter and P. J.-S. Shiue, eds., Lecture Notes in Statistics, Springer-Verlag, Berlin, to appear.

Acknowledgments. We thank Peter Glynn and George Heine for helpful comments.

REFERENCES

[1] P. Bratley, B. L. Fox, and L. E. Schrage, *A Guide to Simulation*, 2nd edition, Springer-Verlag, New York, 1987.
[2] T.-S. Chiang and Y. Chow, *On the convergence rate of annealing processes*, SIAM J. Control Optim., 26 (1988), pp. 1455–1470.
[3] K. L. Chung, *A Course in Probability Theory*, 2nd edition, Academic Press, New York, 1974.
[4] O. Catoni, *Rough large deviation estimates for simulated annealing: application to exponential schedules*, Ann. Probab., 20 (1992), pp. 1109–1146.
[5] L. Devroye, *Non-uniform Random Variate Generation*, Springer-Verlag, New York, 1986.
[6] B. L. Fox, *Uniting probabilistic algorithms for optimation*, Proc. Winter Simulation Conference, 1992, pp. 500–505.
[7] ———, *Integrating and accelerating tabu search, simulated annealing, and genetic algorithms*. Ann. Oper. Res. 41 (1993), pp. 47–67. Special issue on tabu search.
[8] ———, *Random restarting versus simulated annealing*, Comput. Math. Appl., 27 (1994), Issue 6, pp. 33–36.
[9] B. L. Fox and B. Simon, *Rarity, spectral radii, and left eigenvectors*, Tech. report, University of Colorado, Denver, 1993.
[10] J. W. Greene and K. J. Supowit, *Simulated annealing without rejected moves*, IEEE Trans. Computer-Aided Design, 5 (1986), pp. 221–228.
[11] B. Hajek, *Cooling schedules for optimal annealing*, Math. Oper. Res., 13 (1988), pp. 311–329.
[12] B. Hajek and G. Sasaki, *Simulated annealing—to cool or not*, Syst. Contr. Lett., 12 (1989), pp. 443–447.
[13] G. W. Heine, Smart Simulated Annealing, Ph.D. thesis dissertation, University of Colorado, Denver, 1994.
[14] S. Karlin and H. M. Taylor, *A Second Course in Stochastic Processes*, Academic Press, New York, 1981.
[15] J. Keilson, *Markov Chain Models—Rarity and Exponentiality*, Springer-Verlag, New York, 1979.
[16] P. A. W. Lewis and G. S. Shedler, *Simulation of nonhomogeneous Poisson processes by thinning*, Nav. Res. Logistics Quart., 26 (1979), pp. 403–414.
[17] F. Romeo and A. Sangiovanni-Vincentelli, *A theoretical framework for simulated annealing*, Algorithmica, 6 (1991), pp. 302–345.
[18] R. Shonkwiler and E. Van Vleck, Parallel speed-up of Monte Carlo methods for global optimization, *J. Complexity*, 10 (1994), pp. 64–95.

SIAM J. OPTIMIZATION
Vol. 5, No. 3, pp. 506–524, August 1995

003

INCORPORATING CONDITION MEASURES INTO THE COMPLEXITY THEORY OF LINEAR PROGRAMMING*

JAMES RENEGAR†

Abstract. This work is an attempt, among other things, to begin developing a complexity theory in which problem instance data is allowed to consist of real, even irrational, numbers and yet computations are of finite precision.

Complexity theory generally assumes that the exact data specifying a problem instance is used by algorithms. The efficiency of an algorithm is judged relative to the *size* of the input. For the Turing model of computation, size refers to the bit-length of the input, which is required to consist of integers (or rational numbers separated into numerators and denominators).

We replace customary measures of size with *condition measures*. These measures reflect the amount of data accuracy necessary to achieve the desired computational goal. The measures are similar in spirit, and closely related, to condition numbers.

Key words. linear programming, complexity theory, condition numbers

AMS subject classifications. 90C05, 68Q25, 65Y20, 65G99

1. Introduction.

To introduce concepts gradually, we begin by discussing the most basic decision problem in linear programming, that of determining if a system of constraints is consistent. Our main technical results do not concern this problem.

Let $Ax \leq b$, $x \geq 0$ be the system of interest, where A is an $m \times n$ matrix whose coefficients are real numbers. The system is represented by the *data vector* $d := (A, b) \in \mathbb{R}^{mn+m}$; think of the coefficients of A and b as being strung into a long vector. We refer to $\mathbb{R}^{mn+m}$ as *data space*; each vector in data space represents a problem instance.

For an instance $d' = (A', b')$, let

$$\text{Soln}(d') := \{x; A'x \leq b',\ x \geq 0\}.$$

In attempting to determine if the instance d is a consistent system of constraints, we assume that algorithms will be provided with rational approximate data $\bar{d} = (\bar{A}, \bar{b})$ and an upper bound $\bar{\delta}$ on its error, i.e., $\|d - \bar{d}\|_\infty < \bar{\delta}$. For example, one can think of the approximate data as consisting of truncated decimal expansions of the actual real number data.

In working only with the approximate data $\bar{d}$ and the upper bound $\bar{\delta}$, an algorithm will not be able to distinguish the actual instance d from any other instance within distance $\bar{\delta}$ of $\bar{d}$. Thus, to make a decision about the actual instance d, the decision must be correct for all instances within distance $\bar{\delta}$ of $\bar{d}$. With this as motivation, we define the "condition measure of instance d with respect to the decision problem" as follows.

If $\text{Soln}(d) \neq \phi$ define

$$C(d) := \|d\|_\infty / \sup\{\delta; \|d' - d\|_\infty < \delta \Rightarrow \text{Soln}(d') \neq \emptyset\}. \tag{1.1}$$

Replace $\neq$ with $=$ if Soln $(d) = \emptyset$.

Observe that $1/C(d)$ is the minimal relative perturbation size required to obtain a system from d whose answer for the decision problem is different than the answer for d. Roughly speaking, $\log C(d)$ relative bits of data accuracy are necessary to reach a decision.

Note that $0 \leq C(d) \leq \infty$.

* Received by the editors June 29, 1992; accepted for publication (in revised form) March 15, 1994. This research was supported by IBM and National Science Foundation Grant CCR- 9103285.

† Mathematical Sciences Department, IBM T. J. Watson Research Center, Yorktown Heights, New York 10598. Present address, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York 14853 (renegar@orie.cornell.edu).

Instance d is "ill-posed for the decision problem" if $C(d) = \infty$; it is *ill-conditioned* if $C(d)$ is large.

Note that $C(d)$ is invariant under positive scaling $d \mapsto td$ just as the decision problem is invariant; the reader might find it useful to assume $\|d\|_\infty = 1$ in what follows, or what is essentially the same, assume $\|\bar{d}\|_\infty = 1$.

Now we discuss what we want of an algorithm. We consider algorithms with input and output as follows.

Input. $\bar{d} = (\bar{A}, \bar{b}), \bar{\delta}$.

Output. One of the following statements:

A. Consistent,

B. Inconsistent,

C. Decision deferred.

There are three properties we want such an algorithm to possess.

I. Correctness. The algorithm should never make an incorrect decision for the actual problem instance d. As the algorithm must be applicable to any instance (i.e., d may vary from one application to the next), if the algorithm replies Consistent then correctness requires that all systems within distance $\bar{\delta}$ of the input $\bar{d}$ be consistent. Similarly, if the algorithm replies Inconsistent.

II. Computational efficiency. There are various ways to define this. In this paper, we take the approach of traditional complexity theory: requiring the input $\bar{d}, \bar{\delta}$ to consist only of rational numbers, we say the algorithm is computationally efficient if it terminates within polynomial-time as measured in terms of the bit-length of the input.

III. Data efficiency. We want the algorithm to make a decision using nearly minimal data precision. We say that the algorithm is data efficient if there exist a positive constant E and a polynomial $p(m, n)$, both independent of the actual instance d and input $(\bar{d}, \bar{\delta})$, such that the algorithm makes a decision if

$$\frac{\bar{\delta}}{\|d\|_\infty} \leq \frac{1}{p(m,n)C(d)^E}, \tag{1.2}$$

i.e., the algorithm makes a decision when provided with E times the number of relative bits of accuracy necessary to make a decision (plus a number of bits growing only like the logarithm of the dimensions of the instance).

We say that an algorithm for the decision problem is fully efficient if it is correct, computationally efficient, and data efficient.

Important note. It is conceivable that when a more formal framework is developed to encompass a broader class of problems, it may be necessary to replace (1.2) with something like

$$\frac{\bar{\delta}}{\|d\|_\infty} \leq \frac{1}{p_1(m,n)C(d)^{p_2(m,n)}}, \tag{1.3}$$

i.e., a decision is made when provided with $p_2(m, n)$ times the number of bits of relative accuracy necessary to make a decision.

Is there an algorithm for the decision problem which is fully efficient? The answer is yes as is shown in §3. In fact, for constraints of the form $Ax \leq b$, $x \geq 0$ as we are considering, the construction and analysis of such an algorithm are deceptively trivial, assuming that the algorithm can call on a polynomial-time algorithm for LP as a subroutine. In terms of (1.2) we have $E = 1, p(m, n) \equiv 2$.

(In all of our algorithm constructions we rely on a polynomial-time LP algorithm as a subroutine; any such algorithm is adequate. We treat the polynomial-time LP algorithm as a *black box*.)

If one removes the nonnegativity constraints, considering systems $Ax \leq b$, it is not so easy to argue the existence of a fully efficient algorithm for the decision problem. However, Vera [7] has constructed and analyzed one, obtaining $E = 3$ in (1.2).

A foremost goal in this type of complexity theory is to keep E in (1.2) as small as possible, subject to the condition of polynomially bounded running time in terms of the bit-length of $\bar{d}, \bar{\delta}$.

It is important to understand that once one has a good algorithm for one form of constraints it does not immediately yield a good algorithm for other forms. This is in contrast to traditional complexity theory. For example, in traditional complexity theory one can replace the single-variable, single-equation system $3x = 6$ with the equivalent two-constant system $3x \leq 6$, $3x \geq 6$; such transformations roughly preserve bit-length. However, when developing a complexity theory based on condition measures, such transformations are inadequate. The first system is well-posed with respect to the decision problem; small perturbations preserve consistency. The second system is ill-posed; arbitrarily small perturbations can destroy consistency.

Judging the efficiency of algorithms relative to condition measures introduces demands on algorithms not required in traditional complexity theory, but the converse is also true. Judged relative to condition measures, algorithms are required to perform few operations in deciding that an instance is consistent (inconsistent) if the instance is far from being inconsistent (consistent). However, algorithms are not even required to make a decision for ill-posed instances, regardless of how accurate the data is. The reason why is the requirement that input $\bar{d}, \bar{\delta}$ satisfy the strict inequality $\|\bar{d} - d\|_\infty < \bar{\delta}$. If we replaced $<$ with $\leq$, the results of this paper would be unaffected, but the character of the general theory would not be. The relation $\leq$ would force our theory to be strictly more stringent than traditional complexity theory because it would require that any rational data instance be solved in polynomial-time; just input $\bar{d} = d$, $\bar{\delta} = 0$. A strict inequality leaves rational data vectors d undistinguished from irrational ones. A strict inequality leaves open the possibility of efficient algorithms, when judged in terms of condition measures, for problems that are NP-hard in the sense of traditional complexity theory; this possibility is not addressed in this paper.

1.1. We now move to the next level of difficulty, constructing an approximate solution to a system of constraints when one exists. Again we let $d = (A, b)$ denote the data vector of the actual instance, $\bar{d} = (\bar{A}, \bar{b})$ denote rational approximate data, and $\bar{\delta}$ denote a rational upper bound on the data error, $\|\bar{d} - d\|_\infty < \bar{\delta}$.

We consider algorithms with input and output as follows.

Input. $\bar{d} = (\bar{A}, \bar{b})$, $\bar{\delta}$.

Output. One of the following statements.

A. Consistent. Approximate solution: $\bar{x}$. Error bound: $\bar{\varepsilon}$.

B. Inconsistent.

C. Decision deferred.

The approximate solution $\bar{x}$ and error bound $\bar{\varepsilon}$ are computed by the algorithm. If the algorithm replies statement A, then it is asserting that there exists $x \in \text{Soln}(d)$ satisfying $\|x - \bar{x}\|_\infty \leq \bar{\varepsilon}$.

We allow $\bar{\varepsilon} = \infty$; we assume that the algorithm (Turing machine) has a distinguished symbol for ∞.

What do we want of such an algorithm?

I. Correctness. Besides the aspects of correctness previously discussed, correctness requires that if the algorithm replies statement A, then

$$\|d' - \bar{d}\|_\infty < \bar{\delta} \Rightarrow \exists\, x' \in \text{Soln}(d') \text{ s.t. } \|x' - \bar{x}\|_\infty \leq \bar{\varepsilon},$$

that is, $\bar{x}$ is an approximate solution for all instances d' within error $\bar{\delta}$ of $\bar{d}$.

II. Computational efficiency. The algorithm terminates within polynomial-time as measured in terms of the bit-length of the input $\bar{d}, \bar{\delta}$.

III. Data efficiency. The next few paragraphs are devoted to a discussion of this.

Data efficiency is more involved here, primarily because we are not simply dealing with a yes-no answer, but also because the tasks are becoming layered; first, there is the task of deciding consistency; second, there is the task of computing $\bar{x}$ and $\bar{\varepsilon}$ if the system is determined to be consistent.

We associate a condition measure with each task layer. For the first layer, that of deciding consistency, the condition measure is the same as before, $C(d)$. The first requirement of data efficiency is that the algorithm reply statement A or B whenever $\bar{\delta}$ satisfies (1.2), where E and $p(m, n)$ are again instance- and input-independent.

The second layer of tasks is that of computing $\bar{x}$ and $\bar{\varepsilon}$ (possibly ∞) if consistency has been determined. The condition measure associated with this should reflect the finest solution accuracy one could hope for with the given data accuracy. There are various nonequivalent ways to formalize this, at least one of which is natural for algorithm analysis.

For instance, d and all $\varepsilon \geq 0$, define

(1.4)
$$C(d, \varepsilon) := \|d\|_\infty / \sup\{\delta; \exists \tilde{x} \text{ s.t. } \|d' - d\|_\infty < \delta \Rightarrow \exists x' \in \text{Soln}(d') \text{ s.t. } \|x' - \tilde{x}\|_\infty \leq \varepsilon\}.$$

Thus, $1/C(d, \varepsilon)$ represents the largest relative inaccuracy in the data with which one could hope to compute a point $\tilde{x}$ guaranteed to be within error ε of a solution for d, i.e., $\log C(d, \varepsilon)$ relative bits of accuracy are necessary.

Note that $C(d) \leq C(d, \varepsilon)$.

Besides requiring an algorithm to decide consistency efficiently, we also require for all $\varepsilon \geq 0$ that

$$\frac{\bar{\delta}}{\|d\|_\infty} < \frac{1}{p(m, n) C(d, \varepsilon)^E} \Rightarrow [\text{Algorithm replies } A \text{ where } \bar{\varepsilon} \leq \varepsilon],$$

i.e., the algorithm requires at most E times the number of relative bits of accuracy necessary to compute an ε- approximate solution.

We now summarize our discussion of data efficiency.

III. Data efficiency. There exist positive constants E_i and polynomials $p_i(m, n)$, $i = 1$, 2, all instance- and input-dependent, such that (i) the algorithm replies A or B if

(1.5)
$$\frac{\bar{\delta}}{\|d\|_\infty} < \frac{1}{p_1(m, n) C(d)^{E_1}}.$$

(ii) For all $\varepsilon \geq 0$,

(1.6)
$$\frac{\bar{\delta}}{\|d\|_\infty} < \frac{1}{p_2(m, n) C(d, \varepsilon)^{E_2}} \Rightarrow [\text{Algorithm replies A where } \bar{\varepsilon} \leq \varepsilon].$$

Again we say that an algorithm is fully efficient if it is correct, computationally efficient, and data efficient.

There do exist fully efficient algorithms in this context. It is again a deceptively trivial matter to construct and analyze such an algorithm for constraints of the form $Ax \leq b$, $x \geq 0$. We do so in §3, obtaining $E_1 = E_2 = 1$, $p_1(m, n) \equiv p_2(m, n) \equiv 2$. It is not so easy to argue the existence of fully efficient algorithms for other forms of constraint. Vera [8] has done so.

Some readers must wonder how our so-called condition measures relate to condition numbers. Roughly, one can think of the limit

$$\limsup_{\varepsilon \downarrow 0} \varepsilon C(d, \varepsilon)$$

as a condition number for instance d. In the context of linear equations, one can easily verify that an analogous limit gives the usual condition number. However, the above limit is not necessarily close to condition numbers for linear inequalities as defined, say, by Mangasarian [3]; the main difference stems from the fact that $C(d, \varepsilon)$ is highly dependent on both A and b in $d = (A, b)$, whereas condition numbers in the literature as represented by [3] are assigned to A by considering the worst-case b; in the context of square systems of linear equations the two approaches are roughly equivalent, but in the context of linear inequalities they are not. Both approaches have their merits.

Condition numbers are defined asymptotically; by contrast, our *condition measures* are global.

1.2. Now we consider linear programming proper, our main focus. We restrict attention to problems with constraints of the form $Ax \leq b$, $x \geq 0$. In this context such constraints ease the analysis but do not make it trivial.

So consider LPs of the form

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & Ax \leq b, \\ & x \geq 0. \end{aligned}$$

The data vector is $d = (A, b, c)$. Approximate data, assumed to be rational, is denoted by $\bar{d} = (\bar{A}, \bar{b}, \bar{c})$, and $\bar{\delta}$ again denotes an upper bound on the error, $\|\bar{d} - d\|_\infty < \bar{\delta}$.

For an LP instance $d' = (A', b', c')$, let $\text{Opt}(d')$ denote the optimal solution set and let $\text{Feas}(d')$ denote the feasible region, i.e., $\text{Feas}(d') := \{x;\ A'x \leq b',\ x \geq 0\}$. Let $\text{DualFeas}(d')$ denote the feasible region of the dual LP.

We consider algorithms with input and output as follows.

Input. $\bar{d} = (\bar{A}, \bar{b}, \bar{c})$, $\bar{\delta}$.

Output. One of the following statements.

A. There is an optimal solution.
   Approximation: $\bar{x}$.
   Error bound: $\bar{\varepsilon}$.

B. Unbounded optimal solution.

C. Infeasible.

D. Feasible, but decision on the existence of an optimal solution is deferred.

E. All decisions deferred.

As before we allow $\bar{\varepsilon} = \infty$. If the algorithm replies statement A, then it is asserting that there exists $x \in \text{Opt}(d)$ such that $\|x - \bar{x}\|_\infty \leq \varepsilon$.

We now have three layers of tasks: (1) decide primal feasibility; (2) if primal feasible then decide dual feasibility; (3) if both primal and dual feasible, then compute $\bar{x}$ and $\bar{\varepsilon}$. With each task layer we have a condition measure:

(1) The same as the value we have been denoting $C(d)$, i.e., if $\text{Feas}(d) \neq \emptyset$ then define

$$C_P(d) := \|d\|_\infty / \sup\{\delta;\ \|d' - d\|_\infty < \delta \Rightarrow \text{Feas}(d') \neq \emptyset\}. \tag{1.7}$$

Replace $\neq$ with $=$ if $\text{Feas}(d) = \emptyset$.

(2) We define

$$C_{PD}(d) := \max\{C_P(d), C_D(d)\}, \tag{1.8}$$

where $C_D(d)$ is defined as $C_P(d)$ is, but with DualFeas(d) instead of Feas(d).

(3) For all $\varepsilon \geq 0$,

(1.9)
$$C(d,\varepsilon) := \|d\|_\infty / \sup\{\delta;\ \exists \tilde{x} \text{ s.t. } \|d' - d\|_\infty < \delta \Rightarrow \exists x' \in \mathrm{Opt}(d') \text{ s.t. } \|x' - \tilde{x}\|_\infty \leq \varepsilon\}.$$

Note that $C_P(d) \leq C_{PD}(d) \leq C(d,\varepsilon)$; the condition measures are monotonic in the task number.

A fully efficient algorithm is one possessing the following three properties.

I. Correctness. The statement replied must be valid for all instances within distance $\bar{\delta}$ of $\bar{d}$.

II. Computational efficiency. Polynomial-time in the input bit-length.

III. Data efficiency. There exist positive constants E_i and polynomials $p_i(m,n)$, $i = 1,2,3$, all instance- and input-independent, such that

$$\frac{\bar{\delta}}{\|d\|_\infty} \leq \frac{1}{p_1(m,n)C_P(d)^{E_1}} \Rightarrow [\text{Algorithm replies A, B, C, or D}], \tag{1.10}$$

$$\frac{\bar{\delta}}{\|d\|_\infty} \leq \frac{1}{p_2(m,n)C_{PD}(d)^{E_2}} \Rightarrow [\text{Algorithm replies A, B, or C}]. \tag{1.11}$$

For all $\varepsilon \geq 0$,

$$\frac{\bar{\delta}}{\|d\|_\infty} \leq \frac{1}{p_3(m,n)C(d,\varepsilon)^{E_3}} \Rightarrow [\text{Algorithm replies A where } \bar{\varepsilon} \leq \varepsilon]. \tag{1.12}$$

In §4 we construct and analyze a fully efficient algorithm. Regarding (1.12), we obtain

$$\frac{\bar{\delta}}{\|d\|_\infty} \leq \frac{1}{KnC_{PD}(d)^3 C(d,\varepsilon)^3} \Rightarrow [\text{Algorithm replies A where } \bar{\varepsilon} \leq \varepsilon], \tag{1.13}$$

K denoting a constant. Strictly speaking we thus obtain $E_3 = 6$, but in a sense, $E_3 \approx 3$, at least when $C(d,\varepsilon)$ is large relative to $C_{PD}(d)$, as it will be as $\varepsilon \downarrow 0$ if d has a unique optimal solution and $C_{PD}(d) < \infty$.

Remarks. Requirement (1.12) is stringent, perhaps too much so even for linear programming; it creates technical headaches when Opt(d) has positive circumradius (i.e., Opt(d) is a positive-dimensional facet) and ε is only slightly larger than the circumradius of Opt(d).

Our algorithm measures the exact ℓ_∞-radius of certain polytopes specified by rational constraints. This can be done in polynomial time; by contrast, it is NP-hard to measure the ℓ_2-radius of polytopes (Bodlaender et al. [2]).

It is the author's opinion that the particular norm should not be of extreme importance in a complexity theory based on condition measures. Perhaps the most natural way to remove the dependence is to replace the right side of (1.12) with

$$[\text{Algorithm replies A where } \bar{\varepsilon} \leq p_4(m,n)\varepsilon], \tag{1.14}$$

where $p_4(m,n)$ is yet another polynomial. This alleviates the technical headaches mentioned above. For our algorithm, we obtain

$$\frac{\bar{\delta}}{\|d\|_\infty} \leq \frac{1}{KnC_{PD}(d)^3 C(d,\varepsilon)^2} \Rightarrow [\text{Algorithm replies A where } \bar{\varepsilon} \leq 2\varepsilon]. \tag{1.15}$$

Comparing (1.13) and (1.15), we save a factor of $C(d, \varepsilon)$ in the denominator on the left side only at the expense of a factor of 2 on the right side. However, to see what we lose with (1.15) consider an LP instance d such that Opt(d) is of radius $\tilde{\varepsilon} > 0$ and $C_{PD}(d) < \infty$; then $C(d, \varepsilon) = \infty \Leftrightarrow \varepsilon \leq \tilde{\varepsilon}$. Note (1.15) does not require the algorithm be able to compute ε-approximate optimal solutions when $\tilde{\varepsilon} < \varepsilon < 2\tilde{\varepsilon}$, whereas (1.13) does.

A final remark. In a more formal theory pertaining to a broader class of problem one might want to replace the exponents E_i with polynomials.

In §5 we consider the problem of computing a feasible point whose objective value is nearly optimal, again assuming that constraints are of the form $Ax \leq b$, $x \geq 0$. As the reader might expect, continuity of the optimal objective value under data perturbations makes this problem much easier than that of approximating optimal solutions.

Vera [8] has "extended" all of our results to other forms of LPs. Although he relies on some of our ideas, the other forms of LPs present many complications requiring additional ideas; his work is a significant step beyond ours. Readers might be interested to know that he finds analytical centers to be particularly useful.

Questions concerning the stability of linear programming solutions have been studied for many years, although not in the context of complexity theory; cf., Ashmanov [1] and Robinson [5]. For example, it is well known that $C_{PD}(d)$ is finite if and only if the optimal solution sets of both the instance d and its dual are bounded. Also of related interest are the regularization techniques of Tikhonov and his followers; cf., Tikhonov, Ryutin, and Agayan [6].

2. Relations between measures of condition and solution size. In this section we establish a few simple, but crucial, relations between condition measures and sizes of solutions. These relations are similar in spirit, and similar in role, to the following much-used relation in traditional complexity theory: if an L-bit linear programming problem has a feasible point (optimal solution), then it has one satisfying $\|x\|_\infty \leq 2^L$.

The relations established in this section generalize substantially as is shown in Renegar [4]. However, we now only consider problem instances $d = (A, b, c)$ of the form

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & Ax \leq b, \\ & x \geq 0. \end{aligned}$$

Fixing m and n (the number of constraints and variables), let Pri$\emptyset$ denote the set of primal infeasible LPs (represented as data vectors), and let Dual$\emptyset$ denote the set of dual infeasible LPs.

For $d = (A, b, c)$ let dis(d, Pri$\emptyset$) denote the ℓ_∞-distance from d to the set Pri$\emptyset$; define dis(d, Dual$\emptyset$) analogously.

Let d^* denote the LP that is dual to d, i.e., d^* is as follows:

$$\begin{aligned} \min \quad & b^T y \\ \text{s.t.} \quad & A^T y \geq c, \\ & y \geq 0. \end{aligned}$$

Let $k(d)$ denote the optimal objective value of d; if d is unbounded define $k(d) = \infty$; if d is primal infeasible define $k(d) = -\infty$.

PROPOSITION 2.1. *Assume* $d = (A, b, c)$ *satisfies* Opt(d) $\neq \emptyset$ *and* dis(d, Dual$\emptyset$) > 0. *Then*

$$x \in \text{Opt}(d) \Rightarrow \|x\|_1 \leq \frac{\max\{\|b\|_\infty, -k(d)\}}{\text{dis}(d, \text{Dual}\emptyset)}.$$

Proof. Fix an optimal solution $x \neq 0$. Let $\rho > 0$ and consider the perturbed LP

$$d + \Delta d := (A + \Delta A, b, c + \Delta c),$$

where

$$\Delta A := -\left(\frac{1}{\|x\|_1}\right) b e^T,$$

$$\Delta c := \left(\frac{\max\{0, -k(d) + \rho\}}{\|x\|_1}\right) e,$$

with e denoting the vector of all ones. Note that

$$(A + \Delta A)x \leq 0,$$
$$(c + \Delta c)^T x > 0.$$

Farkas' lemma implies $d + \Delta d \in \text{Dual}\emptyset$. Since

$$\|\Delta d\|_\infty \leq \frac{\max\{\|b\|_\infty, -k(d) + \rho\}}{\|x\|_1},$$

and since $\rho > 0$ is arbitrary, the proposition follows. □

PROPOSITION 2.2. *Assume d^* has an optimal solution. Then every optimal solution y for d^* satisfies*

$$\|y\|_1 \leq \frac{\max\{\|c\|_\infty, k(d)\}}{\text{dis}(d, \text{Pri}\emptyset)}.$$

Proof. This proof is analogous to the proof of Proposition 2.1. □

PROPOSITION 2.3. *Assume $k(d)$ is finite. Then*

$$-\frac{\|b\|_\infty \|c\|_\infty}{\text{dis}(d, \text{Pri}\emptyset)} \leq k(d) \leq \frac{\|b\|_\infty \|c\|_\infty}{\text{dis}(d, \text{Dual}\emptyset)}.$$

Proof. In proving the rightmost inequality, we may assume $k(d) > 0$. Letting x denote an optimal solution for d, we then have from Proposition 2.1,

$$k(d) = c^T x \leq \|c\|_\infty \|x\|_1 \leq \frac{\|b\|_\infty \|c\|_\infty}{\text{dis}(d, \text{Dual}\emptyset)}.$$

The leftmost inequality is established analogously, relying on Proposition 2.2. □

LEMMA 2.4. *Assume $k(d)$ is finite and $\Delta d := (0, \Delta b, 0)$. Then*

$$k(d + \Delta d) - k(d) \leq \|\Delta b\|_\infty \frac{\max\{\|c\|_\infty, k(d)\}}{\text{dis}(d, \text{Pri}\emptyset)}.$$

Proof. Let y denote an optimal solution for d^*. Since y is also feasible for the dual of $d + \Delta d$, we have

$$k(d + \Delta d) \leq (b + \Delta b)^T y = k(d) + (\Delta b)^T y \leq k(d) + \|\Delta b\|_\infty \|y\|_1.$$

Substituting the bound of Proposition 2.2 for $\|y\|_1$ completes the proof. □

PROPOSITION 2.5. *Assume* $\Delta d := (\Delta A, \Delta b, \Delta c)$ *and assume both* $k(d)$ *and* $k(d + \Delta d)$ *are finite. Then*

$$k(d+\Delta d) - k(d) \leq \|\Delta A\|_\infty \left[\frac{\max\{\|c\|_\infty, k(d)\}}{\text{dis}(d,\ \text{Pri}\emptyset)}\right] \left[\frac{\max\{\|b+\Delta b\|_\infty, -k(d+\Delta d)\}}{\text{dis}(d+\Delta d,\ \text{Dual}\emptyset)}\right]$$
$$+\|\Delta b\|_\infty \left[\frac{\max\{\|c\|_\infty, k(d)\}}{\text{dis}(d,\ \text{Pri}\emptyset)}\right]$$
$$+\|\Delta c\|_\infty \left[\frac{\max\{\|b+\Delta b\|_\infty, -k(d+\Delta d)\}}{\text{dis}(d+\Delta d,\ \text{Dual}\emptyset)}\right].$$

Trivially, an analogous lower bound on $k(d + \Delta d) - k(d)$ *is obtained by interchanging the roles of* d *and* $d + \Delta d$.

Remark. The value $-k(d + \Delta d)$ occurring on the right side of the inequality can be replaced with $-k(d)$; this follows immediately from the inequality by considering the two cases $k(d + \Delta d) \leq k(d)$ and $k(d) \leq k(d + \Delta d)$. Similarly, the value $k(d + \Delta d)$ appearing in the analogous lower bound can be replaced with $k(d)$.

Proof. Let x denote an optimal solution for $d + \Delta d$. Let $\Delta' d := (0, \Delta' b, 0)$, where

$$\Delta' b := \Delta b - (\Delta A)x.$$

Note that x is feasible for $d + \Delta' d$ and hence $c^T x \leq k(d + \Delta' d)$. Thus

$$k(d+\Delta d) - k(d+\Delta' d) \leq (\Delta c)^T x \leq \|\Delta c\|_\infty \|x\|_1$$

and hence

$$k(d+\Delta d) - k(d) \leq \|\Delta c\|_\infty \|x\|_1 + [k(d+\Delta' d) - k(d)].$$

Noting that $\|\Delta' d\|_\infty \leq \|\Delta b\|_\infty + \|\Delta A\|_\infty \|x\|_1$, the proof is now easily completed using Proposition 2.1 and Lemma 2.4. □

Whenever we speak of a "dimension independent constant K," we mean that the constant does not depend on the dimensions of the LP instances being considered.

COROLLARY 2.6. *There exist dimension independent constants* $K_1 > 0$ *and* $K_2 > 0$ *with the following property. If* d *and* Δd *satisfy*

$$\|\Delta d\|_\infty < K_1 \,\text{dis}(d,\ \text{Pri}\emptyset \cup \text{Dual}\emptyset),$$

then

$$|k(d+\Delta d) - k(d)| < K_2 \|\Delta d\|_\infty \frac{\|d\|_\infty \max\{\|d\|_\infty, |k(d)|\}}{\text{dis}(d,\ \text{Pri}\emptyset)\text{dis}(d,\ \text{Dual}\emptyset)}.$$

Proof. The proof follows immediately from Proposition 2.5, relying on the previous assertion that the value $-k(d+\Delta d)$ occurring on the right side of the inequality in Proposition 2.5 can be replaced with $-k(d)$ (similarly, the value $k(d + \Delta d)$ appearing in the analogous lower bound can be replaced with $k(d)$), and relying on the relations $\text{dis}(d, \text{Pri}\phi) \leq \|d\|_\infty$, $\text{dis}(d, \text{Dual}\phi) \leq \|d\|_\infty$. □

3. Trivialities. In this section we consider the problem of deciding if $Ax \leq b$, $x \geq 0$ is consistent and, if so, of computing a solution. The form of the constraints makes this section deceptively trivial; by contrast, see Vera [7] for other forms of constraints.

Let $\bar{d} = (\bar{A}, \bar{b})$ denote approximate data and let $\bar{\delta}$ denote an upper bound or its error, i.e., $\|\bar{d} - d\|_\infty < \bar{\delta}$, where d is the actual data.

The triviality of this section results from the fact that there are two elements in the set of instances

$$B_\infty(\bar{d}, \bar{\delta}) := \{d'; \|d' - \bar{d}\|_\infty \leq \bar{\delta}\}$$

that determine the consistency or inconsistency of all instances in the set. The two instances are

$$d_1 := (\bar{A} + \bar{\delta} e e^T, \bar{b} - \bar{\delta} e),$$
$$d_2 := (\bar{A} - \bar{\delta} e e^T, \bar{b} + \bar{\delta} e).$$

Defining

$$\text{Soln}(d') := \{x;\ A'x \leq b',\ x \geq 0\},$$

where $d' = (A', b')$, it is easily proven that

$$d' \in B_\infty(\bar{d}, \bar{\delta}) \Rightarrow \text{Soln}(d_1) \subseteq \text{Soln}(d') \subseteq \text{Soln}(d_2). \tag{3.16}$$

The fully efficient algorithms alluded to in §§1 and 1.1 follow from this implication. For example, the one alluded to in §1.1 is as follows.

Input. $\bar{d} = (\bar{A}, \bar{b}), \bar{\delta}$.

1. Check consistency of d_2. If inconsistent, then reply "Inconsistent" and STOP.
2. Check consistency of d_1. If inconsistent, then reply "Decision deferred" and STOP.
3. Compute a feasible point $\bar{x}$ for d_1. Let $\bar{\varepsilon} = 0$. Reply "Consistent. Approximate solution: $\bar{x}$. Error bound: $\bar{\varepsilon}$" and STOP.

Assuming that one uses a polynomial-time algorithm for checking the consistency in steps 1 and 2, and for computing $\bar{x}$ in step 3, the claims of §§1 and 1.1 follow trivially.

4. Linear programming proper. In this section we construct and analyze the fully efficient algorithm mentioned in §1.2, for LPs of the form

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & Ax \leq b, \\ & x \geq 0. \end{aligned}$$

The construction of the algorithm and the proofs of correctness and computational efficiency are all simple. The interesting aspect is the proof that the algorithm is data efficient in approximating optimal solutions.

The ℓ_∞-*radius* of a closed set S is defined to be the smallest value r for which there exists $\bar{x}$ satisfying $S \subseteq \{x;\ \|x - \bar{x}\|_\infty < r\}$; a corresponding $\bar{x}$ is called a *mid-point* of S; the ℓ_∞-radius may be ∞.

If S is specified as the feasible region for a system of linear inequalities with rational coefficients, then its ℓ_∞- radius and a mid-point can be computed in time polynomial in the bit-length of the coefficients, as the reader can easily verify.

Letting $\bar{d} = (\bar{A}, \bar{b}, \bar{c})$ and $\bar{\delta}$ denote input, define

$$d_1 := (\bar{A} + \bar{\delta} e e^T, \bar{b} - \bar{\delta} e, \bar{c} - \bar{\delta} e),$$
$$d_2 := (\bar{A} - \bar{\delta} e e^T, \bar{b} + \bar{\delta} e, \bar{c} + \bar{\delta} e).$$

The algorithm is as follows.

Input. $\bar{d} = (\bar{A}, \bar{b}, \bar{c}), \bar{\delta}$.

1. Check primal feasibility of d_2. If infeasible then reply "Infeasible" and STOP.

2. Check primal feasibility of d_1. If infeasible then reply "All decisions deferred" and STOP.

3. Check dual feasibility of d_1. If infeasible then reply "Unbounded optimal solution" and STOP.

4. Check dual feasibility of d_2. If infeasible then reply "Feasible, but decision on the existence of an optimal solution is deferred" and STOP.

5. Compute the ℓ_∞-radius $\bar{\varepsilon}$ and a mid-point $\bar{x}$ for the feasible region of the following system:

$$\begin{aligned} A_2 x &\leq b_2, \\ c_2^T x &\geq k(d_1), \\ x &\geq 0, \end{aligned} \tag{4.1}$$

where $d_2 = (A_2, b_2, c_2)$. Reply "There is an optimal solution. Approximation: $\bar{x}$. Error bound: $\bar{\varepsilon}$." STOP.

Assuming polynomial-time LP algorithms are used as subroutines, the computational efficiency of the above algorithm is immediate.

Correctness of the algorithm follows from the easily proven fact that if $d' \in B_\infty(\bar{d}, \delta)$ then

$$\text{Feas}(d_1) \subseteq \text{Feas}(d') \subseteq \text{Feas}(d_2), \tag{4.2}$$

$$\text{DualFeas}(d_2) \subseteq \text{DualFeas}(d') \subseteq \text{DualFeas}(d_1), \tag{4.3}$$

$$k(d_1) \leq k(d') \leq c_2^T x(d'), \tag{4.4}$$

where $x(d')$ denotes any optimal solution of d' (assuming one exists). We leave verification of correctness as a simple exercise.

We discussed in §1.2 that, regarding data efficiency, there are three layers of tasks, each with an appropriate condition measure: $C_P(d), C_{PD}(d)$, and $C(d, \varepsilon)$. The definition of *data efficiency* in that section addresses the task layers consecutively.

For the first task layer, that of deciding primal feasibility, the data efficiency of our algorithm is an immediate consequence of (4.2); in fact, it follows immediately that E_1 and $p_1(m, n)$ in (1.10) can be taken as the constants 1 and 2, respectively. Similarly for the second task layer, that of deciding dual feasibility, i.e., (1.11).

Finally, we come to something interesting; proving data efficiency of the algorithm in approximating an optimal solution. Fixing $\varepsilon > 0$, we wish to establish (1.12). In doing so, we may assume the input $\bar{d}$, $\bar{\delta}$ satisfies $\bar{\delta} < 1/2C(d, \varepsilon)$; it follows we may assume that upon input $\bar{d}$, $\bar{\delta}$ the algorithm does not terminate until step 5.

In what follows, $\bar{x}$ and $\bar{\varepsilon}$ refer to the approximate optimal solution and error bound computed by the algorithm upon input $\bar{d}$, $\bar{\delta}$. As always, d refers to the actual instance, i.e., the one $\bar{d}$ is considered to approximate.

Most of the remainder of this section is devoted to proving the following two propositions, the first of which is largely a consequence of the second. The first proposition is appropriate for the more stringent definition of data efficiency relying on (1.12). Either proposition is appropriate for the less stringent definition relying on (1.14), although the second proposition provides better bounds.

Recall that $C_{PD}(d) \leq C(d, \varepsilon)$.

PROPOSITION 4.1. *There is a dimension independent constant $K_3 > 0$ with the following property: For all $\varepsilon \geq 0$,*

$$\frac{\bar{\delta}}{\|d\|_\infty} \leq \frac{K_3}{nC_{PD}(d)^3 C(d,\varepsilon)^3} \Rightarrow \bar{\varepsilon} \leq \varepsilon.$$

PROPOSITION 4.2. *There is a dimension independent constant $K_4 > 0$ with the following property: For all $\varepsilon \geq 0$ and ρ satisfying $0 < \rho \leq 1$,*

$$\frac{\bar{\delta}}{\|d\|_\infty} \leq \frac{K_4 \rho}{nC_{PD}(d)^3 C(d,\varepsilon)^2} \Rightarrow \bar{\varepsilon} \leq (1+\rho)\varepsilon.$$

Before proving the propositions we use them.

THEOREM 4.3. *The preceding algorithm is fully efficient.*

Proof. The proof follows immediately from Proposition 4.1, (1.12), and the preceding discussion. □

Remark. Note that in the notation of (1.12), $E_3 = 6$ and $p(m,n) = n/K_3$; moreover, if $C(d,\varepsilon)$ is large relative to $C_{PD}(d)$ then, in a sense, $E_3 = 3$. If one instead uses the weaker definition of data efficiency relying on (1.14), then Proposition 4.2 provides better values.

Before proceeding to the proofs, we introduce simplifying notation:

$$s_{PD}(d) := \operatorname{dis}(d_1, \operatorname{Pri}\emptyset \cup \operatorname{Dual}\emptyset),$$

i.e., the ℓ_∞-distance from d to the set of LP instances, which are either primal or dual infeasible. Also, for all $\varepsilon \geq 0$,

$$s(d,\varepsilon) := \sup\{\delta; \exists\, \tilde{x} \text{ s.t. } \|d' - d\|_\infty < \delta \Rightarrow \exists x' \in \operatorname{Opt}(d') \text{ s.t. } \|x' - \tilde{x}\|_\infty \leq \varepsilon\}.$$

Note that

$$s(d,\varepsilon) = \|d\|_\infty / C(d,\varepsilon) \tag{4.5}$$

and

$$s_{PD}(d) > 0 \Rightarrow s_{PD}(d) = \|d\|_\infty / C_{PD}(d). \tag{4.6}$$

4.1. In this section we derive Proposition 4.1 from Proposition 4.2. The next section is devoted to proving Proposition 4.2.

We begin with two lemmas, the first of which is only an intermediate step to the second.

LEMMA 4.4. *If $0 < \varepsilon' \leq \varepsilon$, then*

$$s(d,\varepsilon) \leq \left(\frac{\varepsilon}{\varepsilon'}\right) s(d,\varepsilon') + \left(\frac{\varepsilon}{\varepsilon'} - 1\right) \|d\|_\infty.$$

Proof. We may assume $s(d,\varepsilon') < s(d,\varepsilon)$.

Let δ', δ satisfy $0 \leq s(d,\varepsilon') < \delta' \leq \delta < s(d,\varepsilon)$. We first show it suffices to prove that

$$\delta' \leq s(d, \varepsilon/\rho), \tag{4.7}$$

where

$$\rho := \frac{\|d\|_\infty + \delta}{\|d\|_\infty + \delta'}. \tag{4.8}$$

To see that this suffices, observe that (4.7) and $s(d,\varepsilon') < \delta'$ imply $s(d,\varepsilon') < s(d,\varepsilon/\rho)$; hence, $\varepsilon' < \varepsilon/\rho$. Substituting from (4.8) for ρ thus yields

$$\varepsilon' < \frac{\|d\|_\infty + \delta'}{\|d\|_\infty + \delta}\,\varepsilon,$$

and hence

$$\delta < \left(\frac{\varepsilon}{\varepsilon'}\right)\delta' + \left(\frac{\varepsilon}{\varepsilon'} - 1\right)\|d\|_\infty.$$

Taking the limit as $\delta' \downarrow s(d,\varepsilon')$, $\delta \uparrow s(d,\varepsilon)$ gives the lemma.

Now to prove (4.7). Consider the set of instances

$$S := \{\hat{d};\ \hat{d} = (A',\ \rho b',\ c') \text{ where } (A',b',c') \in B_\infty(d,\delta')\}.$$

Observe that $S \subseteq B_\infty(d,\delta)$; for if $\hat{d} = (A',\rho b',c')$ and $d' = (A',b',c')$ then

$$\begin{aligned}\|\hat{d} - d\|_\infty &\le \|\hat{d} - d'\|_\infty + \|d' - d\|_\infty \\ &= (\rho - 1)\|b'\|_\infty + \|d' - d\|_\infty \\ &= \frac{\delta - \delta'}{\|d\|_\infty + \delta'}\|b'\|_\infty + \|d' - d\|_\infty \\ &\le \delta - \delta' + \|d' - d\|_\infty \\ &< \delta.\end{aligned}$$

Since $\delta < s(d,\varepsilon)$ there thus exists $\tilde{x}$ such that

$$\hat{d} \in S \Rightarrow \exists \hat{x} \in \mathrm{Opt}(\hat{d}) \text{ s.t. } \|\hat{x} - \tilde{x}\|_\infty \le \varepsilon.$$

Note that if $\hat{d} = (A',\rho b',c')$ and $d' = (A',b',c')$, then

$$\hat{x} \in \mathrm{Opt}(\hat{d}) \Leftrightarrow \left(\frac{1}{\rho}\right)\hat{x} \in \mathrm{Opt}(d').$$

It follows that

$$d' \in B_\infty(d,\delta') \Rightarrow \exists x' \in \mathrm{Opt}(d') \text{ s.t. } \|x' - \left(\frac{1}{\rho}\right)\tilde{x}\|_\infty \le \frac{\varepsilon}{\rho}.$$

Hence (4.7). □

LEMMA 4.5. *Assume* $\varepsilon \ge 0$ *and define*

$$\varepsilon' := \frac{\varepsilon}{1 + \frac{s(d,\varepsilon)}{2\|d\|_\infty}}. \tag{4.9}$$

Then $s(d,\varepsilon) \le 3s(d,\varepsilon')$.

Proof. We may assume $\varepsilon > 0$. Since $\varepsilon' \le \varepsilon$, Lemma 4.4 is applicable. Substituting (4.9) for ε' in that lemma, and rearranging yields

$$s(d,\varepsilon) \le 2\left(1 + \frac{s(d,\varepsilon)}{2\|d\|_\infty}\right)s(d,\varepsilon').$$

Finally, note that $s(d,\varepsilon) \le \|d\|_\infty$. □

Proof of Proposition 4.1 *from Proposition* 4.2. We assume $\bar{\delta}$ satisfies the assumed upper bound, that is,

$$\frac{\bar{\delta}}{\|d\|_\infty} \leq \frac{K_3}{n}\left[\frac{s(d,\varepsilon)}{\|d\|_\infty}\right]^3\left[\frac{s_{PD}(d)}{\|d\|_\infty}\right]^3. \tag{4.10}$$

Let

$$\rho := \frac{s(d,\varepsilon)}{2\|d\|_\infty}, \qquad \varepsilon' := \frac{\varepsilon}{1+\rho}. \tag{4.11}$$

Lemma 4.5 shows

$$s(d,\varepsilon) \leq 3s(d,\varepsilon'). \tag{4.12}$$

Together, (4.10), (4.11), and (4.12) imply that we may assume (by requiring K_3 to be sufficiently small)

$$\frac{\bar{\delta}}{\|d\|_\infty} \leq \frac{K_4\rho}{n}\left[\frac{s(d,\varepsilon')}{\|d\|_\infty}\right]^2\left[\frac{s_{PD}(d)}{\|d\|_\infty}\right]^3$$

where K_4 is as in Proposition 4.2; thus, from that proposition,

$$\bar{\varepsilon} \leq (1+\rho)\varepsilon' = \varepsilon. \qquad \square$$

4.2. In this section we prove Proposition 4.2. We begin with a proposition which in effect asserts that for slightly worse data error than $\bar{\delta}$, one cannot hope for much better solution accuracy than $\bar{\varepsilon}$ if $s_{PD}(d)$ is not small (i.e., if $C_{PD}(d)$ is not large).

PROPOSITION 4.6. *There exist dimension-independent positive constants* K_5, K_6 *with the following property: If* $\bar{\delta}$ *and* $\Delta\delta$ *are positive numbers satisfying*

$$\bar{\delta} + \Delta\delta \leq K_5 s_{PD}(d),$$

then for all $\varepsilon \geq 0$,

$$\varepsilon < \bar{\varepsilon} - K_6\left(\frac{\bar{\delta}}{\Delta\delta}\right)\left[\frac{\|d\|_\infty}{s_{PD}(d)}\right]^3 \Rightarrow 2\bar{\delta} + \Delta\delta \geq s(d,\varepsilon). \tag{4.13}$$

We begin the proof of Proposition 4.6 with a lemma. The instances d_1 and d_2 referred to are those of the algorithm.

LEMMA 4.7. *Assume* $\Delta\delta$ *is a real number satisfying*

$$0 < \Delta\delta \text{ and } \bar{\delta} + \Delta\delta < s_{PD}(\bar{d}). \tag{4.14}$$

There exist instances d' *and* d'' *satisfying*

$$\|d' - \bar{d}\|_\infty \leq \bar{\delta} + \Delta\delta, \tag{4.15}$$

$$\|d'' - \bar{d}\|_\infty \leq \bar{\delta} + \Delta\delta, \tag{4.16}$$

$$\operatorname{dis}(\operatorname{Opt}(d'), \operatorname{Opt}(d'')) \geq 2\left[\bar{\varepsilon} - \frac{k(d_2) - k(d_1)}{\Delta\delta}\right], \tag{4.17}$$

where $\operatorname{dis}(S,T)$ *is the* ℓ_∞*-distance between subsets* S *and* T. (*Note*: (4.14) *and* (4.15) *imply* $\operatorname{Opt}(d') \neq \emptyset$; *similarly*, $\operatorname{Opt}(d'') \neq \emptyset$.)

Proof. First assume that $\bar{\varepsilon} < \infty$; recall that $\bar{\varepsilon}$ is the ℓ_∞-radius of the feasible region for (4.1), so the projection of that feasible region onto some coordinate axis is an interval of length $2\bar{\varepsilon}$; assume this is so for the first coordinate axis. Let x' denote a feasible point for (4.1) whose projection is least, and let x'' denote a feasible point whose projection is greatest. So $e_1^T(x'' - x') = 2\bar{\varepsilon}$, where e_1 is the first unit vector.

Let

$$d' = (A', b', c') := (A_2, b_2, c_2 - (\Delta\delta)e_1),$$
$$d'' := (A_2, b_2, c_2 + (\Delta\delta)e_1),$$

where $d_2 = (A_2, b_2, c_2)$. Since $\operatorname{Feas}(d') = \operatorname{Feas}(d_2)$, we have

$$x \in \operatorname{Feas}(d') \Rightarrow (c')^T x \leq k(d_2) = (\Delta\delta)e_1^T x. \tag{4.18}$$

Since x' is feasible for (4.1), we have $c_2^T x' \geq k(d_1)$ and hence

$$(c')^T x' \geq k(d_1) - (\Delta\delta)e_1^T x'. \tag{4.19}$$

Noting that $x' \in \operatorname{Feas}(d')$, together with (4.18) and (4.19), yield

$$x \in \operatorname{Opt}(d') \Rightarrow k(d_2) - (\Delta\delta)e_1^T x \geq k(d_1) - (\Delta\delta)e_1^T x',$$

that is,

$$x \in \operatorname{Opt}(d') \Rightarrow e_1^T x \leq e_1^T x' + \frac{k(d_2) - k(d_1)}{\Delta\delta}. \tag{4.20}$$

Similarly,

$$x \in \operatorname{Opt}(d'') \Rightarrow e_1^T x \geq e_1^T x'' - \frac{k(d_2) - k(d_1)}{\Delta\delta}. \tag{4.21}$$

From (4.20), (4.21), and $e_1^T(x'' - x') = 2\bar{\varepsilon}$, we obtain (4.17).

If $\bar{\varepsilon} = \infty$ then the proof proceeds exactly as above except that x' and x'' are chosen so that $e_1^T(x'' - x')$ is arbitrarily large. □

Proof of Proposition 4.6. Since $s_{PD}(d) < s_{PD}(\bar{d}) + \bar{\delta}$, we may assume by choosing K_5 sufficiently small that $\Delta\delta$ satisfies the assumptions of Lemma 4.7. Hence there exist instances d' and d'' satisfying

$$d',\ d'' \in B_\infty(d, 2\bar{\delta} + \Delta\delta), \tag{4.22}$$

$$\operatorname{dis}(\operatorname{Opt}(d'),\ \operatorname{Opt}(d'')) \geq 2\left[\bar{\varepsilon} - \frac{k(d_2) - k(d)}{\Delta\delta} - \frac{k(d) - k(d_1)}{\Delta\delta}\right]. \tag{4.23}$$

Choosing K_5 sufficiently small we may assume d and $\Delta d := d_1 - d$ satisfy the assumption of Corollary 2.6; similarly for d and $d_2 - d$. Substituting the implied bounds on $k(d_2) - k(d)$ and $k(d) - k(d_1)$ into (4.23), then substituting the bound $|k(d)| \leq \|d\|_\infty^2 / s_{PD}(d)$ implied by Proposition 2.3, one obtains (using $s_{PD}(d) \leq \|d\|_\infty$)

$$\operatorname{dis}(\operatorname{Opt}(d'),\ \operatorname{Opt}(d'')) \geq 2\left(\bar{\varepsilon} - K_6\left(\frac{\bar{\delta}}{\Delta\delta}\right)\left[\frac{\|d\|_\infty}{s_{PD}(d)}\right]^3\right), \tag{4.24}$$

where K_6 is a dimension-independent constant. Together, (4.22) and (4.24) give (4.13). □

The fact that the nonnegativity constraints $x \geq 0$ are unaffected by data perturbations forces special attention be given the zero vector as an optimal solution; the set of instances for which $\vec{0}$ is optimal has nonempty interior. With this in mind we define

$$s_0(d) := \sup\{\delta; \|d' - d\|_\infty < \delta \Rightarrow \vec{0} \in \text{Opt}(d')\}.$$

If $\vec{0} \notin \text{Opt}(d)$ then $s_0(d) = 0$.

It is easily seen that

$$s_0(d) = \sup\{\delta; \|d' - d\|_\infty < \delta \Rightarrow \text{Opt}(d') = \{\vec{0}\}\};$$

for if $\text{Opt}(d') \neq \{\vec{0}\}$ then an arbitrarily slight perturbation of the objective for d' yields an instance for which $\vec{0}$ is not optimal.

LEMMA 4.8. *If $\bar{\delta} < s_0(d)/2$ then $\bar{x} = \vec{0}$, $\bar{\varepsilon} = 0$.*

Proof. If $\bar{\delta} < s_0(d)/2$ then $\text{Opt}(d_1) = \text{Opt}(d_2) = \{\vec{0}\}$ and hence $k(d_1) = k(d_2) = 0$. Lemma 4.7 then implies $\bar{\varepsilon} = 0$; for the lemma implies, by choosing $\Delta\delta < s_0(d) - 2\bar{\delta}$, that if $\bar{\varepsilon} > 0$ then there exists d' and d'', both of distance less than $s_0(d)$ from d, and such that either $\text{Opt}(d') \neq \{\vec{0}\}$ or $\text{Opt}(d'') \neq \{\vec{0}\}$, contradicting the relation for $s_0(d)$ noted just prior to the statement of Lemma 4.8.

If $\bar{\varepsilon} = 0$, the correctness of the algorithm implies $\bar{x}$ is optimal for all instances in the open set $\{d'; \|d' - \bar{d}\|_\infty < \bar{\delta})\}$; from this it easily argued that $\bar{x} = \vec{0}$. □

One should keep in mind the relations

$$s_0(d) \leq s(d, \varepsilon) \leq s_{PD}(d).$$

LEMMA 4.9. *For all $\varepsilon \geq 0$, $s(d, \varepsilon) \leq s_0(d) + 4\varepsilon n \|d\|_\infty$.*

Proof. We may assume $s_0(d) < s(d, \varepsilon)$ and hence $s_0(d) < s_{PD}(d)$. Then it is easily seen that for each $\rho > 0$ there exists an instance $d' = (A', b', c')$ satisfying

$$\|d' - d\|_\infty < \rho + s_0(d)$$

and such that $\text{Opt}(d')$ consists of a single point $x' \neq \vec{0}$.

Consider the instance

$$d'' := \left(A', b' + \left(\frac{2\varepsilon + \rho}{\|x'\|_\infty}\right) A'x', c'\right).$$

It follows from the complementary slackness conditions for optimality that $\text{Opt}(d'') = \{x''\}$, where

$$x := x' + \left(\frac{2\varepsilon + \rho}{\|x'\|_\infty}\right) x'.$$

Hence,

$$\text{dis}(\text{Opt}(d'), \text{Opt}(d'')) = \|x' - x''\|_\infty = 2\varepsilon + \rho.$$

Consequently,

$$\begin{aligned} s(d, \varepsilon) &\leq \max\{\|d' - d\|_\infty, \|d'' - d\|_\infty\} \\ &\leq \|d' - d\|_\infty + \|d'' - d'\|_\infty \\ &\leq \|d' - d\|_\infty + (2\varepsilon + \rho) n \|d'\|_\infty \\ &< [s_0(d) + \rho] + (2\varepsilon + \rho) n [\|d\|_\infty + s_0(d) + \rho]. \end{aligned}$$

Noting $s_0(d) \leq \|d\|_\infty$, the lemma follows since $\rho > 0$ is arbitrary. □

Proof of Proposition 4.2. We assume $\bar{\delta}$ satisfies the assumed upper bound, that is

$$\frac{\bar{\delta}}{\|d\|_\infty} \leq \frac{K_4 \rho}{n} \left[\frac{s(d,\varepsilon)}{\|d\|_\infty}\right]^2 \left[\frac{s_{PD}(d)}{\|d\|_\infty}\right]^3. \tag{4.25}$$

We prove the proposition assuming that K_4 is *sufficiently small*; what constitutes sufficiently small will become evident during the course of the proof.

We may assume $\bar{\varepsilon} > 0$ and hence, by Lemma 4.8,

$$\bar{\delta} \geq \frac{s_0(d)}{2}. \tag{4.26}$$

Since

$$s(d,\varepsilon) \leq s_{PD}(d) \leq \|d\|_\infty, \tag{4.27}$$

it follows from (4.25), (4.26), and $\rho \leq 1$ that by choosing K_4 sufficiently small, we may assume

$$s_0(d) \leq \frac{s(d,\varepsilon)}{2}.$$

Thus, by Lemma 4.9,

$$s(d,\varepsilon) \leq 8\varepsilon n \|d\|_\infty. \tag{4.28}$$

Also note (4.25) and $\bar{\delta} > 0$ imply

$$s(d,\varepsilon) > 0. \tag{4.29}$$

Define

$$\Delta\delta := \min\left\{\frac{1}{2}, K_5\right\} s(d,\varepsilon) - 2\bar{\delta}, \tag{4.30}$$

where K_5 is as in Proposition 4.6. Note that (4.29) and (4.30) imply

$$2\bar{\delta} + \Delta\delta < s(d,\varepsilon). \tag{4.31}$$

Also note that (4.25), (4.27), (4.30), and $\rho \leq 1$ imply that by choosing K_4 sufficiently small we may assume

$$\Delta\delta \geq \frac{1}{2} \min\left\{\frac{1}{2}, K_5\right\} s(d,\varepsilon). \tag{4.32}$$

From (4.29) and (4.32), we have

$$\Delta\delta > 0. \tag{4.33}$$

Moreover, (4.27), (4.29), and (4.30) imply

$$\bar{\delta} + \Delta\delta \leq K_5 s_{PD}(d). \tag{4.34}$$

From (4.33) and (4.34) we find that Proposition 4.6 is applicable; consideration of the proposition in conjunction with (4.31) gives

$$\varepsilon \geq \bar{\varepsilon} - K_6 \left(\frac{\bar{\delta}}{\Delta\delta}\right) \left[\frac{\|d\|_\infty}{s_{PD}(d)}\right]^3. \tag{4.35}$$

Substituting (4.25) and (4.32) into (4.35), we find

$$\varepsilon \geq \bar{\varepsilon} - \frac{K_4 K' \rho}{n} \left[\frac{s(d,\varepsilon)}{\|d\|_\infty}\right],$$

where

$$K' := \frac{2K_6}{\min\{\frac{1}{2}, K_5\}}.$$

Hence, by choosing K_4 sufficiently small, we may assume

$$\varepsilon \geq \bar{\varepsilon} - \frac{\rho}{8n} \left[\frac{s(d,\varepsilon)}{\|d\|_\infty}\right].$$

Then, by (4.28)

$$\varepsilon \geq \bar{\varepsilon} - \rho\varepsilon,$$

completing the proof. □

5. More simple stuff. In this section we consider the problem of computing a feasible point whose objective value is nearly optimal assuming, as always, constraints are of the form $Ax \leq b$, $x \geq 0$.

The algorithm is identical with that of §4 except we replace step 5 with the following.

5. Compute an optimal solution $\bar{x}$ for d_1, compute $k(d_1)$ and $k(d_2)$. Let $\bar{\varepsilon} := k(d_2) - k(d_1)$. Reply "There is an optimal solution. Feasible point $\bar{x}$. Bound on the difference between the optimal value and the objective value of the feasible point: $\bar{\varepsilon}$."

Assuming polynomial-time LP algorithms are used as subroutines, the computational efficiency of the algorithm is immediate: it terminates within time polynomial in the bit length of the input $\bar{d}, \bar{\delta}$.

Correctness of the algorithm is a simple exercise relying on relations (4.2), (4.3), and (4.4).

It only remains to prove the algorithm is *data efficient*, a phrase which we have yet to define in this context but which the reader no doubt can infer from the previous development. The definition is the same as (1.10), (1.11), and (1.12) except for two changes. First, statement A in the algorithm referred to there should be replaced with the statement replied in step 5 above. Second, $C(d,\varepsilon)$ must be redefined:

$$C(d,\varepsilon) := \|d\|_\infty / \sup\{\delta;\ \exists k \text{ s.t. } \|d' - d\|_\infty < \delta \Rightarrow |k(d') - k| \leq \varepsilon\}.$$

This value indicates the data accuracy necessary to approximate the optimal value to within error ε, but does not seem to indicate the accuracy needed to compute a feasible point whose objective value is within ε of the optimal value. The fact that it does so follows from the relations (4.2) and (4.4) which are very particular to constraints of the form $Ax \leq b$, $x \geq 0$. In fact

$$C(d,\varepsilon) = \|d\|_\infty / \sup\{\delta;\ \exists \tilde{x} \text{ s.t. } \|d' - d\|_\infty < \delta \Rightarrow [\tilde{x} \in \text{Feas}(d') \text{ and } |k(d') - c'\tilde{x}| \leq \varepsilon]\},$$

where c' refers to the objective of d'.

Relying on the relations (4.2), (4.3), and (4.4) the reader should have no difficulty verifying that the algorithm is data efficient. Once again constraints of the form $Ax \leq b$, $x \geq 0$ result in a deceptively simple proof, unlike that of §4.

Acknowledgment. In closing I wish to thank a referee for the careful reading of the manuscript and many thoughtful suggestions.

REFERENCES

[1] S. A. Ashmanov, *Stability conditions for linear programming problems*, USSR Comput. Maths. Math. Phys., 21 (1981), pp. 40–49.

[2] H. L. Bodlaender, P. Gritzmann, V. Klee, and J. Van Leeuwen, *The computational complexity of norm-maximization*, Combinatorica, 10 (1990), pp. 203–225.

[3] O. L. Mangasarian, *A condition number for linear inequalities and linear programs*, Proc. 6th Symp. Oper. Res., Augsburg, September 7–9, 1981, G. Bamberg and O. Opitz, eds., Verlagsgroppe Athenaum, Hain, Scriptor, Hanstein, Konigstein, pp. 3–15.

[4] J. Renegar, *Some perturbation theory for linear programming*, Math. Programming, 65 (1994), pp. 73 – 91.

[5] S. Robinson, *A characterization of stability in linear programming*, Oper. Res., 25 (1977), pp. 435–447.

[6] A. N. Tikhonov, A. A. Ryutin, and G. M. Agayan, *On a stable method of solving a linear programming problem with approximate data*, Soviet Math. Dokl, 28 (1983), pp. 494–500.

[7] J. Vera, *Ill-posedness and the complexity of deciding existence of solutions to linear programs*, SIAM J Optim., to appear.

[8] ———, *Ill-posedness and the computation of solutions to linear programs with approximate data*, preprint. (Available from jvera@dii.uchile.cl.)

SIAM J. OPTIMIZATION
Vol. 5, No. 3, pp. 525–551, August 1995

004

GLOBAL CONVERGENCE OF A LONG-STEP AFFINE SCALING ALGORITHM FOR DEGENERATE LINEAR PROGRAMMING PROBLEMS *

TAKASHI TSUCHIYA† AND MASAKAZU MURAMATSU‡

Abstract. In this paper we present new global convergence results on a long-step affine scaling algorithm obtained by means of the local Karmarkar potential functions. This development was triggered by Dikin's interesting result on the convergence of the dual estimates associated with a long-step affine scaling algorithm for homogeneous LP problems with unique optimal solutions. Without requiring any assumption on degeneracy, we show that moving a fixed proportion λ up to two-thirds of the way to the boundary at each iteration ensures convergence of the iterates to an interior point of the optimal face as well as the dual estimates to the analytic center of the dual optimal face, where the asymptotic reduction rate of the value of the objective function is $1-\lambda$. We also give an example showing that this result is tight to obtain convergence of the dual estimates to the analytic center of the dual optimal face.

Key words. linear programming, interior point methods, affine scaling algorithm, global analysis, degenerate problems

AMS subject classification. 90C05

0. Introduction. Since Karmarkar [17] proposed the projective scaling algorithm for linear programming in 1984, a number of interior point algorithms have been proposed and implemented. The affine scaling algorithm, originated by Dikin [7] and rediscovered by several authors including Barnes [4], Vanderbei, Meketon, and Freedman [43], Karmarkar and Ramakrishnan [18], and Adler et al. [1], is one of the most popular interior point algorithms obtained by substituting the affine scaling transformation in place of the projective transformation in Karmarkar's algorithm. This simple algorithm works well in practice, and now several promising experimental results [1], [2], [6], [12], [21], [24], [26], [30], [33] are reported. In contrast to its great performance in practice, our knowledge on this algorithm is rather poor, particularly under the existence of degeneracy, and there are large gaps between the theoretical convergence results and the existing efficient implementations.

The first problem to be addressed is global convergence, which is one of the most fundamental properties to be shown from the theoretical point of view. There have been several milestone papers on this problem [3], [4], [8], [11], [14], [23], [34], [43], [44] under various step-size choices and nondegeneracy assumptions; see [15] for a survey. The analysis becomes difficult, particularly when we remove the primal nondegeneracy condition. We introduced the local Karmarkar potential function [39] to overcome the difficulty and then succeeded in proving the global convergence without any assumption on degeneracy with the step-size 1/8 [38], where the Dikin's displacement vector is taken as the unit. This bound has been the best obtained so far theoretically, but it

*Received by the editors February 24, 1992; accepted for publication (in revised form) January 19, 1994. This research was supported in part by a Grant-in-Aid for Encouragement of Young Scientists 03740126 (1991) and Overseas Research Scholars (1992) of the Ministry of Education, Science and Culture of Japan..

†The Institute of Statistical Mathematics, 4-6-7 Minami-Azabu Minato-ku Tokyo 106 Japan (tsuchiya@sun312.ism.ac.jp).

‡Department of Mechanical Engineering, Sophia University, 7-1, Kioicho, Chiyoda-ku, Tokyo 102 Japan.

is quite unsatisfactory from the viewpoint of practice. In fact, most of the efficient implementations adopt the long-step step-size choice procedure proposed by Vanderbei, Meketon, and Freedman [43], which determines the next iterate at the point obtained by proceeding a (fixed) proportion $\lambda < 1$ of the way to the boundary. Usually λ is taken to be $0.9 \sim 0.99$ and this is a step-size much greater than the theoretically best bound mentioned above.

The second problem concerns terminating the algorithm and recovering a dual optimal solution. In contrast to the primal-dual interior point algorithms [19], [20], [22], [27], [28], the affine scaling algorithm generates sequence only in the space of the primal problem, and there is no dual feasible solution available during the iterations. This is a serious disadvantage of the algorithm, since without dual feasible solutions, it is very difficult to know whether the iterates come close to the optimal face to stop iterations. To remedy this drawback, we compute a quantity so called the dual estimate that satisfies only the linear equality constraints of the dual problem [4], [8], [43], expecting its convergence to an optimal solution of the dual problem. If the convergence is confirmed theoretically under realistic assumptions from the viewpoint of implementation, we obtain meaningful (and hopefully, powerful) stopping criteria by computing the duality gap. To date, convergence of the dual estimate is only shown under nondegeneracy assumptions [4], [8], [43], or, if we do not require any nondegeneracy assumption, for a short-step version [37], [42] or for the continuous version [3].

Thus, while most of the implementations use the long-step version of the algorithm, almost nothing is proved on this version without requiring nondegeneracy conditions.

In this paper we establish new convergence results on the long-step version of the affine scaling algorithm obtained by developing the approach taken in [37]–[39], [42]. Specifically, we will show that, without requiring any nondegeneracy conditions, any step-size choice up to $\lambda = 2/3$ ensures global convergence of the iterates to an interior point of the optimal face as well as the dual estimates to the analytic center of the dual optimal face, while the asymptotic reduction rate of the value of objective function is $1/3$ (in the case of $\lambda = 2/3$), not dependent on the dimension of the problem! These results seem to make it possible to overcome the two major difficulties in implementation discussed above by adopting the strategy choosing $\lambda = 2/3$ if the reduction of the objective function becomes small.

We also give an example to demonstrate that $2/3$ is the largest step-size that ensures convergence of the dual estimate to the analytic center of the dual optimal face as long as we move with a fixed ratio towards the boundary at each iteration; thus showing that our bound is tight. (See also Hall and Vanderbei [16], who obtained a stronger result on the tightness of $2/3$.)

This development was triggered by the work of Dikin [10], who proved convergence of the dual estimates when applying the algorithm to homogeneous linear programming (LP) problems with unique optimal solutions with $\lambda = 1/2$. Dikin obtained this result by analyzing the reduction of the Karmarkar potential functions associated with the LP problems, which is similar in spirit to our approach. In fact, after releasing the first version of this paper [41], we received the paper by Dikin [9], where he proved the global convergence of the primal iterates and the dual estimates with $\lambda = 1/2$. The proofs of Dikin and ours are similar as are the results. The major difference between the two is the inequalities to estimate the reduction of the local Karmarkar potential function.

Fairly speaking, the results obtained here are surprising to the authors, who did

not expect to improve the global convergence results [38] of the affine scaling algorithm based on the idea of the local Karmarkar potential function.

1. Problem and main result. We deal with the dual standard form linear programming problem $\langle \mathrm{D} \rangle$:

$$\begin{aligned} &\text{minimize } c^t x\ , \quad \text{subject to } x \in \mathcal{P}, \\ &\quad \mathcal{P} = \{x \in \mathbf{R}^n | A^t x - b \geq 0\}, \\ &A = (a_1, \ldots, a_m) \in \mathbf{R}^{n \times m},\ c \in \mathbf{R}^n,\ b \in \mathbf{R}^m. \end{aligned} \tag{1.1}$$

We study global and local convergence properties of the affine scaling algorithm for the dual standard form linear programming problems [1] under the following assumptions.

Assumption 1. The feasible region $\mathcal{P}$ has an interior point and $\mathrm{Rank}(A) = n$;

Assumption 2. $c \neq 0$.

We do not require any condition on degeneracy. Note also that the boundedness of the optimal face is not assumed.

We use the following notations. For a vector v, we denote by $[v]$ the diagonal matrix whose diagonal entries are elements of v. We denote the slack variables $A^t x - b$ by $\xi(x)$, and define the "metric" matrix $G(x)$ for the affine scaling algorithm as follows:

$$G(x) = A[\xi(x)]^{-2} A^t. \tag{1.2}$$

$\mathbf{1}$ and I denote the vector of all ones and the identity matrix of proper dimension, respectively. We use $\|\cdot\|$ (without subscript) for the 2 norm. Given a vector v of proper dimension, we denote by $\sigma(v)$ the largest component of v. For the sequence $\{x^{(\nu)}\}$ ($\nu = 0, 1, \ldots$; $x^{(\nu)} \in \mathbf{R}^n$), we abbreviate $\{f(x^{(\nu)})\}$, $\{g(x^{(\nu)})\}$, etc. as $\{f^{(\nu)}\}$, $\{g^{(\nu)}\}$, etc. We denote by x^+ the new point obtained by performing one iterative step at the point $x \in \mathbf{R}^n$, and use f^+, g^+, etc. to denote $f(x^+), g(x^+)$, etc. We do not indicate arguments of functions when they are obvious from the context.

Let $x^{(\nu)}$ be an interior point of the polyhedron $\mathcal{P}$. The iteration of the long-step affine scaling algorithm for the dual standard form problem $\langle \mathrm{D} \rangle$ is defined as follows:

$$x^{(\nu+1)} = x^{(\nu)} - \lambda^{(\nu)} \frac{G(x^{(\nu)})^{-1} c}{\sigma([\xi^{(\nu)}]^{-1} A^t G(x^{(\nu)})^{-1} c)}. \tag{1.3}$$

It is easy to check that $x^{(\nu+1)}$ is also an interior point of $\mathcal{P}$ if $0 \leq \lambda^{(\nu)} < 1$, so that the iteration can be continued recursively. Since $G(x)$ is a positive definite matrix, the algorithm is a descendant method for $c^t x$. If $\sigma([\xi^{(\nu)}]^{-1} A^t G(x^{(\nu)})^{-1} c)$ becomes zero or negative, then we stop the iteration, since this means that the problem does not have an optimal solution. To exclude this trivial case, we assume that $\sigma([\xi^{(\nu)}]^{-1} A^t G(x^{(\nu)})^{-1} c) > 0$ throughout the iterations.

The vector

$$\tilde{y}^{(\nu)} = [\xi^{(\nu)}]^{-2} A^t G(x^{(\nu)})^{-1} c \tag{1.4}$$

satisfies the equality constraints of the dual problem of $\langle \mathrm{D} \rangle$, and is referred to as the "dual estimate." The dual estimate plays a role similar to the shadow price in the simplex algorithm [32]. We expect that the quantity converges to an optimal solution of the dual problem.

The goal of this paper is to prove the following theorem.

THEOREM 1.1. *Let* $\langle \mathrm{D} \rangle$ *be a linear programming problem satisfying assumptions* 1 *and* 2, *and let* $\{x^{(\nu)}\}$ *be a sequence generated by the affine scaling algorithm* (1.3) *applied to* $\langle \mathrm{D} \rangle$ *with the step-size*

$$0 < \lambda_{\min} \leq \lambda^{(\nu)} \leq 2/3, \tag{1.5}$$

where $\lambda_{\min}$ *is a positive constant. If* $\langle \mathrm{D} \rangle$ *has an optimal solution, the sequence converges to an interior point* x^* *of the optimal face with* $\|x^{(\nu)} - x^*\| = O(c^t x^{(\nu)} - c^t x^*)$, *where the reduction rate of* $c^t x^{(\nu)} - c^t x^*$ *is* $1 - \lambda^{(\nu)}$ *asymptotically, while the dual estimate* $[\xi^{(\nu)}]^{-2} A^t G(x^{(\nu)})^{-1} c$ *converges to the analytic center of the optimal face of the dual problem of* $\langle \mathrm{D} \rangle$. *On the other hand, if* $\langle \mathrm{D} \rangle$ *does not have an optimal solution, the sequence is unbounded with* $c^t x^{(\nu)} \to -\infty$ *as* $\nu \to \infty$.

It is also worth noting that the pair $(x^{(\nu)}, \tilde{y}^{(\nu)})$ of the iterate and the dual estimate converges to a pair of optimal solutions of $\langle \mathrm{D} \rangle$ and its dual problem satisfying a strict complementarity condition. Hence, the algorithm can be used to determine the optimal faces of $\langle \mathrm{D} \rangle$ and its dual problem [12], [25].

The proof of Theorem 1.1 will be given in §4, and we will show in §5 that this bound is tight. Our result can be directly applied to show global convergence of the affine scaling algorithm for the standard form problems as well. A brief explanation for this is given in the Appendix of [39].

We introduce some more notations related to polyhedra, together with a few specific concepts. See [32] for basic theory of polyhedra.

(1) We use the letters $A, B, \ldots, Z$ to denote the faces of $\mathcal{P}$. We do not treat the empty set as a face. We denote by $\mathcal{S}$ the optimal face of $\langle \mathrm{D} \rangle$ if it exists. For a face $\mathcal{F}$ of $\mathcal{P}$, we denote by $E(\mathcal{F})$ the set of indices of the constraints that are always satisfied with equality on the face. We sometimes abbreviate $E(\mathcal{F})$ as E when the face $\mathcal{F}$ which associates with the notation E is obvious from the context.

(2) Given a set $F \subset \{1, \ldots, m\}$ of indices, we denote by A_F, b_F the matrix and the vector composed of the corresponding coefficient vectors and constants. We use $\xi_F(x)$ for $A_F^t x - b_F$. Analogously, for a vector v, we denote by v_F the vector composed of the part of v associated with F.

(3) A point x on a face $\mathcal{F}$ of $\mathcal{P}$ is referred to as an "interior point of $\mathcal{F}$" if $\xi_{E(\mathcal{F})}(x) = 0$ and $\xi_i(x) > 0$ $(i \notin E(\mathcal{F}))$. The interior point of a vertex is the vertex itself. The face $\mathcal{F}$ is the smallest face (as a set) among the faces that contain the point x as their element.

(4) For an index set F, we use $|F|$ to denote its cardinality. If F is a (proper) subset of another index set F', we denote $F \subseteq (\subset) F'$. Then we denote by $F' - F$ the set consisting of the indices that belong to F' but not to F. The complement of F, which is defined as $\{1, \ldots, m\} - F$, is written by F^c.

(5) A face $\mathcal{F}$ of $\mathcal{P}$ is referred to as a "dual degenerate face" if the objective function $c^t x$ is constant on the face. We include vertices also as dual degenerate faces. Dual degenerate faces are characterized as follows (Proposition 3.2 of [39]): A face $\mathcal{F}$ of $\mathcal{P}$ is a dual degenerate face if and only if $c \in \mathrm{Im}(A_{E(\mathcal{F})})$.

By definition, the optimal face is a dual degenerate face. We note that a dual degenerate face does not necessarily contain an optimal solution. Any face $\mathcal{F}$ of $\mathcal{P}$ that is contained in a hyperplane $\{x | c^t x = c_0\}$ with appropriate c_0 is a dual degenerate face. For example, every vertex is a dual degenerate face.

We conclude this section with a basic and general result for the iteration sequence of (1.3), obtained as a direct consequence of Lemma 5.1 of [38].

LEMMA 1.2. *Let $\{x^{(\nu)}\}$ be a sequence generated by the iteration* (1.3) *of the affine scaling algorithm with the step-size $\lambda^{(\nu)}$ bounded below by a positive constant $\lambda_{\min}$. If the monotone decreasing sequence $\{c^t x^{(\nu)}\}$ has the limiting value c^∞, then the following is true.*

(1) *The sequence $\{x^{(\nu)}\}$ converges to an interior point x^* of a dual degenerate face $\mathcal{X}$ with*

$$\frac{c^t x^{(\nu)} - c^\infty}{\|\xi_{E(\mathcal{X})}(x^{(\nu)})\|} > \eta > 0 \tag{1.6}$$

and $\|x^{(\nu)} - x^*\| = O(c^t x^{(\nu)} - c^\infty)$, where η is a constant.

(2) *The value $c^t x^{(\nu)}$ of the objective function converges linearly to c^∞ asymptotically, where the reduction rate is less than $(1 - \lambda_{\min}/|E(\mathcal{X})|^{1/2})$.*

Proof. We rewrite the iteration (1.3) as follows:

$$x^{(\nu+1)} = x^{(\nu)} - \lambda^{(\nu)} \frac{G(x^{(\nu)})^{-1}c}{\sigma([\xi^{(\nu)}]^{-1}A^t G(x^{(\nu)})^{-1}c)} = x^{(\nu)} - \mu^{(\nu)} \frac{G(x^{(\nu)})^{-1}c}{\{c^t G(x^{(\nu)})^{-1}c\}^{\frac{1}{2}}}, \tag{1.7}$$

where

$$\mu^{(\nu)} = \lambda^{(\nu)} \frac{\{c^t G(x^{(\nu)})^{-1}c\}^{\frac{1}{2}}}{\sigma([\xi^{(\nu)}]^{-1}A^t G(x^{(\nu)})^{-1}c)}. \tag{1.8}$$

We show that $\mu^{(\nu)}$ is bounded below by a positive constant. Since $\sigma(\cdot) \le \|\cdot\|_\infty$, we have

$$\begin{aligned} \mu^{(\nu)} &= \lambda^{(\nu)} \frac{\{c^t G^{-1}c\}^{\frac{1}{2}}}{\sigma([\xi^{(\nu)}]^{-1}A^t G^{-1}c)} \ge \lambda^{(\nu)} \frac{\{c^t G^{-1}c\}^{\frac{1}{2}}}{\|[\xi^{(\nu)}]^{-1}A^t G^{-1}c\|_\infty} \\ &= \lambda^{(\nu)} \frac{\|[\xi^{(\nu)}]^{-1}A^t G^{-1}c\|}{\|[\xi^{(\nu)}]^{-1}A^t G^{-1}c\|_\infty} \ge \lambda^{(\nu)} \ge \lambda_{\min}. \end{aligned} \tag{1.9}$$

This is all what we need to apply Lemma 5.1 of [38], from which the lemma immediately follows. □

Thus, under the very weak condition, the sequence generated by (1.3) converges to an interior point of a dual degenerate face (or diverges with $c^t x^{(\nu)} \to -\infty$).

2. Asymptotic search direction of the affine scaling algorithm. As we showed above, the limiting point of the iteration sequence of (1.3) lies in the interior of a dual degenerate face of $\mathcal{P}$. Hence in order to study properties of the limiting point, we need to obtain an asymptotic formula of the search direction when the sequence approaches an interior point of a dual degenerate face. In this section we derive an expression of the search direction in the space of the slack variables which is useful for this purpose.

Let y be a vector in $\mathbf{R}^m$ which satisfies the equality $Ay = c$. The iteration (1.3) is written as follows in the space of the slack variables $\xi(x)$:

$$\begin{aligned} \xi(x^{(\nu+1)}) &= A^t x^{(\nu+1)} - b = A^t x^{(\nu)} - b - \lambda^{(\nu)} A^t \frac{G(x^{(\nu)})^{-1}c}{\sigma([\xi^{(\nu)}]^{-1}A^t G(x^{(\nu)})^{-1}c)} \\ &= A^t x^{(\nu)} - b - \lambda^{(\nu)} A^t \frac{G(x^{(\nu)})^{-1}Ay}{\sigma([\xi^{(\nu)}]^{-1}A^t G(x^{(\nu)})^{-1}c)} \\ &= \xi(x^{(\nu)}) - \lambda^{(\nu)}[\xi^{(\nu)}] \frac{P(x^{(\nu)})\alpha(x^{(\nu)})}{\sigma(P(x^{(\nu)})\alpha(x^{(\nu)}))} \\ &= \xi(x^{(\nu)}) - \lambda^{(\nu)} \Delta\xi^{(\nu)}, \end{aligned} \tag{2.1}$$

where

$$\Delta\xi^{(\nu)} = [\xi^{(\nu)}]\frac{P(x^{(\nu)})\alpha(x^{(\nu)})}{\sigma(P(x^{(\nu)})\alpha(x^{(\nu)}))}, \qquad P(x) = [\xi(x)]^{-1}A^tG(x)^{-1}A[\xi(x)]^{-1}, \tag{2.2}$$

and $\alpha(x) = [\xi(x)]y$. We refer to $\Delta\xi^{(\nu)}$ as the displacement vector of the affine scaling algorithm in the space of the slack variables. Note that $P(x)$ is a projection matrix. Multiplying both sides of (2.1) by $[\xi^{(\nu)}]^{-1}$, we have

$$[\xi^{(\nu)}]^{-1}\xi^{(\nu+1)} = \mathbf{1} - \lambda^{(\nu)}\frac{P^{(\nu)}\alpha^{(\nu)}}{\sigma(P^{(\nu)}\alpha^{(\nu)})}. \tag{2.3}$$

This means that the value of each slack variable $\xi_i^{(\nu+1)}$ at the next iteration is bounded below by $(1-\lambda^{(\nu)})\xi_i^{(\nu)}$.

In [39], an asymptotic formula for the affine scaling algorithm near the boundary of feasible region is investigated intensively. From Lemmas 4.1 and 4.2 of the paper, we obtain the following lemma. (A long and cumbersome matrix manipulation is necessary to derive this lemma, but we think that the result is rather simple and natural.)

LEMMA 2.1. *Let $\mathcal{F}$ be a dual degenerate face, and let y be a vector such that*

$$Ay = A_{E(\mathcal{F})}y_{E(\mathcal{F})} = c, \quad y_{E^c(\mathcal{F})} = 0, \tag{2.4}$$

the existence of which follows from (5) *of* §1, *and, for any $x \in \mathcal{P}$, let*

$$\alpha(x) = [\xi(x)]y. \tag{2.5}$$

Then the displacement vector $\Delta\xi^{(\nu)}$ of the algorithm in the space of the slack variables is written as follows:

$$\begin{aligned}\Delta\xi^{(\nu)} &= \begin{pmatrix}\Delta\xi_E^{(\nu)}\\ \Delta\xi_{E^c}^{(\nu)}\end{pmatrix} = [\xi^{(\nu)}]\frac{P^{(\nu)}\alpha^{(\nu)}}{\sigma(P^{(\nu)}\alpha^{(\nu)})}\\ &= \begin{pmatrix}[\xi_E^{(\nu)}]\widehat{P}_{EE}^{(\nu)}(I - Q_{EE}^{(\nu)})\widehat{P}_{EE}^{(\nu)}\\ -[\xi_{E^c}^{(\nu)}]Q_{E^cE}^{(\nu)}\end{pmatrix}\frac{\alpha_E^{(\nu)}}{\sigma(P^{(\nu)}\alpha^{(\nu)})},\end{aligned} \tag{2.6}$$

where $E = E(\mathcal{F})$. Here $\widehat{P}_{EE}^{(\nu)} = \widehat{P}_{EE}(\xi_E(x^{(\nu)}))$, $\widehat{P}_{EE}(\xi_E)$ is a projection matrix onto $\mathrm{Im}([\xi_E]^{-1}A_E^t)$ with respect to the Euclidean metric satisfying

$$\widehat{P}_{EE}\mathbf{1} = \mathbf{1}, \tag{2.7}$$

and $Q_{EE}^{(\nu)} = Q_{EE}(\xi(x^{(\nu)}))$ and $Q_{E^cE}^{(\nu)} = Q_{E^cE}(\xi(x^{(\nu)}))$ are matrices whose norm is bounded by $\|Q_{EE}(\xi(x))\| = O(\|\xi_E(x)\|^2)$ and $\|Q_{E^cE}(\xi(x))\| = O(\|\xi_E(x)\|)$ when x converges to an interior point of $\mathcal{F}$.

Proof. The formula (2.6) follows by substituting the expressions (4.4) and (4.5) of $P(x)$ in Lemma 4.1 of [39] into (2.2) with $F := E(\mathcal{F})$, and by taking note of $\alpha_{E^c} = 0$. We may regard the matrices $\widehat{P}_{FF}$ and Q_{FF} in (4.4), (4.5) of [39] exactly as the same matrices as those appearing in (2.6) above. It is not written explicitly in Lemma 4.1 of [39] that $\widehat{P}_{EE}(\xi_E)$ is a projection matrix onto $\mathrm{Im}([\xi_E]^{-1}A_E^t)$, but we can confirm this fact by a simple calculation from its definition.

The order of $\|Q_{EE}\|$ and $\|Q_{E^cE}\|$ follows from Lemma 4.2 of [39] using the fact that the quantity Φ_E appearing in Lemma 4.2 has the order of $\|\xi_E\|$ when x converges to an interior point of the face $\mathcal{F}$. Finally, we see (2.7) easily, since we have

$$\mathbf{1} = [\xi_E]^{-1}\xi_E = [\xi_E]^{-1}A_E^t u, \tag{2.8}$$

choosing appropriate u and $\widehat{P}_{EE}(\xi_E)$ is a projection matrix onto $\mathrm{Im}([\xi_E]^{-1}A_E^t)$. □

It is important to say something about the limiting scaled search direction of the affine scaling algorithm here. Now suppose that we have the limiting point in the interior of the face $\mathcal{F}$. Then the lemma tells us that the $E(\mathcal{F})$ part of the scaled search direction $[\xi_E^{(\nu)}]^{-1}\Delta\xi_E^{(\nu)}$ becomes proportional to $\widehat{P}_{EE}^{(\nu)}\alpha_E^{(\nu)}$ asymptotically, since $Q_{EE}^{(\nu)}$ converges to zero. What is $\widehat{P}_{EE}^{(\nu)}\alpha_E^{(\nu)}$? Interestingly, this direction is identical to the scaled search direction of the affine scaling algorithm for the following homogeneous problem:

$$\min y_E^t\tilde{\xi}_E \quad \text{subject to } A_E^t z + \tilde{\xi}_E = 0, \quad \tilde{\xi}_E \geq 0 \tag{2.9}$$

at $\tilde{\xi}_E = \xi_E^{(\nu)}$, which can be interpreted as a search direction of a version of Karmarkar's algorithm with conical formulation [5], [13], [35], [36], [45]. This is why analysis of the projective scaling algorithm comes in the analysis of the affine scaling algorithm, and is one of the key observations on which our analysis is based.

3. Basic lemmas. In this chapter we provide basic lemmas. In particular, the first lemma, which is an improvement of the result obtained by Dikin [10], plays a very important role in the proof of the global convergence of the iterates (Theorem 4.8) and the dual estimates (Theorem 4.9).

LEMMA 3.1. *Let*

$$H_k(\tilde{\beta},\tilde{\lambda}) = k\log(1-\tilde{\lambda}\|\tilde{\beta}\|^2) - \sum_{i=1}^{k}\log(1-\tilde{\lambda}\tilde{\beta}_i), \tag{3.1}$$

and

$$T_k(\tilde{\beta},\tilde{\lambda}) = \frac{k\tilde{\lambda}}{k-\tilde{\lambda}}\|\tilde{\beta}-\frac{\mathbf{1}}{k}\|^2\left(-k+\frac{\tilde{\lambda}}{2(1-\tilde{\lambda}\sigma(\tilde{\beta}))}\right), \tag{3.2}$$

which are well-defined on the set

$$\Omega_0 = \{(\tilde{\beta},\tilde{\lambda}) \mid \tilde{\beta}\in\mathbf{R}^k, \quad \tilde{\lambda}\in\mathbf{R}, \quad \tilde{\beta}^t\mathbf{1}=1, \quad \tilde{\lambda}\|\tilde{\beta}\|^2<1, \quad 0<\tilde{\lambda}\sigma(\tilde{\beta})<1\}. \tag{3.3}$$

Then, H_k is bounded as follows over Ω_0:

$$H_k(\tilde{\beta},\tilde{\lambda}) \leq T_k(\tilde{\beta},\tilde{\lambda}). \tag{3.4}$$

Furthermore, if

$$(\tilde{\beta},\tilde{\lambda}) \in \Omega_1 \equiv \{(\tilde{\beta},\tilde{\lambda})\in\Omega_0 | \tilde{\lambda}\sigma(\tilde{\beta}) \leq 2/3\}, \tag{3.5}$$

we have

$$H_k(\tilde{\beta},\tilde{\lambda}) \leq T_k(\tilde{\beta},\tilde{\lambda}) \leq 0, \tag{3.6}$$

where $T_k(\tilde{\beta}, \tilde{\lambda}) = 0$ holds if and only if $\tilde{\beta} = \mathbf{1}/k$.

Proof. We introduce a new variable $\gamma = \tilde{\beta} - \mathbf{1}/k$. Then we have

$$1 - \tilde{\lambda}\tilde{\beta}_i = 1 - \frac{\tilde{\lambda}}{k} - \tilde{\lambda}\gamma_i, \qquad 1 - \tilde{\lambda}\|\tilde{\beta}\|^2 = 1 - \frac{\tilde{\lambda}}{k} - \tilde{\lambda}\|\gamma\|^2. \tag{3.7}$$

Putting

$$\theta = \frac{k\tilde{\lambda}}{k - \tilde{\lambda}} \tag{3.8}$$

and taking note that $0 < \tilde{\lambda} < k$ in Ω_0 (this follows from $\tilde{\beta}^t\mathbf{1} = 1$), we obtain

$$H_k(\tilde{\beta}, \tilde{\lambda}) = k\log(1 - \theta\|\gamma\|^2) - \sum_{i=1}^{k}\log(1 - \theta\gamma_i). \tag{3.9}$$

Note also that

$$\sigma(\gamma) = \sigma(\tilde{\beta}) - \frac{1}{k} \tag{3.10}$$

and

$$\gamma^t\mathbf{1} = 0, \tag{3.11}$$

which follow from $\tilde{\beta}^t\mathbf{1} = 1$. By using the well-known inequalities on the logarithmic function

$$\log(1 - \delta) \le -\delta \quad (\delta < 1), \tag{3.12}$$

$$\begin{aligned}
&\sum_{i=1}^{k}\log(1 - \zeta_i) \\
&= \sum_{i:\zeta_i > -\sigma(\zeta)}\left(-\zeta_i - \frac{\zeta_i^2}{2} - \frac{\zeta_i^3}{3} - \cdots\right) + \sum_{i:\zeta_i \le -\sigma(\zeta)}\log(1 - \zeta_i) \\
&\ge \sum_{i:\zeta_i > -\sigma(\zeta)}\left(-\zeta_i - \frac{|\zeta_i|^2}{2} - \frac{|\zeta_i|^3}{2} - \cdots\right) + \sum_{i:\zeta_i \le -\sigma(\zeta)}\log(1 - \zeta_i) \\
&\ge \sum_{i:\zeta_i > -\sigma(\zeta)}\left(-\zeta_i - \frac{\zeta_i^2}{2(1 - |\zeta_i|)}\right) + \sum_{i:\zeta_i \le -\sigma(\zeta)}\left(-\zeta_i - \frac{\zeta_i^2}{2(1 - \sigma(\zeta))}\right) \\
&\ge -\zeta^t\mathbf{1} - \frac{\|\zeta\|^2}{2(1 - \sigma(\zeta))} \quad (\zeta \in \mathbf{R}^k, \quad 0 \le \sigma(\zeta) < 1),
\end{aligned} \tag{3.13}$$

we see that H_k is bounded above by

$$\begin{aligned}
H_k(\tilde{\beta}, \tilde{\lambda}) &= k\log(1 - \theta\|\gamma\|^2) - \sum_{i=1}^{k}\log(1 - \theta\gamma_i) \\
&\le -k\theta\|\gamma\|^2 + \frac{\theta^2\|\gamma\|^2}{2(1 - \theta\sigma(\gamma))} \\
&= \theta\|\gamma\|^2\left(-k + \frac{\theta}{2(1 - \theta\sigma(\gamma))}\right)
\end{aligned} \tag{3.14}$$

provided that $\theta\sigma(\gamma) < 1$ and $\theta\|\gamma\|^2 < 1$. These conditions are equivalent to $\tilde{\lambda}\sigma(\tilde{\beta}) < 1$ and $\tilde{\lambda}\|\tilde{\beta}\|^2 < 1$, which are always satisfied on Ω_0. Substituting the definition of γ and θ into the rightmost side of (3.14) to represent it in terms of $\tilde{\beta}$ and $\tilde{\lambda}$, we obtain

$$H_k(\tilde{\beta}, \tilde{\lambda}) \leq \frac{k\tilde{\lambda}}{k - \tilde{\lambda}} \|\tilde{\beta} - \frac{\mathbf{1}}{k}\|^2 \left(-k + \frac{\tilde{\lambda}}{2(1 - \tilde{\lambda}\sigma(\tilde{\beta}))} \right) = T_k(\tilde{\beta}, \tilde{\lambda}). \tag{3.15}$$

To see the lemma, it is enough to check that

$$-k + \frac{\tilde{\lambda}}{2(1 - \tilde{\lambda}\sigma(\tilde{\beta}))} \leq 0 \tag{3.16}$$

is satisfied over Ω_1, where the equality holds only if $\tilde{\beta} = \mathbf{1}/k$. This can be done as follows:

$$-k + \frac{\tilde{\lambda}}{2(1 - \tilde{\lambda}\sigma(\tilde{\beta}))} \leq -k + \frac{2/(3\sigma(\tilde{\beta}))}{2(1 - 2/3)} \leq -k + \frac{1}{\sigma(\tilde{\beta})} \leq 0, \tag{3.17}$$

by using $\sigma(\tilde{\beta}) \geq 1/k$, which completes the proof. □

The function H_k was introduced independently by Dikin and by Tsuchiya and Tanabe in the context of studying the convergence of the dual estimates in a long-step affine scaling algorithm applied to homogeneous LP problems with unique optimal solutions [10] and of analyzing the local convergence of the dual estimates in a short-step affine scaling algorithm under the assumption of uniqueness of the optimal solution [42], respectively.

The second lemma is used to observe asymptotic behavior of the dual estimates in the proof of Lemma 4.7 and Theorem 4.9.

LEMMA 3.2. *Let $\mathcal{F}$ be a dual degenerate face, and choose y such that*

$$Ay = c, \quad y_{E^c(\mathcal{F})} = 0. \tag{3.18}$$

If x is an interior point of $\mathcal{P}$, then

$$\hat{y}(x) = (\hat{y}_E, \hat{y}_{E^c}) = ([\xi_E(x)]^{-1} \widehat{P}_{EE}(\xi_E(x))[\xi_E(x)]y_E, 0) \tag{3.19}$$

is a solution of the equality

$$A\hat{y} = c, \tag{3.20}$$

where $E = E(\mathcal{F})$.

Proof. It is enough to show $A_E\hat{y}_E = c$. This is equivalent to

$$A_E[\xi_E]^{-1}(I - \widehat{P}_{EE})[\xi_E]y = 0, \tag{3.21}$$

which holds obviously because of the definition of $\widehat{P}_{EE}$. □

The third lemma is a characterization of the analytic center of the optimal face of the dual problem of ⟨D⟩ used in the proof of Theorem 4.9.

LEMMA 3.3. *If ⟨D⟩ has an optimal solution, the analytic center of the optimal face of the dual problem to ⟨D⟩ is the unique solution y^* of the following system of equations*:

$$[y_{E(\mathcal{S})}]^{-1}\mathbf{1} + A^t_{E(\mathcal{S})}u = 0, \quad A_{E(\mathcal{S})}y_{E(\mathcal{S})} = c, \quad y_{E^c(\mathcal{S})} = 0, \tag{3.22}$$

where $\mathcal{S}$ is the optimal face of $\langle \mathrm{D} \rangle$.

Proof. Due to the strict complementarity, the optimal face of the dual problem to $\langle \mathrm{D} \rangle$ is written explicitly as follows:

$$A_{E(\mathcal{S})} y_{E(\mathcal{S})} = c, \quad y_{E(\mathcal{S})} \geq 0, \quad y_{E^c(\mathcal{S})} = 0. \tag{3.23}$$

The analytic center of (3.23) is defined as the unique optimal solution for the following strictly convex optimization problem:

$$\min \; - \sum_{i \in E(\mathcal{S})} \log y_i \quad \text{subject to } A_E y_E = c, \quad y \geq 0. \tag{3.24}$$

The system of (3.22) is obtained immediately from the Karush–Kuhn–Tucker condition for the optimal solution for this problem. □

4. Convergence results. Now we are ready to prove the main results. By moving a ratio $\lambda^{(\nu)}$ up to two-thirds of the way toward the boundary at νth iteration, we will show convergence of the iterates to an interior point of the optimal face of $\langle \mathrm{D} \rangle$ (Theorem 4.8), convergence of the dual estimates (1.4) to the analytic center of the optimal face of the dual problem of $\langle \mathrm{D} \rangle$ (Theorem 4.9), and asymptotic linear convergence of the objective function, where the asymptotic reduction rate is $1 - \lambda^{(\nu)}$ (Theorem 4.10). Theorem 1.1 follows immediately from Theorems 4.8–4.10.

4.1. Outline of the proofs. Let $\{x^{(\nu)}\}$ be a sequence generated by the iteration (1.3) of the affine scaling algorithm with step-size $\{\lambda^{(\nu)}\}$ under the assumption of Lemma 2.1, i.e., $0 < \lambda_{\min} \leq \lambda^{(\nu)} \leq \lambda_{\max} < 1$ and boundedness of $\{c^t x^{(\nu)}\}$. Due to Lemma 2.1, the sequence converges to an interior point of a dual degenerate face. We denote by x^* the limiting point and by $\mathcal{X}$ the face that contains x^* as its interior point, and denote by c^∞ the limiting value of the objective function. We use k for $|E(\mathcal{X})|$.

The local Karmarkar potential function which was introduced in [39] and was used to prove global convergence of a short-step version in [38] plays an important role in the proof of the main theorems. In this subsection we briefly outline how this function is used in the proof. The definition of the local Karmarkar potential function associated with the dual degenerate face $\mathcal{X}$ is given by

$$f_{\mathcal{X}}(x) \equiv |E(\mathcal{X})| \log(c^t x - c^\infty) - \sum_{i \in E(\mathcal{X})} \log \xi_i(x). \tag{4.1}$$

We observe that $f_{\mathcal{X}}(x)$ is a homogeneous function in $\xi_{E(\mathcal{X})}(x)$. Let us take y such that

$$A_{E(\mathcal{X})} y_{E(\mathcal{X})} = c, \qquad y_{E^c(\mathcal{X})} = 0, \tag{4.2}$$

the existence of which is ensured since $\mathcal{X}$ is a dual degenerate face. Since $A_E^t x^* = b_E$, we have

$$\begin{aligned} c^t x - c^\infty &= c^t (x - x^*) = y^t A^t (x - x^*) \\ &= y_E^t A_E^t (x - x^*) = y_E^t (A_E^t x - b_E) \\ &= y_E^t \xi_E = y^t \xi. \end{aligned} \tag{4.3}$$

With this relation, we can rewrite $f(x)$ as

$$f_{\mathcal{X}}(x) = |E(\mathcal{X})| \log y_{E(\mathcal{X})}^t \xi_{E(\mathcal{X})} - \sum_{i \in E(\mathcal{X})} \log \xi_i(x), \tag{4.4}$$

thus we see that $f_{\mathcal{X}}(x)$ is a homogeneous function.

If $\mathcal{X}$ is the optimal face $\mathcal{S}$, the function $f_{\mathcal{S}}$ is bounded below by a constant, because, in this case, we can choose y such that $y_{E(\mathcal{S})} > 0$ due to strict complementarity. This is a crucial property of the local Karmarkar potential function associated with the optimal face, which holds if and only if $\mathcal{X}$ is the optimal face.

Now, we are ready to outline the proof of the main results. In view of Lemma 1.2 and well-known inequality between the arithmetic mean and geometric mean, this function is bounded below by a constant, since

(4.5)
$$\exp(f_{\mathcal{X}}(x^{(\nu)})) = \frac{(c^t x^{(\nu)} - c^{\infty})^{|E(\mathcal{X})|}}{\prod_{i \in E(\mathcal{X})} \xi_i} \geq \left(|E(\mathcal{X})| \frac{(c^t x^{(\nu)} - c^{\infty})}{\|\xi^{(\nu)}_{E(\mathcal{X})}\|_1} \right)^{|E(\mathcal{X})|} \geq (\sqrt{k}\eta)^k$$

holds, where η is the constant appearing in (1.6).

On the other hand, we are able to show the following claim.

CLAIM. *$f_{\mathcal{X}}(x^{(\nu)})$ tends to minus infinity if $\lambda^{(\nu)} \leq 2/3$ and $\mathcal{X} \neq \mathcal{S}$.*

Together with (4.5), this fact immediately implies $\mathcal{X} = \mathcal{S}$, thus global convergence of the sequence when $\lambda^{(\nu)} \leq 2/3$ is obvious if it can be shown. To obtain this claim on $\{f^{(\nu)}_{\mathcal{X}}\}$, we use the function T_k introduced in Lemma 3.1. Roughly speaking, the difference $f_{\mathcal{X}}(x^{(\nu+1)}) - f_{\mathcal{X}}(x^{(\nu)})$ of the potential is approximately bounded by $T_k(\tilde{\beta}^{(\nu)}, \tilde{\lambda}^{(\nu)})$ when ν is sufficiently large, where $\tilde{\beta}^{(\nu)}$ and $\tilde{\lambda}^{(\nu)}$ are functions of $x^{(\nu)}$, and this leads us to the following asymptotic bound for the value of the potential function when ν is sufficiently large (Lemma 4.5):

(4.6)
$$\begin{aligned} f^{(\nu)}_{\mathcal{X}} &= f_{\mathcal{X}}(x^{(\nu_0)}) + \sum_{\tau=\nu_0}^{\nu-1} (f_{\mathcal{X}}(x^{(\tau+1)}) - f_{\mathcal{X}}(x^{(\tau)})) \\ &\leq f_{\mathcal{X}}(x^{(\nu_0)}) + \sum_{\tau=\nu_0}^{\nu-1} T_k(\tilde{\beta}^{(\tau)}, \tilde{\lambda}^{(\tau)}) + M, \end{aligned}$$

where ν_0 and M are constants. We can show that $T_k(\tilde{\beta}^{(\tau)}, \tilde{\lambda}^{(\tau)})$ in the last sum is smaller than a strictly negative constant if $\mathcal{X} \neq \mathcal{S}$ and $\lambda^{(\nu)} \leq 2/3$ (Lemma 4.7), and this proves the claim.

Thus $\mathcal{X} = \mathcal{S}$ is shown, and we move to prove that dual estimates converge to the analytic center of the dual optimal face and that asymptotic convergence rate of the objective function is given by $1 - \lambda^{(\nu)}$. As we mentioned before, $f_{\mathcal{S}}(x^{(\nu)})$ is bounded below by a constant. In view of (4.6), we have

$$T_k(\tilde{\beta}^{(\nu)}, \tilde{\lambda}^{(\nu)}) \to 0, \tag{4.7}$$

which implies, due to Lemma 3.1,

$$\tilde{\beta}^{(\nu)} \to \frac{\mathbf{1}}{k}. \tag{4.8}$$

Once (4.8) is obtained, convergence of the dual estimates to the analytic center of the dual optimal face and the result on convergence rate of the objective function follows without much difficulty. This is the outline of the proof.

We divide the proof of these results into two parts. In §4.2 we obtain preliminary results under the assumption of Lemma 1.2, and then prove the main results adding the condition $\lambda^{(\nu)} \leq 2/3$ in §4.3.

4.2. Preliminary observations. In this subsection we make preliminary observations under the assumption of Lemma 1.2, i.e., $\{x^{(\nu)}\}$ is generated under the step-size satisfying $0 < \lambda_{\min} \leq \lambda^{(\nu)} \leq \lambda_{\max} < 1$ and $\{c^t x^{(\nu)}\}$ is bounded below. From property (1) of Lemma 1.2, we see that

$$\eta \|\xi^{(\nu)}_{E(\mathcal{X})}\| \leq c^t x^{(\nu)} - c^\infty, \tag{4.9}$$

where η is a positive constant. Now, let

$$\alpha = [\xi]y \quad \text{and} \quad \beta = \frac{[\xi]y}{\xi^t y} = \frac{\alpha}{\alpha^t \mathbf{1}} = \frac{[\xi]y}{c^t x^{(\nu)} - c^\infty}. \tag{4.10}$$

By definition, we have $\beta^t \mathbf{1} = 1$, and due to (4.9),

$$\|\beta^{(\nu)}_E\| = \|\beta^{(\nu)}\| \leq \eta^{-1}\|y\|. \tag{4.11}$$

Equations (4.9) and (4.11) are important relations, which will be used frequently in the consecutive analysis. The unit displacement vector $\Delta\xi^{(\nu)}$ of the algorithm in the slack space (cf. (2.2)) is written as

$$\Delta\xi^{(\nu)} = [\xi^{(\nu)}] \frac{P^{(\nu)}\beta^{(\nu)}}{\sigma(P^{(\nu)}\beta^{(\nu)})}. \tag{4.12}$$

From Lemma 2.1, it is easy to see that $P^{(\nu)}\beta^{(\nu)}$ is written as

$$P^{(\nu)}\beta^{(\nu)} = \begin{pmatrix} (P^{(\nu)}\beta^{(\nu)})_{E(\mathcal{X})} \\ (P^{(\nu)}\beta^{(\nu)})_{E^c(\mathcal{X})} \end{pmatrix} = \begin{pmatrix} \widehat{P}^{(\nu)}_{EE}(I - Q^{(\nu)}_{EE})\widehat{P}^{(\nu)}_{EE} \\ Q^{(\nu)}_{E^cE} \end{pmatrix} \beta^{(\nu)}_{E(\mathcal{X})}. \tag{4.13}$$

Since (2.7) holds, we have

$$\mathbf{1}^t \widehat{P}^{(\nu)}_{EE}\beta^{(\nu)}_E = \mathbf{1}^t \beta^{(\nu)}_E = 1. \tag{4.14}$$

$\widehat{P}^{(\nu)}_{EE}\beta^{(\nu)}_E$ plays a very important role in the proof. In fact, we will follow the argument outlined in §4.1 letting

$$(\tilde{\beta}^{(\nu)}, \tilde{\lambda}^{(\nu)}) = (\widehat{P}^{(\nu)}_{EE}\beta^{(\nu)}_E, \lambda^{(\nu)}/\sigma(\widehat{P}^{(\nu)}_{EE}\beta^{(\nu)}_E)) \qquad \text{(cf. (4.6))}. \tag{4.15}$$

LEMMA 4.1. *We have*

$$\frac{1}{|E(\mathcal{X})|} \leq \beta^{(\nu)t}_E \widehat{P}^{(\nu)}_{EE}\beta^{(\nu)}_E \leq \eta^{-2}\|y\|^2, \tag{4.16}$$

and, for sufficiently large ν,

$$1 - M_0(c^t x^{(\nu)} - c^\infty)^2 \leq \frac{\beta^{(\nu)t}P^{(\nu)}\beta^{(\nu)}}{\beta^{(\nu)t}_E \widehat{P}^{(\nu)}_{EE}\beta^{(\nu)}_E} \leq 1 + M_0(c^t x^{(\nu)} - c^\infty)^2, \tag{4.17}$$

where M_0 is a positive constant.

Proof. Since $(\widehat{P}^{(\nu)}_{EE}\beta^{(\nu)}_E)^t \mathbf{1} = 1$ (cf. (4.14)) and $\widehat{P}^{(\nu)}_{EE}$ is a projection matrix, we have

$$\frac{1}{|E(\mathcal{X})|} \leq \|\beta^{(\nu)t}_E \widehat{P}^{(\nu)}_{EE}\|^2 = \beta^{(\nu)t}_E \widehat{P}^{(\nu)}_{EE}\beta^{(\nu)}_E. \tag{4.18}$$

The second inequality in (4.16) follows from (4.11) because

$$\beta_E^{(\nu)t}\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)} \le \|\beta_E^{(\nu)}\|^2 \le (\eta^{-1}\|y\|)^2. \tag{4.19}$$

Now we prove (4.17). From (4.13) and $\beta_{E^c(\mathcal{X})}^{(\nu)} = 0$, we see

$$\beta^{(\nu)t}P^{(\nu)}\beta^{(\nu)} = \beta_E^{(\nu)t}\widehat{P}_{EE}^{(\nu)}(I - Q_{EE}^{(\nu)})\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)}. \tag{4.20}$$

Since

$$\frac{\beta^{(\nu)t}P^{(\nu)}\beta^{(\nu)}}{\beta_E^{(\nu)t}\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)}} = \frac{\beta_E^{(\nu)t}\widehat{P}_{EE}^{(\nu)}(I - Q_{EE}^{(\nu)})\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)}}{\beta_E^{(\nu)t}\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)}} = 1 - \frac{\beta_E^{(\nu)t}\widehat{P}_{EE}^{(\nu)}Q_{EE}^{(\nu)}\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)}}{\beta_E^{(\nu)t}\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)}}, \tag{4.21}$$

and the norm of the last term on the rightmost side is bounded by

$$\begin{aligned}
&\left\| \frac{\beta_E^{(\nu)t}\widehat{P}_{EE}^{(\nu)}Q_{EE}^{(\nu)}\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)}}{\beta_E^{(\nu)t}\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)}} \right\| \\
&\le \|Q_{EE}^{(\nu)}\| \quad \text{(use the fact } \widehat{P}_{EE}^{(\nu)} \text{ is a projection matrix.)} \\
&\le M_1\|\xi_E^{(\nu)}\|^2 \quad \text{(use Lemma 2.1)} \\
&\le M_0(c^t x^{(\nu)} - c^\infty)^2 \quad \text{(use (4.9)),}
\end{aligned} \tag{4.22}$$

where $M_1 > 0$ is an appropriate constant, the relation (4.17) follows. □

LEMMA 4.2. *We have*

$$\frac{1}{|E(\mathcal{X})|} \le \sigma(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)}) \le \eta^{-1}\|y\|, \tag{4.23}$$

$$|(P^{(\nu)}\beta^{(\nu)})_{E(\mathcal{X})} - \widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)}| \le M_2(c^t x^{(\nu)} - c^\infty)^2\mathbf{1} \tag{4.24}$$

and, for sufficiently large ν,

$$1 - M_3(c^t x^{(\nu)} - c^\infty)^2 \le \frac{\sigma((P^{(\nu)}\beta^{(\nu)})_{E(\mathcal{X})})}{\sigma(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)})} \le 1 + M_3(c^t x^{(\nu)} - c^\infty)^2, \tag{4.25}$$

$$\|(P^{(\nu)}\beta^{(\nu)})_{E^c(\mathcal{X})}\| = O(c^t x^{(\nu)} - c^\infty), \tag{4.26}$$

where M_2 *and* M_3 *are positive constants.*

Proof. Since $(\beta_E^{(\nu)t}\widehat{P}_{EE}^{(\nu)})\mathbf{1} = 1$ (cf. (4.14)), we have

$$\frac{1}{|E(\mathcal{X})|} \le \sigma(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)}). \tag{4.27}$$

The second inequality of (4.23) is immediately seen from (4.11).

The relation (4.24) is obtained as follows:

$$\begin{aligned}
&|(P^{(\nu)}\beta^{(\nu)})_{E(\mathcal{X})} - \widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)}| \\
&\le \|Q_{EE}^{(\nu)}\|\|\beta_E^{(\nu)}\|\mathbf{1} \quad \text{(use (4.13))} \\
&\le \|Q_{EE}^{(\nu)}\|\eta^{-1}\|y\|\mathbf{1} \quad \text{(use (4.11))} \\
&\le M_2(c^t x^{(\nu)} - c^\infty)^2\mathbf{1}. \quad \text{(use Lemma 2.1 and (4.9)).}
\end{aligned} \tag{4.28}$$

To show (4.25), it is enough to prove

$$\frac{|\sigma((P^{(\nu)}\beta^{(\nu)})_{E(\mathcal{X})}) - \sigma(\widehat{P}^{(\nu)}_{EE}\beta^{(\nu)}_{E})|}{\sigma(\widehat{P}^{(\nu)}_{EE}\beta^{(\nu)}_{E})} \leq M_3(c^t x^{(\nu)} - c^\infty)^2 \mathbf{1}. \tag{4.29}$$

Due to (4.28), we see that the numerator of (4.29) is bounded by $M_2(c^t x^{(\nu)} - c^\infty)^2$. Together with (4.27), we obtain (4.29), which implies (4.25).

Now we show (4.26). Due to (4.13), we have

$$(P^{(\nu)}\beta^{(\nu)})_{E^c(\mathcal{X})} = Q^{(\nu)}_{E^cE}\beta^{(\nu)}_{E(\mathcal{X})}. \tag{4.30}$$

Since $\|Q^{(\nu)}_{E^cE}\| = O(\|\xi^{(\nu)}_E\|) = O(c^t x^{(\nu)} - c^\infty)$ by Lemma 2.1 and (4.9), and $\|\beta^{(\nu)}_{E(\mathcal{X})}\|$ is bounded by a constant due to (4.9), the relation (4.26) is seen immediately. □

LEMMA 4.3. *If $\{\lambda^{(\nu)}\}$ has an upper bound $\lambda_{\max} < 1$, we have*

$$\frac{c^t x^{(\nu+1)} - c^\infty}{c^t x^{(\nu)} - c^\infty} = 1 - \lambda^{(\nu)}\beta^{(\nu)t}\frac{P^{(\nu)}\beta^{(\nu)}}{\sigma(P^{(\nu)}\beta^{(\nu)})} \geq \eta' > 0 \tag{4.31}$$

and

$$\frac{\xi^{(\nu+1)}_i}{\xi^{(\nu)}_i} = 1 - \lambda^{(\nu)}\frac{(P^{(\nu)}\beta^{(\nu)})_i}{\sigma(P^{(\nu)}\beta^{(\nu)})} \geq 1 - \lambda_{\max} > 0 \tag{4.32}$$

for all ν, where η' is a positive constant.

Proof. We have the first equality of (4.31) as follows:

$$\begin{aligned}
\frac{c^t x^{(\nu+1)} - c^\infty}{c^t x^{(\nu)} - c^\infty} &= \frac{y^t \xi^{(\nu+1)}}{y^t \xi^{(\nu)}} \quad \text{(use (4.3))} \\
&= \frac{y^t \xi^{(\nu)} - \lambda^{(\nu)} y^t \Delta\xi^{(\nu)}}{y^t \xi^{(\nu)}} \quad \text{(use (2.1))} \\
&= 1 - \lambda^{(\nu)}\frac{1}{y^t \xi^{(\nu)}}\frac{y^t[\xi^{(\nu)}]P^{(\nu)}\beta^{(\nu)}}{\sigma(P^{(\nu)}\beta^{(\nu)})} \quad \text{(use (4.12))} \\
&= 1 - \lambda^{(\nu)}\frac{\beta^{(\nu)t}P^{(\nu)}\beta^{(\nu)}}{\sigma(P^{(\nu)}\beta^{(\nu)})} \quad \text{(use (4.10))},
\end{aligned} \tag{4.33}$$

which proves the first equality of (4.31). The second inequality is seen as follows:

$$\begin{aligned}
c^t x^{(\nu+1)} - c^\infty &\geq \eta\|\xi^{(\nu+1)}_{E(\mathcal{X})}\| \quad \text{(use (4.9))} \\
&\geq \eta(1 - \lambda_{\max})\|\xi^{(\nu)}_{E(\mathcal{X})}\| \quad \text{(use the remark following (2.3))} \\
&\geq \frac{\eta(1 - \lambda_{\max})}{\|y_E\|}\|y_E\|\|\xi^{(\nu)}_E\| \\
&\geq \eta' y^t_E \xi^{(\nu)}_E \\
&= \eta'(c^t x^{(\nu)} - c^\infty),
\end{aligned} \tag{4.34}$$

where $\eta' \equiv \eta(1 - \lambda_{\max})/\|y_E\|$ is a positive constant. Similarly, we have

$$\begin{aligned}
\frac{\xi^{(\nu+1)}_i}{\xi^{(\nu)}_i} &= 1 - \lambda^{(\nu)}\frac{(P^{(\nu)}\alpha^{(\nu)})_i}{\sigma(P^{(\nu)}\alpha^{(\nu)})} \quad \text{(use (2.3))} \\
&= 1 - \lambda^{(\nu)}\frac{(P^{(\nu)}\beta^{(\nu)})_i}{\sigma(P^{(\nu)}\beta^{(\nu)})} \\
&\geq 1 - \lambda^{(\nu)} \\
&\geq 1 - \lambda_{\max}.
\end{aligned} \tag{4.35}$$

Thus the relation (4.32) is shown. □

Now, we are ready to analyze the reduction of the local Karmarkar potential function. From Lemma 4.3, the change in the value of the local Karmarkar potential function at the νth iteration is written as follows:

(4.36)
$$\begin{aligned} & f_{\mathcal{X}}(x^{(\nu+1)}) - f_{\mathcal{X}}(x^{(\nu)}) \\ &= |E(\mathcal{X})| \log \frac{c^t x^{(\nu+1)} - c^\infty}{c^t x^{(\nu)} - c^\infty} - \sum_{i \in E(\mathcal{X})} \log \frac{\xi_i^{(\nu+1)}}{\xi_i^{(\nu)}} \\ &= |E(\mathcal{X})| \log \left(1 - \lambda^{(\nu)} \beta^{(\nu)t} \frac{P^{(\nu)} \beta^{(\nu)}}{\sigma(P^{(\nu)} \beta^{(\nu)})}\right) - \sum_{i \in E(\mathcal{X})} \log \left(1 - \lambda^{(\nu)} \frac{(P^{(\nu)} \beta^{(\nu)})_i}{\sigma(P^{(\nu)} \beta^{(\nu)})}\right). \end{aligned}$$

We recall the definition (3.1) of $H_k(\tilde{\beta}, \tilde{\lambda})$, and substitute $(\tilde{\beta}, \tilde{\lambda}) = (\widehat{P}_{EE}^{(\nu)} \beta_E^{(\nu)}, \lambda^{(\nu)} / \sigma(\widehat{P}_{EE}^{(\nu)} \beta_E^{(\nu)}))$, then obtain

(4.37)
$$\begin{aligned} & H_k \left(\widehat{P}_{EE}^{(\nu)} \beta_E^{(\nu)}, \frac{\lambda^{(\nu)}}{\sigma(\widehat{P}_{EE}^{(\nu)} \beta_E^{(\nu)})}\right) \\ &= |E(\mathcal{X})| \log \left(1 - \lambda^{(\nu)} \beta_E^{(\nu)t} \frac{\widehat{P}_{EE}^{(\nu)} \beta_E^{(\nu)}}{\sigma(\widehat{P}_{EE}^{(\nu)} \beta_E^{(\nu)})}\right) - \sum_{i \in E(\mathcal{X})} \log \left(1 - \lambda^{(\nu)} \frac{(\widehat{P}_{EE}^{(\nu)} \beta_E^{(\nu)})_i}{\sigma(\widehat{P}_{EE}^{(\nu)} \beta_E^{(\nu)})}\right). \end{aligned}$$

Making use of the similarity between the rightmost sides of (4.36) and (4.37), below we derive an upper bound and lower bound for $f_{\mathcal{X}}^{(\nu)}$ written in terms of H_k for sufficiently large ν.

LEMMA 4.4. *If $\{\lambda^{(\nu)}\}$ has an upper bound $\lambda_{\max} < 1$, we have*

$$1 - M_4 (c^t x^{(\nu)} - c^\infty)^2 \leq \frac{1 - \lambda^{(\nu)} \dfrac{\beta_E^{(\nu)t} \widehat{P}_{EE}^{(\nu)} \beta_E^{(\nu)}}{\sigma(\widehat{P}_{EE}^{(\nu)} \beta_E^{(\nu)})}}{1 - \lambda^{(\nu)} \dfrac{\beta^{(\nu)t} P^{(\nu)} \beta^{(\nu)}}{\sigma(P^{(\nu)} \beta^{(\nu)})}} \leq 1 + M_4 (c^t x^{(\nu)} - c^\infty)^2 \tag{4.38}$$

and, for $i \in E(\mathcal{X})$,

$$1 - M_5 (c^t x^{(\nu)} - c^\infty)^2 \leq \frac{1 - \dfrac{\lambda^{(\nu)} (\widehat{P}_{EE}^{(\nu)} \beta_E^{(\nu)})_i}{\sigma(\widehat{P}_{EE}^{(\nu)} \beta_E^{(\nu)})}}{1 - \dfrac{\lambda^{(\nu)} (P^{(\nu)} \beta^{(\nu)})_i}{\sigma(P^{(\nu)} \beta^{(\nu)})}} \leq 1 + M_5 (c^t x^{(\nu)} - c^\infty)^2 \tag{4.39}$$

if ν is sufficiently large, where $M_4, M_5 > 0$ are appropriate constants.

Proof. Let

$$\Delta^{(\nu)} \equiv \frac{\beta_E^{(\nu)t} \widehat{P}_{EE}^{(\nu)} \beta_E^{(\nu)}}{\sigma(\widehat{P}_{EE}^{(\nu)} \beta_E^{(\nu)})} - \frac{\beta^{(\nu)t} P^{(\nu)} \beta^{(\nu)}}{\sigma(P^{(\nu)} \beta^{(\nu)})}. \tag{4.40}$$

Then we have

$$\frac{1 - \lambda^{(\nu)} \dfrac{\beta_E^{(\nu)t} \widehat{P}_{EE}^{(\nu)} \beta_E^{(\nu)}}{\sigma(\widehat{P}_{EE}^{(\nu)} \beta_E^{(\nu)})}}{1 - \lambda^{(\nu)} \dfrac{\beta^{(\nu)t} P^{(\nu)} \beta^{(\nu)}}{\sigma(P^{(\nu)} \beta^{(\nu)})}} = 1 - \frac{\lambda^{(\nu)} \Delta^{(\nu)}}{1 - \lambda^{(\nu)} \dfrac{\beta^{(\nu)t} P^{(\nu)} \beta^{(\nu)}}{\sigma(P^{(\nu)} \beta^{(\nu)})}}. \tag{4.41}$$

Due to (4.31) and $\lambda^{(\nu)} \leq 1$, we see

$$\left| \lambda^{(\nu)} \frac{\Delta^{(\nu)}}{1 - \lambda^{(\nu)} \dfrac{\beta^{(\nu)t} P^{(\nu)} \beta^{(\nu)}}{\sigma(P^{(\nu)}\beta^{(\nu)})}} \right| \leq \lambda^{(\nu)} |\eta'^{-1} \Delta^{(\nu)}| \leq |\eta'^{-1} \Delta^{(\nu)}|. \tag{4.42}$$

In view of (4.41), to prove (4.38), it is enough to show

$$|\Delta^{(\nu)}| \leq M_4 (c^t x^{(\nu)} - c^\infty)^2. \tag{4.43}$$

Due to Lemmas 4.1 and 4.2, we have

$$1 - M_6 (c^t x^{(\nu)} - c^\infty)^2 \leq \frac{\dfrac{\beta_E^{(\nu)t} \widehat{P}_{EE}^{(\nu)} \beta_E^{(\nu)}}{\sigma(\widehat{P}_{EE}^{(\nu)} \beta_E^{(\nu)})}}{\dfrac{\beta^{(\nu)t} P^{(\nu)} \beta^{(\nu)}}{\sigma(P^{(\nu)}\beta^{(\nu)})}} \leq 1 + M_6 (c^t x^{(\nu)} - c^\infty)^2 \tag{4.44}$$

for sufficiently large ν, where $M_6 > 0$ is a constant. By multiplying this inequality by $\beta^{(\nu)t} P^{(\nu)} \beta / \sigma(P^{(\nu)} \beta^{(\nu)})$, we have

$$|\Delta^{(\nu)}| \leq \frac{\beta^{(\nu)t} P^{(\nu)} \beta^{(\nu)}}{\sigma(P^{(\nu)}\beta^{(\nu)})} M_6 (c^t x^{(\nu)} - c^\infty)^2. \tag{4.45}$$

From Lemma 4.3, we have

$$\frac{c^t x^{(\nu+1)} - c^\infty}{c^t x^{(\nu)} - c^\infty} = 1 - \lambda^{(\nu)} \frac{\beta^{(\nu)t} P^{(\nu)} \beta^{(\nu)}}{\sigma(P^{(\nu)}\beta^{(\nu)})} \geq 0, \tag{4.46}$$

hence we obtain

$$\frac{\beta^{(\nu)t} P^{(\nu)} \beta^{(\nu)}}{\sigma(P^{(\nu)}\beta^{(\nu)})} \leq \frac{1}{\lambda^{(\nu)}} \leq \frac{1}{\lambda_{\min}}. \tag{4.47}$$

The relation (4.43) follows from (4.45) and (4.47) immediately, completing the proof of (4.38).

The relation (4.39) follows in a similar manner, by taking note of (4.32) and the fact that

(4.48)

$$\begin{aligned}
&\left\| \frac{\widehat{P}_{EE}^{(\nu)} \beta_E^{(\nu)}}{\sigma(\widehat{P}_{EE}^{(\nu)} \beta_E^{(\nu)})} - \frac{(P^{(\nu)}\beta^{(\nu)})_E}{\sigma((P^{(\nu)}\beta^{(\nu)})_E)} \right\| \\
&= \left\| \left(\frac{1}{\sigma(\widehat{P}_{EE}^{(\nu)} \beta_E^{(\nu)})} - \frac{1}{\sigma((P^{(\nu)}\beta^{(\nu)})_E)} \right) \widehat{P}_{EE}^{(\nu)} \beta_E^{(\nu)} + \frac{Q_{EE}^{(\nu)} \widehat{P}_{EE}^{(\nu)} \beta_E^{(\nu)}}{\sigma((P^{(\nu)}\beta^{(\nu)})_E)} \right\| \qquad \text{(use (4.13).)} \\
&\leq \left\| \frac{\sigma((P^{(\nu)}\beta^{(\nu)})_E) - \sigma(\widehat{P}_{EE}^{(\nu)} \beta_E^{(\nu)})}{\sigma(\widehat{P}_{EE}^{(\nu)} \beta_E^{(\nu)}) \sigma((P^{(\nu)}\beta^{(\nu)})_E)} \widehat{P}_{EE}^{(\nu)} \beta_E^{(\nu)} \right\| + \left\| \frac{Q_{EE}^{(\nu)} \widehat{P}_{EE}^{(\nu)} \beta_E^{(\nu)}}{\sigma((P^{(\nu)}\beta^{(\nu)})_E)} \right\| \\
&\leq \left\| \frac{\sigma((P^{(\nu)}\beta^{(\nu)})_E) - \sigma(\widehat{P}_{EE}^{(\nu)} \beta_E^{(\nu)})}{(|E|/2) \sigma(\widehat{P}_{EE}^{(\nu)} \beta_E^{(\nu)})} \right\| \eta^{-1} \|y\| + \frac{\|Q_{EE}^{(\nu)}\| \eta^{-1} \|y\|}{|E|/2} \\
&\qquad \text{(Use (4.11) and } \sigma((P^{(\nu)}\beta^{(\nu)})_E) \geq |E|/2 \text{ where the latter follows} \\
&\qquad \text{from (4.23) and (4.25).)} \\
&\leq M_8 (c^t x^{(\nu)} - c^\infty)^2 \qquad \text{(use Lemmas 4.2 and 2.1 and (4.9).)}
\end{aligned}$$

where M_7 and M_8 are positive constants. □

LEMMA 4.5. *If $\{\lambda^{(\nu)}\}$ has an upper bound $\lambda_{\max} < 1$ and ν_0 is sufficiently large, the value of $f_{\mathcal{X}}^{(\nu)}$ is bounded below and above as follows for all $\nu \geq \nu_0$:*

$$
\begin{aligned}
& f_{\mathcal{X}}^{(\nu_0)} + \sum_{\tau=\nu_0}^{\nu-1} H_k\left(\widehat{P}_{EE}^{(\tau)}\beta_E^{(\tau)}, \frac{\lambda^{(\tau)}}{\sigma(\widehat{P}_{EE}^{(\tau)}\beta_E^{(\tau)})}\right) - M_9 \\
& \leq f_{\mathcal{X}}^{(\nu)} \\
& \leq f_{\mathcal{X}}^{(\nu_0)} + \sum_{\tau=\nu_0}^{\nu-1} H_k\left(\widehat{P}_{EE}^{(\tau)}\beta_E^{(\tau)}, \frac{\lambda^{(\tau)}}{\sigma(\widehat{P}_{EE}^{(\tau)}\beta_E^{(\tau)})}\right) + M_9 \\
& \leq f_{\mathcal{X}}^{(\nu_0)} + \sum_{\tau=\nu_0}^{\nu-1} T_k\left(\widehat{P}_{EE}^{(\tau)}\beta_E^{(\tau)}, \frac{\lambda^{(\tau)}}{\sigma(\widehat{P}_{EE}^{(\tau)}\beta_E^{(\tau)})}\right) + M_9,
\end{aligned} \tag{4.49}
$$

where M_9 is a positive constant that does not depend on ν.

Proof. From Lemmas 4.3 and 4.4, we see that $H_k(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)}, \lambda^{(\nu)}/\sigma(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)}))$ is well-defined for sufficiently large ν. Then, by using the relations obtained by taking logarithm of (4.38) and (4.39) to evaluate $f_{\mathcal{X}}$, we have the following estimate on (4.36):

(4.50)

$$
\begin{aligned}
& H_k\left(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)}, \frac{\lambda^{(\nu)}}{\sigma(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)})}\right) - M_{10}(c^t x^{(\nu)} - c^\infty)^2 \\
& \leq |E(\mathcal{X})| \log\left(1 - \lambda^{(\nu)}\beta^{(\nu)t}\frac{P^{(\nu)}\beta^{(\nu)}}{\sigma(P^{(\nu)}\beta^{(\nu)})}\right) - \sum_{i\in E(\mathcal{X})} \log\left(1 - \lambda^{(\nu)}\frac{(P^{(\nu)}\beta^{(\nu)})_i}{\sigma(P^{(\nu)}\beta^{(\nu)})}\right) \\
& = f_{\mathcal{X}}^{(\nu+1)} - f_{\mathcal{X}}^{(\nu)} \\
& \leq H_k\left(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)}, \frac{\lambda^{(\nu)}}{\sigma(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)})}\right) + M_{10}(c^t x^{(\nu)} - c^\infty)^2,
\end{aligned}
$$

where $M_{10} > 0$ is an appropriate constant.

By using the inequality above, the value of the local Karmarkar potential function at the νth iteration is bounded below and above as follows for sufficiently large ν:

$$
\begin{aligned}
& f_{\mathcal{X}}^{(\nu_0)} + \sum_{\tau=\nu_0}^{\nu-1} H_k\left(\widehat{P}_{EE}^{(\tau)}\beta_E^{(\tau)}, \frac{\lambda^{(\tau)}}{\sigma(\widehat{P}_{EE}^{(\tau)}\beta_E^{(\tau)})}\right) - M_9 \\
& \leq f_{\mathcal{X}}^{(\nu_0)} + \sum_{\tau=\nu_0}^{\nu-1} H_k\left(\widehat{P}_{EE}^{(\tau)}\beta_E^{(\tau)}, \frac{\lambda^{(\tau)}}{\sigma(\widehat{P}_{EE}^{(\tau)}\beta_E^{(\tau)})}\right) - \sum_{\tau=\nu_0}^{\nu-1} M_{10}(c^t x^{(\tau)} - c^\infty)^2 \\
& \leq f_{\mathcal{X}}^{(\nu_0)} + \sum_{\tau=\nu_0}^{\nu-1} (f_{\mathcal{X}}^{(\tau+1)} - f_{\mathcal{X}}^{(\tau)}) \\
& = f_{\mathcal{X}}^{(\nu)} \\
& \leq f_{\mathcal{X}}^{(\nu_0)} + \sum_{\tau=\nu_0}^{\nu-1} H_k\left(\widehat{P}_{EE}^{(\tau)}\beta_E^{(\tau)}, \frac{\lambda^{(\tau)}}{\sigma(\widehat{P}_{EE}^{(\tau)}\beta_E^{(\tau)})}\right) + \sum_{\tau=\nu_0}^{\nu-1} M_{10}(c^t x^{(\tau)} - c^\infty)^2 \\
& \leq f_{\mathcal{X}}^{(\nu_0)} + \sum_{\tau=\nu_0}^{\nu-1} H_k\left(\widehat{P}_{EE}^{(\tau)}\beta_E^{(\tau)}, \frac{\lambda^{(\tau)}}{\sigma(\widehat{P}_{EE}^{(\tau)}\beta_E^{(\tau)})}\right) + M_9,
\end{aligned} \tag{4.51}
$$

where ν_0 is a number such that the inequality (4.50) holds for all $\nu \geq \nu_0$, and M_9 is the positive constant such that

$$\sum_{\tau=0}^{\infty} M_{10}(c^t x^{(\tau)} - c^\infty)^2 \leq M_9, \tag{4.52}$$

the existence of which follows from the linear convergence of $\{c^t x^{(\nu)} - c^\infty\}$ shown in Lemma 1.2.

It remains to show the last inequality in (4.49). By using Lemmas 4.3 and 4.4, we see that

$$1 - \frac{\lambda^{(\nu)}}{\sigma(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)})}\|\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)}\|^2 > 0 \tag{4.53}$$

for sufficiently large ν. Together with (4.14), we see that

$$(\widehat{P}_{EE}^{(\tau)}\beta_E^{(\tau)}, \lambda^{(\tau)}/\sigma(\widehat{P}_{EE}^{(\tau)}\beta_E^{(\tau)})) \in \Omega_0 \qquad \text{(cf. (3.3))}, \tag{4.54}$$

then we can apply the inequality (3.4) of Lemma 3.1, to obtain the last inequality by replacing H_k by T_k. □

4.3. Main results. Now, we restrict ourselves to the case with the step-size satisfying (1.5) and prove the main results. The results obtained in §4.2 are available in this analysis because (1.5) is a stronger condition than the conditions on $\lambda^{(\nu)}$ adopted in §4.2.

LEMMA 4.6. *If* (1.5) *is satisfied throughout the iteration, we have*

$$\lambda_{\min}^2 \leq \frac{\lambda_{\min}}{\sigma(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)})} \leq \frac{\lambda^{(\nu)}}{\sigma(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)})} \leq \frac{2}{3}|E(\mathcal{X})| \tag{4.55}$$

for sufficiently large ν.

Proof. The rightmost inequality is obvious by using $\lambda^{(\nu)} \leq 2/3$, $\mathbf{1}^t\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)} = 1$. We show the leftmost inequality. From Lemmas 4.1 and 4.2, we see

$$\frac{\|P^{(\nu)}\beta^{(\nu)}\|^2/\sigma(P^{(\nu)}\beta^{(\nu)})}{\|\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)}\|^2/\sigma(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)})} \to 1 \tag{4.56}$$

as $\nu \to \infty$. Then it follows from Lemma 4.3 that

$$1 - \lambda^{(\nu)}\frac{{\beta_E^{(\nu)}}^t\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)}}{\sigma(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)})} > 0 \tag{4.57}$$

asymptotically. We replace $\lambda^{(\nu)}$ by $\lambda_{\min}$, ${\beta_E^{(\nu)}}^t\widehat{P}_{EE}^{(\nu)}\beta_E (= \|\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)}\|^2)$ by $\sigma(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)})^2$, to obtain

$$\lambda_{\min}^2 \leq \frac{\lambda_{\min}}{\sigma(\widehat{P}_{EE}\beta_E^{(\nu)})}. \tag{4.58}$$

□

LEMMA 4.7. *If* (1.5) *is satisfied throughout the iteration, we have*

$$(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)}, \lambda^{(\nu)}/\sigma(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)})) \in \Omega_1$$

(cf. *Lemma* 3.1) *for sufficiently large* ν, *yielding*

$$H_k\left(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)}, \frac{\lambda^{(\nu)}}{\sigma(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)})}\right) \le T_k\left(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)}, \frac{\lambda^{(\nu)}}{\sigma(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)})}\right) \le 0. \tag{4.59}$$

Furthermore, if $\mathcal{X}$ *is not the optimal face* $\mathcal{S}$ *of* $< \mathrm{D} >$, *then we have*

$$H_k\left(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)}, \frac{\lambda^{(\nu)}}{\sigma(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)})}\right) \le T_k\left(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)}, \frac{\lambda^{(\nu)}}{\sigma(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)})}\right) \le -\frac{\lambda_{\min}^2}{|E|(|E|-1)}. \tag{4.60}$$

(N. B. $|E(\mathcal{X})| \ge 2$ *is always satisfied under the assumptions of the lemma as explained in the proof.*)

Proof. As shown in the proof of Lemma 4.5, $(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)}, \lambda^{(\nu)}/\sigma(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)})) \in \Omega_0$ holds for sufficiently large ν. Now we have the condition $\lambda^{(\nu)} \le 2/3$, hence

$$\left(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)}, \frac{\lambda^{(\nu)}}{\sigma(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)})}\right) \in \Omega_1, \tag{4.61}$$

which, together with Lemma 3.1, proves the first part of the lemma.

To see the latter part (4.60), we show that

$$\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)} \not> 0 \tag{4.62}$$

if $\mathcal{X}$ is not the optimal face $\mathcal{S}$. Once (4.62) can be shown, it follows, by using $\mathbf{1}^t\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)} = 1$, that

$$\left\|\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)} - \frac{\mathbf{1}}{k}\right\|^2 \ge \frac{1}{k(k-1)}, \qquad \sigma(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)}) \ge \frac{1}{k-1}, \tag{4.63}$$

which, together with Lemmas 3.1 and 4.6, implies (4.60). Here we note that k is guaranteed to be greater than 1, because a dual degenerate face $\mathcal{Y}$ can be $|E(\mathcal{Y})| = 1$ only if $\mathcal{Y} = \mathcal{S}$ or $\mathcal{Y}$ is the optimal face for the problem $\{\max\ c^t x |\ x \in \mathcal{P}\}$, both of which are excluded due to the assumptions.

Assume, by contradiction, $\mathcal{X} \ne \mathcal{S}$ and $\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)} > 0$. Since $y^t\xi^{(\nu)} = c^t x^{(\nu)} - c^\infty > 0$, we have

$$(y^t\xi^{(\nu)})\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)} = [\xi_E^{(\nu)}]^{-1}\widehat{P}_{EE}^{(\nu)}[\xi_E^{(\nu)}]y_E > 0. \tag{4.64}$$

Due to Lemma 3.2, putting

$$(\hat{y}_E, \hat{y}_{E^c}) = ([\xi_E{}^{(\nu)}]^{-1}\widehat{P}_{EE}^{(\nu)}[\xi_E{}^{(\nu)}]y_E, 0) \tag{4.65}$$

we see $A\hat{y} = c$. Then the pair of an interior point of $\mathcal{X}$ and $\hat{y}$ satisfies a strictly complementarity condition, yielding that $\mathcal{X}$ is the optimal face $\mathcal{S}$ of $\langle \mathrm{D} \rangle$. But this contradicts the assumption $\mathcal{X} \ne \mathcal{S}$, thus, $\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)} \not> 0$ if $\mathcal{X}$ is not the optimal face. This completes the proof. □

Now, we are ready to see the main results.

THEOREM 4.8. *Let* $\{x^{(\nu)}\}$ *be the sequence generated by the iteration* (1.3) *of the affine scaling algorithm with the step-size satisfying* (1.5). *If* $\{c^t x^{(\nu)}\}$ *has the limiting*

value $c^\infty > -\infty$, then $\{x^{(\nu)}\}$ converges to the interior point x^* of the optimal face $\mathcal{S}$ with $\|x^{(\nu)} - x^*\| = O(c^t x^{(\nu)} - c^\infty)$.

Proof. We already observed that the sequence converges to the interior point x^* of the dual degenerate face $\mathcal{X}$ with $\|\xi^{(\nu)}_{E(\mathcal{X})}\| = O(c^t x^{(\nu)} - c^\infty)$ and $\|x^{(\nu)} - x^*\| = O(c^t x^{(\nu)} - c^\infty)$ (cf. Lemma 1.2). Then it is remaining to show that $\mathcal{X}$ is the optimal face $\mathcal{S}$ of $\langle \mathrm{D} \rangle$.

As was mentioned in (4.5), (1) of Lemma 1.2 and the well-known inequality between the arithmetic mean and geometric mean imply

$$\begin{aligned} \exp(f_{\mathcal{X}}(x^{(\nu)})) &= \frac{(c^t x^{(\nu)} - c^\infty)^{|E(\mathcal{X})|}}{\prod_{i \in E(\mathcal{X})} \xi_i} \geq \left(|E(\mathcal{X})| \frac{(c^t x^{(\nu)} - c^\infty)}{\|\xi^{(\nu)}_{E(\mathcal{X})}\|_1} \right)^{|E(\mathcal{X})|} \\ &\geq (\sqrt{|E(\mathcal{X})|}\eta)^{|E(\mathcal{X})|}. \end{aligned} \tag{4.66}$$

On the other hand, due to Lemma 4.7, if $\mathcal{X} \neq \mathcal{S}$, we have

$$H_k \left(\widehat{P}^{(\nu)}_{EE} \beta^{(\nu)}_E, \frac{\lambda^{(\nu)}}{\sigma(\widehat{P}^{(\nu)}_{EE} \beta^{(\nu)}_E)} \right) < -\delta'' \tag{4.67}$$

holds for sufficiently large ν, where δ'' is a positive constant. Together with Lemma 4.5, this implies

$$f^{(\nu)}_{\mathcal{X}} \to -\infty \tag{4.68}$$

as $\nu \to \infty$, and hence

$$\exp(f^{(\nu)}_{\mathcal{X}}) \to 0. \tag{4.69}$$

Comparing (4.66) and (4.69), we see that $\mathcal{X} = \mathcal{S}$ must hold to be consistent, completing the proof. □

THEOREM 4.9. *Under the assumptions of Theorem* 4.8, *the sequence of the dual estimate*

$$\tilde{y}^{(\nu)} = [\xi(x^{(\nu)})]^{-1} A^t G(x^{(\nu)})^{-1} c \tag{4.70}$$

converges to the analytic center of the optimal face of the dual problem of $\langle \mathrm{D} \rangle$.

Proof. Due to the previous theorem, we know now that the sequence $\{x^{(\nu)}\}$ converges to an interior point of the optimal face $\mathcal{S}$ with $\|\xi^{(\nu)}_{E(\mathcal{S})}\| = O(c^t x^{(\nu)} - c^\infty)$. Without loss of generality, we may assume that y defined in (4.2) satisfies

$$(y_{E(\mathcal{X})} =) y_{E(\mathcal{S})} > 0. \tag{4.71}$$

The existence of such y is guaranteed by the strictly complementarity condition. Note that $\xi^t y = c^t x - c^\infty$.

The dual estimate is written as

$$\begin{aligned} \tilde{y}^{(\nu)} &= [\xi^{(\nu)}]^{-2} A^t G^{-1} c = \left[\frac{\xi^{(\nu)}}{y^t \xi^{(\nu)}} \right]^{-1} P^{(\nu)} \beta^{(\nu)} \\ &= \left[\frac{\xi^{(\nu)}}{y^t \xi^{(\nu)}} \right]^{-1} \begin{pmatrix} (P^{(\nu)} \beta^{(\nu)})_{E(\mathcal{X})} \\ (P^{(\nu)} \beta^{(\nu)})_{E^c(\mathcal{X})} \end{pmatrix} \\ &= \left[\frac{\xi^{(\nu)}}{y^t \xi^{(\nu)}} \right]^{-1} \begin{pmatrix} \widehat{P}^{(\nu)}_{EE} (I - Q^{(\nu)}_{EE}) \widehat{P}^{(\nu)}_{EE} \\ Q^{(\nu)}_{E^c E} \end{pmatrix} \beta^{(\nu)}_E. \end{aligned} \tag{4.72}$$

Now we prove the theorem in two steps.

Step 1.

$$(P^{(\nu)}\beta^{(\nu)})_{E(\mathcal{S})} \to \frac{\mathbf{1}}{|E(\mathcal{S})|} \tag{4.73}$$

as $\nu \to \infty$.

Proof of Step 1. We will show that $\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)}$ converges to $\mathbf{1}/|E(\mathcal{S})|$. Due to (4.24), this implies convergence of $(P^{(\nu)}\beta^{(\nu)})_{E(\mathcal{S})}$ to $\mathbf{1}/|E(\mathcal{S})|$. To show this, we assume that $\widehat{P}_{EE}^{(\nu)}\beta_E{}^{(\nu)}$ does not converge to $\mathbf{1}/|E(\mathcal{S})|$ and derive a contradiction. In the case, for an $\varepsilon' > 0$, we can take a subsequence $\{x^{(\nu_\tau)}\}$ of $\{x^{(\nu)}\}$ such that

$$\left\| \widehat{P}_{EE}^{(\nu_\tau)}\beta_E^{(\nu_\tau)} - \frac{\mathbf{1}}{|E(\mathcal{S})|} \right\| \geq \varepsilon'. \tag{4.74}$$

Since $(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)}, \lambda^{(\nu)}/\sigma(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)})) \in \Omega_1$ for sufficiently large ν, we see, by using Lemma 3.1, that

$$H_k\left(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)}, \frac{\lambda^{(\nu)}}{\sigma(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)})}\right) \leq 0 \tag{4.75}$$

for sufficiently large ν, and from (4.74) and Lemma 3.1,

$$H_k\left(\widehat{P}_{EE}^{(\nu_\tau)}\beta_E^{(\nu_\tau)}, \frac{\lambda^{(\nu_\tau)}}{\sigma(\widehat{P}_{EE}^{(\nu_\tau)}\beta_E^{(\nu_\tau)})}\right) \leq -\delta'' \tag{4.76}$$

for sufficiently large τ, where δ'' is a positive constant. Then Lemma 4.5 yields that $f_{\mathcal{S}}^{(\nu)} \to -\infty$ as $\nu \to \infty$. However, this contradicts the fact (4.66) that the local Karmarkar potential function $f_{\mathcal{S}}$ is bounded below. Thus $\widehat{P}_{EE}^{(\nu)}\beta_{E(\mathcal{S})}^{(\nu)}$ converges to $\mathbf{1}/|E(\mathcal{S})|$, completing the proof of Step 1.

Step 2. $\tilde{y}^{(\nu)}$ converges to the analytic center of the optimal face of the dual problem.

Proof of Step 2. Due to Lemma 2.1, (4.9), (4.72), and $\lim_{\nu\to\infty} \xi_{E^c}^{(\nu)} > 0$, we see that $\tilde{y}_{E^c(\mathcal{S})}^{(\nu)}$ converges to zero with an order of $(c^t x^{(\nu)} - c^\infty)^2$. We analyze the behavior of $\tilde{y}_{E(\mathcal{S})}^{(\nu)}$. To begin with, we show that $\{\tilde{y}_{E(\mathcal{S})}^{(\nu)}\}$ is bounded. Let

$$\tilde{\xi}_E^{(\nu)} = \frac{\xi_E^{(\nu)}}{y^t \xi^{(\nu)}}. \tag{4.77}$$

Because of

$$\tilde{\xi}_E^{(\nu)t} y_E = 1 \quad \text{and} \quad y_E > 0, \tag{4.78}$$

$\{\tilde{\xi}_E^{(\nu)}\}$ is bounded. Since

$$f_{\mathcal{S}}^{(\nu)} = -\sum_{i\in E(\mathcal{S})} \log \tilde{\xi}_i^{(\nu)} \tag{4.79}$$

and $f_{\mathcal{S}}^{(\nu)}$ is bounded above by a constant due to Lemmas 4.5 and 4.7, we have

$$\tilde{\xi}_E^{(\nu)} \geq \tilde{\delta}\mathbf{1}, \tag{4.80}$$

where $\tilde{\delta}$ is a positive constant. Then, from (4.72) and (4.73), we see

$$\begin{aligned} \lim_{\nu\to\infty} \left\| \tilde{y}_E^{(\nu)} - [\tilde{\xi}_E^{(\nu)}]^{-1}\frac{\mathbf{1}}{|E|} \right\| &= \lim_{\nu\to\infty} \|[\tilde{\xi}_E^{(\nu)}]^{-1}((P\beta^{(\nu)})_E - \frac{1}{|E|})\| \\ &\leq \lim_{\nu\to\infty} \|[\tilde{\xi}_E^{(\nu)}]^{-1}\| \left\| (P\beta^{(\nu)})_E - \frac{1}{|E|} \right\| = 0. \end{aligned} \tag{4.81}$$

This implies that $\tilde{y}_E^{(\nu)}$ is bounded.

Choose an accumulation point $\tilde{y}_E^*$ of the sequence $\{\tilde{y}_E^{(\nu)}\}$, and let $\{\tilde{y}_E^{(\nu_\tau)}\}$ be a subsequence convergent to $\tilde{y}_E^*$. Because of (4.81), we see $\{\tilde{\xi}_E^{(\nu_\tau)}\}$ also is convergent. Denote by $\tilde{\xi}_E^*$ the limit. Then we obtain

$$\tilde{y}_E^* = \left(\frac{[\tilde{\xi}_E^*]^{-1}\mathbf{1}}{|E(\mathcal{S})|}, 0 \right). \tag{4.82}$$

Now, we check that

$$\lim_{\tau\to\infty} \tilde{y}^{(\nu_\tau)} = \lim_{\tau\to\infty} (\tilde{y}_E^{(\nu_\tau)}, \tilde{y}_{E^c}^{(\nu_\tau)}) = (\tilde{y}_E^*, 0) \tag{4.83}$$

is the unique solution of (3.22). Since

$$\begin{aligned} \tilde{y}^* &= \lim_{\tau\to\infty} [\xi^{(\nu_\tau)}]^{-1} P^{(\nu_\tau)} [\xi^{(\nu_\tau)}] y = (\lim_{\tau\to\infty} [\tilde{\xi}_E^{(\nu_\tau)}]^{-1} \widehat{P}_{EE}^{(\nu_\tau)} \beta_E^{(\nu_\tau)}, 0) \\ &= (\lim_{\tau\to\infty} [\xi_E^{(\nu_\tau)}]^{-1} \widehat{P}_{EE}^{(\nu_\tau)} [\xi_E^{(\nu_\tau)}] y_E, 0) \end{aligned} \tag{4.84}$$

and, due to Lemma 3.2,

$$A_E [\xi_E^{(\nu_\tau)}]^{-1} \widehat{P}_{EE}^{(\nu_\tau)} [\xi_E^{(\nu_\tau)}] y_E = c \tag{4.85}$$

holds for all τ, the limit $\tilde{y}_E^*$ also satisfies $A_E \tilde{y}_E^* = c$. It is also easy to see that the first equation of (3.22) is satisfied, because

$$[\tilde{y}_E^*]^{-1}\mathbf{1} = |E(\mathcal{S})|\tilde{\xi}_E^* \in \mathrm{Im}(A_E^t). \tag{4.86}$$

Due to Lemma 3.3, these relations show that $\tilde{y}^*$ is the analytic center of the dual optimal face. Since this holds for any accumulation point of the sequence of the dual estimates, the dual estimates converge to the analytic center of the dual optimal face, completing the proof. □

THEOREM 4.10. *Under the assumptions of Theorem* 4.8, *the asymptotic reduction rate of the value* $c^t x^{(\nu)} - c^\infty$ *of the objective function is* $1 - \lambda^{(\nu)}$.

Proof. On the way to proving Theorem 4.9, we have shown that $\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)} \to \mathbf{1}/|E|$. $1-\lambda^{(\nu)}$ is bounded below by $1/3$. From these two facts, we see the lemma immediately, by using Lemmas 4.1 and 4.2, as follows:

$$\begin{aligned} \lim_{\nu\to\infty} \frac{c^t x^{(\nu+1)} - c^\infty}{c^t x^{(\nu)} - c^\infty} &= \lim_{\nu\to\infty} \frac{1 - \dfrac{\lambda^{(\nu)}}{\sigma(P^{(\nu)}\beta^{(\nu)})} \beta^{(\nu)t} P^{(\nu)} \beta^{(\nu)}}{1 - \lambda^{(\nu)}} \\ &= \lim_{\nu\to\infty} \frac{1 - \dfrac{\lambda^{(\nu)}}{\sigma(\widehat{P}_{EE}^{(\nu)}\beta_E^{(\nu)})} \beta_E^{(\nu)t} \widehat{P}_{EE}^{(\nu)} \beta_E^{(\nu)}}{1 - \lambda^{(\nu)}} \\ &= 1. \quad \square \end{aligned} \tag{4.87}$$

Now Theorem 1.1 easily follows by joining the contents of Theorems 4.8–4.10.

5. A small example on the convergence of the dual estimates. In this section we show that the bound $2/3$ on $\lambda^{(\nu)}$ is tight to obtain the results of Theorem 1.1 as long as we take a fixed ratio λ at every iteration. (We refer to this step-size as "fixed ratio step-size.") Specifically, we give a two-dimensional example where no fixed ratio step-size choice with $\lambda^{(\nu)} = \lambda > 2/3$ can ensure convergence of the dual estimate to the analytic center of the dual optimal face. On this point we cannot improve Theorem 1.1 any more.

Let us consider the following LP problem:

$$\text{(5.1)} \qquad \min \quad (1 \;\; 1)\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad \text{subject to} \quad \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \geq \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

The optimal solution of this problem is $(0,0)$. The dual problem is the feasibility problem of

$$\text{(5.2)} \qquad \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \qquad y \geq 0.$$

The central trajectory of (5.1) is given as the line $x_1 = x_2$, $x_1 \geq 0$. We will show that the affine scaling algorithm with any fixed ratio step-size $\lambda > 2/3$ can generate iterates exactly on two different rays symmetric with respect to the central trajectory, if we choose an initial point appropriately.

The iterative formula (1.3) of the affine scaling algorithm turns out to be

$$\text{(5.3)} \qquad \begin{pmatrix} x_1^+ \\ x_2^+ \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - \lambda \begin{pmatrix} x_1 \\ x_2^2/x_1 \end{pmatrix} \quad (x_1 \geq x_2).$$

We do not write down the case $x_1 \leq x_2$ because it is obvious from the symmetry.

Let

$$\text{(5.4)} \qquad r = \frac{x_2}{x_1}.$$

Reflecting the homogeneous property of the problem, the dual estimate is a function of r, which can be written as:

$$\text{(5.5)} \qquad \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} + \rho(r)\begin{pmatrix} -1 \\ -1 \\ 1 \end{pmatrix},$$

where

$$\text{(5.6)} \qquad \rho(r) = \frac{1 + r^2}{1 + r^2 + (1 + r)^2}.$$

Note that $\rho(r)$ is a strictly monotone decreasing function in $[0, 1]$, and $\rho(r) = \rho(1/r)$. Thus the dual estimate is identical on two rays symmetric with respect to the central trajectory. If $r = 1$, then we have $\rho(r) = 1/3$, and with this value, (5.5) gives the analytic center of the feasible region of (5.2).

From (5.3), the iteration for r is written as

$$r^+ = \frac{1-\lambda r}{1-\lambda} r \quad (r \leq 1), \tag{5.7}$$

where $r^+ = x_2^+ / x_1^+$.

Now, suppose we have r such that

$$r^+ = \frac{1}{r}. \tag{5.8}$$

Then due to symmetry, we have

$$r^{++} = r. \tag{5.9}$$

This implies that every iterate is exactly on one of the two rays determined by r and r^+ if we start with a point on the ray determined by r. Hence, we focus attention on the solution of

$$r^+ = \frac{1-\lambda r}{1-\lambda} r = \frac{1}{r} \quad (r \leq 1). \tag{5.10}$$

Solving this equation with respect to λ, we have

$$\lambda = \frac{1+r}{1+r+r^2}. \tag{5.11}$$

The right-hand side function is strictly monotone decreasing between 0 and 1, which takes $\lambda = 1$ at $r = 0$ and $\lambda = 2/3$ at $r = 1$. Hence for each value of $1 > \lambda > 2/3$, there exists a unique solution $0 < r(\lambda) < 1$ of (5.11).

Thus, given $1 > \lambda > 2/3$, if we start from a point on the ray determined by $r(\lambda)$, the generated iterates exist just only on one of the two rays determined by $r(\lambda)$ and its reciprocal which are symmetric with respect to the central trajectory. The sequence has the two directions of approach to the optimal solution. Then, what is going on in the dual estimates?

Due to the properties of the dual estimates observed in the remark following (5.6), the dual estimate on the two rays coincides, but is not the analytic center of the dual optimal solution.

Thus, we showed that given any $\lambda > 2/3$, there is an initial point where the dual estimate cannot converge to the analytic center of the dual optimal face.

6. Concluding remarks. Now the theory allows $\lambda^{(\nu)} = 2/3$ to ensure global convergence of the primal-dual iterates, but one may still feel that there remains a gap to $\lambda^{(\nu)} = 0.9 \sim 0.99$ that is often adopted in efficient implementations. A conventional strategy to fill this gap is to use $\lambda^{(\nu)} = 0.99$, say, as a default step-size and switch to $\lambda^{(\nu)} = 2/3$ only if the reduction of the value of the objective function becomes smaller than a tolerance given in advance. The global convergence of the primal iterates and the dual estimates is also ensured with this procedure as well.

In any case, we have to reduce the step-size to about 2/3 in the final stage of the iterations, but this should not be taken badly when we recall that we can obtain the dual optimal solution at this cost, still ensuring the asymptotic reduction rate of the objective function 1/3. The efficiency of the strategy proposed here deserves further investigation by extensive numerical experiments.

We make some comments on the convergence of the dual estimates. Here we showed by the example that the step-size 2/3 is the largest fixed ratio step-size that ensures "convergence of the dual sequence to the analytic center of the dual optimal face."

The example shows that the direction of approach to the optimal does not converge any more if we adopt a fixed ratio greater than 2/3. On the other hand, we know that accumulation points of the dual estimate are determined by the accumulation points of the direction of approach. Hence it looks likely that "the step-size 2/3 is the longest fixed ratio step-size that ensures convergence of the sequence of the dual estimates to one point, i.e., convergence of the pair of the primal iterates and the dual estimates to one point on the primal-dual optimal face." In fact, this conjecture was shown to be true by Hall and Vanderbei [16], who were inspired by the talk given by Tsuchiya at AT&T [40].

Notes added in revision. (1) This is a revised version of the paper [41] where we proved global convergence of the long-step affine scaling algorithm with $\lambda < 2/3$. In the first revision in September 1992, we extended the major results to the case of $\lambda = 2/3$ (we find that the proof substantially holds also in this case), and point out that this is the largest step-size that ensures convergence of the dual estimates to the analytic center of the dual optimal face, as long as one moves with a fixed ratio towards the boundary. (2) Unfortunately, this paper refers some results from [38] and [39] and is not self-contained. We recommend [29] and [31] for self-contained elucidative papers that duplicate the results of this paper.

Acknowledgments. We wish to thank Prof. I.I. Dikin of Siberian Energy Institute, Irkutsk, Russia, for sending the paper [10] that motivated this development. We are grateful to Professors Kunio Tanabe and Shinji Mizuno of the Institute of Statistical Mathematics for their encouragement and stimulating discussions.

A part of the revision was carried out while Takashi Tsuchiya was staying at the Department of Computational and Applied Mathematics of Rice University, Houston, Texas. He would like to thank Professor J. Dennis and the colleagues there for providing the excellent research environment he enjoyed. Professor Tsuchiya also benefitted from continuing valuable discussions with Professor R. D. C. Monteiro of the Systems and Industrial Engineering Department of the University of Arizona since he was given a chance to visit there.

REFERENCES

[1] I. Adler, M. Resende, G. Veiga, and N. Karmarkar, *An implementation of Karmarkar's algorithm for linear programming*, Math. Programming, 44 (1989), pp. 297–335.

[2] I. Adler, N. Karmarkar, M. Resende, and G. Veiga, *Data structures and programming techniques for the implementation of Karmarkar's algorithm*, ORSA J. Comput., 1 (1989), pp. 84–106.

[3] I. Adler and R. D. C. Monteiro, *Limiting behavior of the affine scaling continuous trajectories for linear programming problems*, Math. Programming, 50 (1990), pp. 29–51.

[4] E. R. Barnes, *A variation on Karmarkar's algorithm for solving linear programming problems*, Math. Programming, 36 (1986), pp. 174–182.

[5] D. A. Bayer and J. C. Lagarias, *The nonlinear geometry of linear programming*, I. *Affine and projective trajectories*, Trans. Amer. Math. Soc., 314 (1989), pp. 499–526.

[6] Y.-C. Cheng, D. J. Houck, J.-M. Liu, M. S. Meketon, L. Slutsman, R. J. Vanderbei, and P. Wang, *The* AT&T KORBX *System*, AT&T Tech. J., 68 (1989), pp. 7–19.

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

[8] ———, *O skhodimosti odnogo iteratsionnogo protsessa*, Upravlyaemye Sistemy, 12 (1974), pp. 54-60. (In Russian.)

[9] ———, *The convergence of dual variables*, Tech. Report, Siberian Energy Institute, Irkutsk, Russia, December, 1991.

[10] ———, *Determining the interior point of a system of linear inequalities*, Cybernetics Sys. Anal., 28 (1992), pp. 54–61.

[11] I. I. DIKIN AND V. I. ZORKAL'TSEV, *Iterativnoe Reshenie Zadach Matematicheskogo Programmirovaniya (Algoritmy Metoda Vnutrennikh Tochek)* , Nauka, Novosibirsk, USSR, 1980. (In Russian.)

[12] D. GAY, *Stopping tests that compute optimal solutions for interior-point linear programming algorithms*, Numerical Analysis Manuscript 89-11, AT&T Bell Laboratories, Murray Hill, NJ, 1989.

[13] C. C. GONZAGA, *Conical projection algorithms for linear programming*, Math. Programming, 43 (1989), pp. 151–173.

[14] ———, *Convergence of the large step primal affine-scaling algorithm for primal non-degenerate linear programs*, Tech. Report, Dept. of Systems Engineering and Computer Sciences, COPPE-Federal University of Rio de Janeiro, Brazil, 1990.

[15] O. GÜLER, D. DEN HERTOG, C. ROOS, T. TERLAKY, AND T. TSUCHIYA, *Degeneracy in interior point methods for linear programming*, Ann. Oper. Res., 47 (1993), pp. 107–138.

[16] L. A. HALL AND R. J. VANDERBEI, *Two-thirds is sharp for affine scaling*, Oper. Res. Lett., 13 (1993), pp. 197–201.

[17] N. KARMARKAR, *A new polynomial-time algorithm for linear programming*, Combinatorica, 4 (1984), pp. 373–395.

[18] N. KARMARKAR AND K. RAMAKRISHNAN, *Further developments in the new polynomial-time algorithm for linear programming*, Talk given at ORSA/TIMS National Meeting, Boston, MA, April, 1985.

[19] M. KOJIMA, S. MIZUNO, AND A. YOSHISE, *A primal-dual interior point algorithm for linear programming*, Progress in Math. Programming, N. Megiddo, ed., Springer-Verlag, New York, 1989, pp. 29–47.

[20] ———, *A polynomial-time algorithm for a class of linear complementarity problem*, Math. Programming, 44 (1989), pp. 1–26.

[21] K. A. MCSHANE, C. L. MONMA, AND D. F. SHANNO, *An implementation of a primal-dual interior point method for linear programming*, ORSA J. Comput., 1 (1989), pp. 70–83.

[22] N. MEGIDDO, *Pathways to the optimal set in linear programming*, Progress in Math. Programming, N. Megiddo, ed., Springer-Verlag, New York, 1989, pp. 131–158.

[23] N. MEGIDDO AND M. SHUB, *Boundary behavior of interior point algorithms for linear programming*, Math. Oper. Res., 14 (1989), pp. 97–146.

[24] S. MEHROTRA, *Implementations of affine scaling methods*, Approximate Solutions of System of linear Equations Using Preconditioned Conjugate Gradient Methods, Tech. Report, Dept. of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, 1989.

[25] S. MEHROTRA AND Y. YE, *Finding an interior point in the optimal face of linear programs*, Math. Programming, 62 (1993), pp. 497–515.

[26] C. L. MONMA AND A. J. MORTON, *Computational experience with a dual affine variant of Karmarkar's method for linear programming*, Oper. Res. Lett., 6 (1987), pp. 261–267.

[27] R. MONTEIRO AND I. ADLER, *Interior path-following primal-dual algorithms, Part* I, *linear programming*, Math. Programming, 44 (1989), pp. 27–41.

[28] 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.

[29] R. MONTEIRO, T. TSUCHIYA, AND Y. WANG, *A simplified global convergence proof of the affine scaling algorithm*, Ann. Oper. Res., 47 (1993), pp. 443–482.

[30] M. RESENDE AND G. VEIGA, *An efficient implementation of a network interior point method*, Manuscript, AT&T Bell Laboratories, Murray Hill, NJ, March, 1992.

[31] R. SAIGAL, *A simple proof of primal affine scaling method.* Tech. Report, Dept. of Industrial and Operations Engineering, University of Michigan, Ann Arbor, March, 1993.

[32] A. SCHRIJVER, *Theory of Linear and Integer Programming.* John Wiley & Sons, Chichester, England, 1986.

[33] L. SINHA, B. FREEDMAN, N. KARMARKAR, A. PUTCHA, AND K. RAMAKRISHNAN, *Overseas network planning*, Proc. Third International Network Planning Sysmposium – Networks '86, IEEE Communications Society, June 1–6, 1986, Tarpon Springs, FL, pp. 121–124.

[34] P. TSENG AND Z.-Q. LUO, *On the convergence of the affine-scaling algorithm*, Math. Programming, 56 (1992), pp. 301–319.

[35] T. TSUCHIYA, *On Yamashita's method and Freund's method for linear programming*, Cooperative Research Report Instit. Statist. Math., 10 (1988), pp. 105–115. (In Japanese.)

[36] ———, *Dual standard form linear programming problems and Karmarkar's canonical form*, Lecture Note Res. Instit. Math. Sci., 676 (1988), pp. 330–336. (In Japanese.)

[37] ———, *A Study on the Global and Local Convergence Properties of the Interior Point Algorithms for Linear Programming*, Ph.D. thesis, Faculty of Engineering, University of Tokyo, Tokyo, Japan, 1991. (In Japanese.)

[38] ———, *Global convergence of the affine scaling method for degenerate linear programming problems*, Math. Programming, 52 (1991), pp. 377–404.

[39] ———, *Global convergence property of the affine scaling method for primal degenerate linear programming problems*, Math. Oper. Res., 17 (1992), pp. 527–557.

[40] ———, *Recent development on the global convergence analysis of long-step affine scaling algorithms*, Talk given at AT&T Bell Laboratories, Murray Hill, NJ, August 13, 1992.

[41] T. TSUCHIYA AND M. MURAMATSU, *Global convergence of a long-step affine scaling algorithm for degenerate linear programming problems*, Research Memo. No. 423, the Institute of Statistical Mathematics, Tokyo, Japan, January, 1992.

[42] T. TSUCHIYA AND K. TANABE, *Local convergence properties of new methods in linear programming*, J. Oper. Res. Soc. Japan, 33 (1990), pp. 22–45.

[43] R. J. VANDERBEI, M. S. MEKETON, AND B. A. FREEDMAN, *A modification of Karmarkar's linear programming algorithm*, Algorithmica, 1 (1986), pp. 395–407.

[44] R. J. VANDERBEI AND J. C. LAGARIAS, *I. I. Dikin's convergence result for the affine-scaling algorithm*, Contemp. Math., 114 (1990), pp. 109–119.

[45] H. YAMASHITA, *A polynomially and quadratically convergent method for linear programming*, Tech. Report, Mathematical System Inc., Tokyo, Japan, 1986.

SIAM J. OPTIMIZATION
Vol. 5, No. 3, pp. 552–569, August 1995

005

ON EIGENVALUE OPTIMIZATION*

ALEXANDER SHAPIRO† AND MICHAEL K. H. FAN‡

Abstract. In this paper we study optimization problems involving eigenvalues of symmetric matrices. One of the difficulties with numerical analysis of such problems is that the eigenvalues, considered as functions of a symmetric matrix, are not differentiable at those points where they coalesce. We present a general framework for a smooth (differentiable) approach to such problems. It is based on the concept of transversality borrowed from differential geometry. In that framework we discuss first- and second-order optimality conditions and rates of convergence of the corresponding second-order algorithms. Finally we present some results on the sensitivity analysis of such problems.

Key words. nonsmooth optimization, transversality condition, first- and second-order optimality conditions, Newton's algorithm, quadratic rate of convergence, semi-infinite programming, sensitivity analysis

AMS subject classifications. 90C30, 90C31, 90C34

1. Introduction.

Optimization problems involving eigenvalues of symmetric matrices arise in many applications (see, e.g., [3], [13], [16], [19], [23] and references therein). One of the main difficulties with numerical analysis of such problems is that the eigenvalues, considered as functions of a symmetric matrix, are not differentiable at those points where they coalesce. This results in problems that are typically nonsmooth (nondifferentiable). In the 1970's and 1980's first-order algorithms for optimization of nonsmooth functions were developed and applied, particularly, to the eigenvalue optimization problems (EOP) (cf. [3], [9], [16], [18]). At the same time various attempts were made to develop a second-order theory for nonsmooth optimization problems. In spite of these attempts such a general second-order theory has not crystallized yet.

An approach to a second-order analysis of the EOP was suggested by Overton [12] and developed further in [13] and [14] (see also [27] and [28] for an application of Overton's method to some particular problems). Recently Fan [6] suggested an alternative quadratically convergent algorithm for solving the EOP.

The goal of this paper is to present a general framework for a second-order analysis of the EOP. In the process we intend to clarify the above methods and to obtain a number of new results. The main idea of Overton's approach can be described as follows. Let $\mathcal{A}(x)$ be a differentiable mapping from $\mathbb{R}^m$ into the linear space $\mathcal{S}_n$ of $n \times n$ symmetric matrices and let $\lambda_1(x) \geq \cdots \geq \lambda_n(x)$ be the eigenvalues of $\mathcal{A}(x)$ considered as functions of $x \in \mathbb{R}^m$. Suppose that we want to minimize the largest eigenvalue $\lambda_1(x)$. Let x^* be a minimizer of $\lambda_1(x)$ over the space $\mathbb{R}^m$. If $\lambda_1(x^*)$ has multiplicity $k > 1$, then $\lambda_1(\cdot)$ is not differentiable at the point x^* and consequently the considered optimization problem is essentially nonsmooth. In order to overcome this difficulty let us restrict the feasible set by introducing the constraints $\lambda_1(x) = \cdots = \lambda_k(x)$. Clearly minimization of $\lambda_1(x)$ subject to these constraints is equivalent to minimization of the function $g(x) = \sum_{i=1}^{k} \lambda_i(x)$ subject to the same constraints. It can be shown that, under certain regularity conditions, the such constructed *constrained* optimization

* Received by the editors March 26, 1993; accepted for publication (in revised form) December 20, 1993.

† School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332.

‡ School of Electrical Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332.

problem is smooth in a neighborhood of the point x^* and hence powerful methods of smooth analysis can be applied.

The organization of this paper is as follows. In the next section we discuss regularity conditions that are required for the constrained optimization problem to be smooth. The development of the section is based on the transversality theory borrowed from differential geometry. In particular we give conditions under which the restricted feasible set $\{x : \lambda_1(x) = \cdots = \lambda_k(x)\}$ is a smooth manifold near the point x^*. In §3 we discuss first- and second-order optimality conditions for the constrained and the original (unconstrained) problems. Section 4 is devoted to a discussion of algorithms of Overton and Fan. We show that typically these algorithms converge quadratically. Finally, in §5 we present some results on sensitivity analysis of the EOP depending on parameters.

The described methods can be applied to a variety of optimization problems involving eigenvalues of symmetric matrices. For example, one can consider the constraint $\lambda_n(x) \geq 0$, which is equivalent to the condition that the matrix $\mathcal{A}(x)$ is nonnegative definite. In that case the corresponding constrained problem should be defined by imposing the additional constraints $\lambda_{n-q+1}(x) = \cdots = \lambda_n(x)$, where q is the multiplicity of the smallest eigenvalue of $\mathcal{A}(x^*)$. In order to simplify presentation and to demonstrate main ideas we shall not attempt to discuss the problem in its most general form. Instead we restrict our attention to the following problem:

$$\min_{x \in \mathbb{R}^m} f(x), \tag{1.1}$$

with $f(x) = \sum_{i=1}^{c} \lambda_i(x)$ and $1 \leq c \leq n$. We refer to (1.1) as the *original* or *unconstrained* problem.

2. Transversality condition. In this section we discuss the transversality condition and its application to the EOP. For a detail study of transversality concept and relevant references we refer to [4].

Let X and Y be two finite dimensional vector spaces, $W \subset Y$ be a smooth manifold, and $f : X \to Y$ be a smooth (differentiable) mapping.

Definition. It is said that f intersects W transversally at a point $x \in X$ (denoted by $f \pitchfork_x W$) if either (i) $f(x) \notin W$ or (ii) $f(x) \in W$ and

$$Y = T_{f(x)}W + (df)_x X. \tag{2.1}$$

If $f \pitchfork_x W$ for all $x \in X$ we say that f intersects W transversally (denoted $f \pitchfork W$).

Here $T_y W$ denotes the tangent space to W at $y \in W$ and $(df)_x : X \to Y$ is the linear mapping corresponding to the differential of f at x, i.e., $(df)_x X$ is the range space of the Jacobian matrix $\nabla f(x)$.

The following properties of transversality will be important for our analysis. It is possible to define transversality in an equivalent form as follows. Suppose that in a neighborhood of a point $\bar{y} = f(\bar{x}) \in W$ the manifold W is defined by equations $W = \{y : g(y) = 0\}$, where $g = (g_1, \ldots, g_k)$ is a smooth mapping from Y into $\mathbb{R}^k$ such that the Jacobian matrix $\nabla g(\bar{y})$ has full rank k. Then $f \pitchfork_{\bar{x}} W$ if and only if the Jacobian matrix $\nabla h(\bar{x})$, of the composite mapping $h = g \circ f : X \to \mathbb{R}^k$, has full rank k. Consider now the set $V = f^{-1}(W)$ which can be also written in the form $V = \{x \in X : g \circ f(x) = 0\}$. By the Implicit Function Theorem it immediately follows from the above characterization of transversality that if $\bar{x} \in V$ and $f \pitchfork_{\bar{x}} W$, then the set V is a smooth manifold in a neighborhood of the point $\bar{x}$.

Transversality is stable under small perturbations. That is, if $f \pitchfork W$, the set W is closed and $g : X \to Y$ is a smooth mapping sufficiently close to f in the C^1 norm, then $g \pitchfork W$. Alternatively we can say that the set of mappings which intersects a closed manifold W transversally, forms an open set in the normed space $C^1(X, Y)$.

Transversality is a generic property in the following sense. Let Π be a finite dimensional vector space and let $F(x, \pi)$ be a $C^\infty(X \times \Pi, Y)$ mapping, i.e., F is an infinitely differentiable mapping from $X \times \Pi$ into Y. We view Π as a space of parameters and for $\pi \in \Pi$ define the mapping $f_\pi(\cdot) = F(\cdot, \pi)$. Suppose that $F \pitchfork W$. Then it can be shown that for almost every $\pi \in \Pi$ the mapping f_π intersects W transversally. That is, those $\pi \in \Pi$ such that f_π does not intersect W transversally form a set of Lebesgue measure zero in Π. Note that, in particular, $F \pitchfork W$ if the Jacobian matrix $\nabla F(x, \pi)$ has full rank equal to $\dim Y$, i.e., $dF(x, \pi) : X \times \Pi \to Y$, is onto, for all (x, π) such that $F(x, \pi) \in W$.

Finally let us note that if $f(x) \in W$ and $f \pitchfork_x W$, then necessarily the following dimensionality condition holds:

$$\dim W + \dim X \geq \dim Y.$$

We apply now the transversality theory to the space $\mathcal{S}_n$ of symmetric matrices and the mapping $\mathcal{A} : \mathbb{R}^m \to \mathcal{S}_n$. Note that the space $\mathcal{S}_n$ has dimension $n(n+1)/2$. By $\lambda_1(A) \geq \cdots \geq \lambda_n(A)$ we denote the eigenvalues of a symmetric matrix $A \in \mathcal{S}_n$. For two integers p and q, $1 \leq p+1 < q \leq n$, consider

$$W(p, q) = \{A \in \mathcal{S}_n : \lambda_p(A) > \lambda_{p+1}(A) = \cdots = \lambda_q(A) > \lambda_{q+1}(A)\}.$$

Recall that a symmetric matrix $A \in \mathcal{S}_n$ can be written in the form (spectral decomposition) $A = E\Lambda E^T$, where Λ is the diagonal matrix $\Lambda = \operatorname{diag}(\lambda_1(A), \ldots, \lambda_n(A))$ and E is an orthogonal matrix formed from the corresponding set of orthonormal eigenvectors. This decomposition can be viewed as the mapping $\Phi : (\Lambda, E) \to E\Lambda E^T$ from the manifold $\mathcal{D}_n \times \mathcal{O}(n)$ into $\mathcal{S}_n$. (Here $\mathcal{D}_n$ and $\mathcal{O}(n)$ denote the sets of diagonal and orthogonal $n \times n$ matrices, respectively.) The set $W(p, q)$ is then the image, under the mapping Φ, of the submanifold of $\mathcal{D}_n \times \mathcal{O}(n)$ obtained by restricting the diagonal entries $\lambda_{p+1} = \cdots = \lambda_q$ of the matrices $\Lambda \in \mathcal{D}_n$. By verifying that the Jacobian matrix of the mapping Φ restricted to this submanifold has a constant rank it is possible to show that $W(p, q)$ is a smooth manifold (cf. [2]). Moreover, the tangent space of $W(p, q)$ can be derived by linearization of the mapping Φ. That is, $d\Phi = (dE)\Lambda E^T + E\Lambda(dE^T) + E(d\Lambda)E^T$ and since $E^T E = I_n$, we have that $(dE^T)E + E^T(dE) = 0$. It follows from these equations and the constraints $\lambda_{p+1} = \cdots = \lambda_q$ that the tangent space to $W = W(p, q)$ at $A \in W(p, q)$ is given by the linear equations

$$T_A W = \{d\Phi \in \mathcal{S}_n : e_i^T(d\Phi)e_j = 0,\ p+1 \leq i < j \leq q;\ e_i^T(d\Phi)e_i = \delta,\ i = p+1, \ldots, q\},$$

where $e_1, \ldots, e_n$ is a set of orthonormal eigenvectors of A corresponding to the eigenvalues $\lambda_1(A), \ldots, \lambda_n(A)$ and δ is the additional parameter representing the common value of $e_i^T(d\Phi)e_i$, $i = p+1, \ldots, q$.

These arguments can be extended to the following more general situation. Consider a sequence of integers $p_1, q_1, \ldots, p_k, q_k$ such that $1 \leq p_1 + 1 < q_1 < p_2 + 1 < q_2 < \cdots < p_k + 1 < q_k \leq n$ and the manifold

$$W(p_1, q_1, \ldots, p_k, q_k) = W(p_1, q_1) \cap \ldots \cap W(p_k, q_k).$$

This manifold is formed by symmetric matrices with the corresponding eigenvalues of multiplicities $r_i = q_i - p_i$, $i = 1, \ldots, k$.

PROPOSITION 2.1. *The set $W = W(p_1, q_1, \ldots, p_k, q_k)$ is a smooth manifold of dimension $n(n+1)/2 + k - \sum_{i=1}^{k} r_i(r_i+1)/2$. The tangent space to W at $A \in W$ is given by the linear equations*

$$T_A W = \{X \in \mathcal{S}_n : \begin{array}{l} e_i^T X e_j = 0,\ p_\ell + 1 \le i < j \le q_\ell; \\ e_i^T X e_i = \delta_\ell,\ i = p_\ell + 1, \ldots, q_\ell;\ \ell = 1, \ldots, k\}, \end{array} \tag{2.2}$$

where $e_1, \ldots, e_n$ is a set of orthonormal eigenvectors corresponding to the eigenvalues $\lambda_1(A), \ldots, \lambda_n(A)$ and δ_ℓ is the additional parameter representing the common value of $e_i^T X e_i$, $i = p_\ell + 1, \ldots, q_\ell$.

Denote $e_1(x), \ldots, e_n(x)$ a set of orthonormal eigenvectors of $\mathcal{A}(x)$ corresponding to the eigenvalues $\lambda_1(x), \ldots, \lambda_n(x)$. With the mapping $\mathcal{A} : \mathbb{R}^m \to \mathcal{S}_n$ are associated symmetric matrices $A_s(x)$ given by the partial derivatives $A_s(x) = \partial \mathcal{A}(x)/\partial x_s$, $s = 1, \ldots, m$, and m-dimensional vectors

$$v_{ij}(x) = (e_i(x)^T A_1(x)\, e_j(x), \ldots, e_i(x)^T A_m(x)\, e_j(x))^T,\ i, j = 1, \ldots, n. \tag{2.3}$$

Transversality conditions for the mapping $\mathcal{A}$ with respect to the smooth manifold $W = W(p_1, q_1, \ldots, p_k, q_k)$ are given now in the following theorem.

THEOREM 2.2. *Suppose that $\mathcal{A}(x) \in W$. Then $\mathcal{A} \pitchfork_x W$ if and only if vectors $v_{ij}(x)$, $p_\ell + 1 \le i < j \le q_\ell$; $v_{ii}(x) - v_{q_\ell q_\ell}(x)$, $i = p_\ell + 1, \ldots, q_\ell - 1$; $\ell = 1, \ldots, k$ are linearly independent.*

Proof. Consider the inner product $\langle B, C \rangle = tr\, B^T C$, on the space of $n \times n$ square matrices, and let $\mathcal{L}$ be the linear space generated by the matrices $A_1(x), \ldots, A_m(x)$. Note that $\mathcal{L} = (d\mathcal{A})_x \mathbb{R}^m$. Consider the tangent space $T_A W$ at $A = \mathcal{A}(x)$. Recall that this tangent space is defined by the linear equations specified in (2.2) and note that $e_i^T X e_j = 0$ if and only if $\langle X, e_j e_i^T \rangle = 0$. Therefore the orthogonal complement $(T_A W)^\perp$ to the space $T_A W$, with respect to the inner product $\langle \cdot, \cdot \rangle$, is generated by matrices $e_j(x) e_i(x)^T$, $p_\ell + 1 \le i < j \le q_\ell$; $e_i(x) e_i(x)^T - e_{q_\ell}(x) e_{q_\ell}(x)^T$, $i = p_\ell + 1, \ldots, q_\ell - 1$; $\ell = 1, \ldots, k$.

It will be sufficient to show that $\mathcal{L}^\perp \cap (T_A W)^\perp = \{0\}$. That is, if $Y \in (T_A W)^\perp$ and $\langle Y, A_s(x) \rangle = 0$, $s = 1, \ldots, m$, then $Y = 0$. These last conditions can be written as a system of m linear equations with unknowns corresponding to the matrices $e_j(x) e_i(x)^T$, $p_\ell + 1 \le i < j \le q_\ell$; $e_i(x) e_i(x)^T - e_{q_\ell}(x) e_{q_\ell}(x)^T$, $i = p_\ell + 1, \ldots, q_\ell - 1$; $\ell = 1, \ldots, k$. It remains to note that this system only has the zero solution if and only if the linear independence condition, specified in the formulation of the theorem, holds. □

Consider now the set $V = V(p_1, q_1, \ldots, p_k, q_k) = \mathcal{A}^{-1}(W(p_1, q_1, \ldots, p_k, q_k))$. We have then that if $x \in V$ and $\mathcal{A} \pitchfork_x W$, then V is a smooth manifold, of dimension $m + k - \sum_{i=1}^{k} r_i(r_i+1)/2$, in a neighborhood of the point x. The tangent space to this manifold at $x \in V$ is given by the linear equations

$$T_x V = \{y \in \mathbb{R}^m : \begin{array}{l} y^T v_{ij}(x) = 0,\ p_\ell + 1 \le i < j \le q_\ell; \\ y^T v_{ii}(x) = \delta_\ell,\ i = p_\ell + 1, \ldots, q_\ell;\ \ell = 1, \ldots, k\}. \end{array} \tag{2.4}$$

Note that $\mathcal{A}(x) \in W$ and $\mathcal{A} \pitchfork_x W$ imply the dimensionality condition

$$m + k - \sum_{i=1}^{k} r_i(r_i+1)/2 \ge 0. \tag{2.5}$$

As we mentioned earlier transversality is a generic property. Suppose, for example, that $\mathcal{A}(x)$ is an affine mapping, i.e., $\mathcal{A}(x) = A_0 + x_1A_1 + \cdots + x_mA_m$, and consider the (symmetric) matrix A_0 as a parameter vector. That is, define $F(x, A) = A + x_1A_1 + \cdots + x_mA_m$. Clearly $\mathcal{A}(\cdot) = F(\cdot, A_0)$. Moreover, $dF(x, A) : \mathbb{R}^m \times \mathcal{S}_n \to \mathcal{S}_n$ is onto and hence $F \pitchfork W$. Consequently, for almost every A_0 the affine mapping $\mathcal{A}$ intersects W transversally and hence for almost every A_0 the corresponding set V is a smooth manifold. Note that, in particular, this implies that if the dimensionality condition (2.5) does not hold, then for almost every A_0, $\mathcal{A}(x) \notin W(p_1, q_1, \ldots, p_k, q_k)$ for all $x \in \mathbb{R}^m$ and hence the set $V(p_1, q_1, \ldots, p_k, q_k)$ is empty (cf. [20]).

3. Optimality conditions. In this section we discuss first- and second-order optimality conditions for the original problem (1.1) and the associated constrained problem (3.2) formulated below. We assume that the mapping $\mathcal{A}(x)$ is twice continuously differentiable. Let x^* be a minimizer of the function $f(x) = \sum_{i=1}^{c} \lambda_i(x)$ over $\mathbb{R}^m$ and let p and q be two integers such that $1 \le p+1 \le c \le q \le n$ and $x^* \in V(p, q)$, where

$$V(p, q) = \{x : \lambda_p(x) > \lambda_{p+1}(x) = \cdots = \lambda_q(x) > \lambda_{q+1}(x)\}. \tag{3.1}$$

Note that the sum of the eigenvalues $\sum_{i=p+1}^{q} \lambda_i(x)$ is a differentiable function at any $x \in V(p, q)$ (cf. [14], [23]).

Together with (1.1) we associate the following optimization problem, referred to as the *constrained* problem,

$$\min_{x \in V(p,q)} g(x), \tag{3.2}$$

where

$$g(x) = \sum_{i=1}^{p} \lambda_i(x) + \frac{c-p}{q-p} \sum_{i=p+1}^{q} \lambda_i(x).$$

Let us observe that the objective functions f and g coincide on the set $V(p, q)$ and hence the constrained problem (3.2) is obtained by restricting the feasible set of problem (1.1) to $V(p, q)$. It follows that x^* is also an optimal solution of (3.2). Moreover, the function g is differentiable in a neighborhood of the point x^* and $V(p, q)$ is a smooth manifold near x^* provided $\mathcal{A} \pitchfork_{x^*} W(p, q)$. Consequently we obtain that the problem (3.2) is smooth near x^* provided the transversality condition holds.

If $c = q$, then f is differentiable at x^* and standard (first-order) necessary conditions are given by $\nabla f(x^*) = 0$. Note that in this case (cf. [10], [14], [23])

$$\nabla f(x^*) = \sum_{i=1}^{c} v_{ii}(x^*) = (\mathrm{tr}A_1(x^*)Q(x^*), \ldots, \mathrm{tr}A_m(x^*)Q(x^*))^T,$$

where the matrix $Q(x^*) = \sum_{i=1}^{c} e_i(x^*)e_i(x^*)^T$ is independent of a particular choice of the orthonormal eigenvectors $e_1(x^*), \ldots, e_n(x^*)$. Suppose now that $c < q$ and that $\mathcal{A} \pitchfork_{x^*} W(p, q)$, and hence the problem (3.2) is smooth at x^*. By the standard first-order necessary conditions we have then that $\nabla g(x^*)$ is orthogonal to the tangent space $T_{x^*}V(p, q)$. This together with the corresponding formula for the tangent space $T_{x^*}V(p, q)$ (see (2.4)) implies existence of multipliers α_{ij} such that

$$\nabla g(x^*) + \sum_{p+1 \le i < j \le q} \alpha_{ij} v_{ij}(x^*) + \sum_{i=p+1}^{q-1} \alpha_{ii}(v_{ii}(x^*) - v_{qq}(x^*)) = 0. \tag{3.3}$$

Moreover,

$$\nabla g(x^*) = \sum_{i=1}^{p} v_{ii}(x^*) + \frac{c-p}{q-p} \sum_{i=p+1}^{q} v_{ii}(x^*).$$

Putting these two equations together we derive existence of multipliers β_{ij} such that

$$\begin{aligned} &\textstyle\sum_{i=1}^{p} v_{ii}(x^*) + \sum_{i,j=p+1}^{q} \beta_{ij} v_{ij}(x^*) = 0, \\ &\beta_{ij} = \beta_{ji},\ i,j = p+1,\ldots,q, \quad \text{and} \quad \textstyle\sum_{i=p+1}^{q} \beta_{ii} = c-p. \end{aligned} \tag{3.4}$$

Equations (3.4) represent first-order necessary conditions for the constrained problem (3.2). Note that because of the transversality, and hence the linear independence condition specified in Theorem 2.2, the multipliers β_{ij} satisfying (3.4) are unique. Consider the $r \times r$, $r = q-p$, symmetric matrix $B = [\beta_{ij}]$, $i,j = p+1,\ldots,q$, formed by the multipliers β_{ij}. It can be shown by methods of convex analysis (e.g., [14]) that if x^* is a minimizer of f over $\mathbb{R}^m$, then in addition to (3.4) the matrix B must satisfy the conditions $B \geq 0$ and $B \leq I_{q-p}$, i.e., the matrices B and $I_{q-p} - B$ must be nonnegative definite. (Note that if $c-p = 1$, then the condition $B \leq I_{q-p}$ follows from the conditions $B \geq 0$ and $\operatorname{tr} B = 1$.) Moreover, if the mapping $\mathcal{A}$ is affine, then the function f is convex and (3.4) together with nonnegative definiteness of B and $I_{q-p} - B$, are sufficient conditions for the optimality of x^*. It is remarkable that the only difference between the first-order optimality conditions for the problems (1.1) and (3.2) is the additional condition of nonnegative definiteness of the matrices B and $I_{q-p} - B$.

Let us discuss now second-order optimality conditions for the optimization problems (1.1) and (3.2). Consider the $n \times (q-p)$ matrix $E(x) = [e_{p+1}(x),\ldots,e_q(x)]$ and the corresponding orthogonal projection matrix $P(x) = E(x)E(x)^T$ onto the space generated by the eigenvectors $e_{p+1}(x),\ldots,e_q(x)$. Although the individual eigenvectors $e_i(x)$ can be even discontinuous, the projection matrix $P(x)$ is a differentiable function of x in a neighborhood of x^* [10]. Let us consider the following representation of the projection matrix $P(x)$. We construct now an $n \times (q-p)$ matrix $U(x) = [u_{p+1}(x),\ldots,u_q(x)]$ such that: (i) $P(x) = U(x)U(x)^T$, (ii) $U(x)^T U(x) = I_{q-p}$, (iii) $U(x^*) = E(x^*) = E$, (iv) $U(x)$ is a differentiable function of x in a neighborhood of x^*, and (v) $U(x^*)^T dU(x^*) = 0$.

In order to construct $U(x)$ we use the least squares method. That is, consider the following set of $n \times (q-p)$ matrices

$$\mathcal{M}(x) = \{G : P(x)G = G,\ G^T G = I_{q-p}\}. \tag{3.5}$$

Note that the set $\mathcal{M}(x)$ is formed by matrices $G = E(x)H$, where H is a $(q-p)\times(q-p)$ orthogonal matrix. We take $U(x)$ to be a matrix $G \in \mathcal{M}(x)$ which minimizes the squared distance $\operatorname{tr}(E-G)^T(E-G)$ from the matrix $E = E(x^*)$ to $\mathcal{M}(x)$. Since $E \in \mathcal{M}(x^*)$, it follows that $U(x^*) = E$. For a fixed x, the set $\mathcal{M}(x)$ is a smooth manifold. Moreover, since $P(x)$ is differentiable, $\mathcal{M}(x)$ depends on x in a smooth (differentiable) way. That is, $\mathcal{M}(x)$ can be locally defined by a system of differentiable equations which are also differentiable functions of x. Since $E \in \mathcal{M}(x^*)$, it is then a general result that the least squares solution $U(x)$ is a differentiable function of x in a neighborhood of x^*. (This general result can be derived by writing the corresponding first-order optimality conditions and applying the Implicit Function Theorem to the obtained system of equations, see, e.g., [1], [8], [22]. A somewhat similar result is given in Goodman

[5, Lemma 4.1] although the construction there is different and the corresponding smooth functions generate the tangent space to the manifold rather than the manifold itself.) Finally, it is known that $E^TU(x)$ is a symmetric matrix [26]. It follows that $E^TdU(x^*)$ is also a symmetric matrix. Moreover, since $U(x)^TU(x) = I_{q-p}$, we have $U(x)^TdU(x)+dU(x)^TU(x) = 0$ and hence, by symmetry of $U(x^*)^TdU(x^*)$, we obtain $U(x^*)^TdU(x^*) = 0$. Note that $u_{p+1}(x),\dots,u_q(x)$ are not necessarily eigenvectors of $\mathcal{A}(x)$ unless $x \in V(p,q)$.

Now, in a neighborhood of x^*, the manifold $V(p,q)$ can be defined by the equations

$$U(x)^T\mathcal{A}(x)U(x) = \delta I_{q-p}, \tag{3.6}$$

where δ is the additional parameter corresponding to the common value of $\lambda_{p+1}(x)$, $\cdots,\lambda_q(x)$, when $x \in V(p,q)$. Denote by δ^* the common value of the eigenvalues $\lambda_{p+1}(x^*),\dots,\lambda_q(x^*)$ and let $\mathcal{A}(x^*) = \mu_1P_1+\cdots+\mu_hP_h$ be the spectral decomposition of the matrix $\mathcal{A}(x^*)$. That is, $\mu_1 > \cdots > \mu_h$ are the distinct eigenvalues of $\mathcal{A}(x^*)$ and $P_1,\dots,P_h$, are the corresponding orthogonal projection matrices. In particular for some $l \in \{1,\dots,h\}$, $\mu_l = \delta^* = \lambda_{p+1}(x^*) = \cdots = \lambda_q(x^*)$, and $P_l = P(x^*)$. We have that $P(x)$ is differentiable at $x = x^*$ and the corresponding differential can be written in the form [10],

$$dP = \sum_{k\neq l}(\mu_l - \mu_k)^{-1}[P_l(d\mathcal{A})P_k + P_k(d\mathcal{A})P_l] \tag{3.7}$$

(all differentials are calculated at $x = x^*$). Moreover,

$$P(x)u_i(x) = u_i(x), \qquad i = p+1,\dots,q,$$

and hence

$$(dP)u_i + P(du_i) = du_i, \qquad i = p+1,\dots,q.$$

Since $P(x^*)dU(x^*) = 0$ and $U(x^*) = E(x^*)$, we obtain

$$du_i(x^*) = (dP)e_i(x^*), \qquad i = p+1,\dots,q.$$

Together with (3.7) this implies that

$$\begin{aligned} du_i(x^*) &= \textstyle\sum_{k\neq l}(\mu_l - \mu_k)^{-1}P_k(d\mathcal{A})e_i(x^*) \\ &= \textstyle\sum_{k\neq p+1,\dots,q}(\delta^* - \lambda_k(x^*))^{-1}e_k(x^*)e_k(x^*)^T(d\mathcal{A})e_i(x^*), \end{aligned}$$

$i = p+1,\dots,q$, and hence

$$\frac{\partial u_i(x^*)}{\partial x_s} = \sum_{k\neq p+1,\dots,q}(\delta^* - \lambda_k)^{-1}\tau_{ik}^s e_k, \tag{3.8}$$

where $\tau_{ik}^s = e_i^TA_se_k$ and all functions are calculated at $x = x^*$. (Note that, by the symmetry of A_s, $\tau_{ik}^s = \tau_{ki}^s$.)

Consider now (3.6) defining the manifold $V(p,q)$. By differentiating these equations at $x = x^*$ we obtain,

$$U^T(d\mathcal{A})U + U^T\mathcal{A}(dU) + (dU^T)\mathcal{A}U = (d\delta)I_{q-p}, \tag{3.9}$$

where all terms in (3.9) are calculated at $x = x^*$. Since $U(x^*) = E$ and $E^T dU = 0$ we obtain from (3.9)

$$E^T(d\mathcal{A})E = (d\delta)I_{q-p}. \tag{3.10}$$

Note that $d\mathcal{A}(x^*) = A_1(x^*)dx_1 + \cdots + A_m(x^*)dx_m$ and hence (3.10) define the tangent space $T_{x^*}V(p,q)$ in the same way as in (2.4) with the vector y in (2.4) replaced by dx.

Consider now the Lagrangian function

$$L(x,\alpha) = g(x) + \sum_{p+1\le i<j\le q} \alpha_{ij} u_i(x)^T \mathcal{A}(x) u_j(x) + \sum_{i=p+1}^{q-1} \alpha_{ii}[u_i(x)^T \mathcal{A}(x) u_i(x) - u_q(x)^T \mathcal{A}(x) u_q(x)]$$

corresponding to the constrained problem (3.2). It follows from the above discussion that the standard first-order optimality conditions, applied to this Lagrangian, lead to the same Lagrange multipliers $\alpha^* = (\alpha_{ij})$ as in (3.3) and hence to the multipliers β_{ij} given in (3.4). Furthermore, (3.8) allow us to calculate the Hessian matrix of this Lagrangian.

Consider the matrices $A_{st}(x) = \partial^2 \mathcal{A}(x)/\partial x_s \partial x_t$ of second-order partial derivatives and the functions $\phi_{ij}(x) = u_i(x)^T \mathcal{A}(x) u_j(x)$, $p+1 \le i,j \le q$. We have

$$\begin{aligned}\frac{\partial^2 \phi_{ij}(x)}{\partial x_s \partial x_t} &= \frac{\partial^2 u_i(x)^T}{\partial x_s \partial x_t}\mathcal{A}(x)u_j(x) + \frac{\partial u_i(x)^T}{\partial x_s}\mathcal{A}(x)\frac{\partial u_j(x)}{\partial x_t} + \frac{\partial u_i(x)^T}{\partial x_s}A_t(x)u_j(x)\\ &\quad + u_i(x)^T\mathcal{A}(x)\frac{\partial^2 u_j(x)}{\partial x_s \partial x_t} + \frac{\partial u_i(x)^T}{\partial x_t}\mathcal{A}(x)\frac{\partial u_j(x)}{\partial x_s} + u_i(x)^T A_t(x)\frac{\partial u_j(x)}{\partial x_s}\\ &\quad + \frac{\partial u_i(x)^T}{\partial x_t}A_s(x)u_j(x) + u_i(x)^T A_s(x)\frac{\partial u_j(x)}{\partial x_t} + u_i(x)^T A_{st}(x)u_j(x).\end{aligned}$$

Moreover, since $u_i(x)^T u_j(x)$ is constant (zero if $i \ne j$ and one if $i = j$), we have

$$\begin{aligned}0 = \frac{\partial^2 [u_i(x)^T u_j(x)]}{\partial x_s \partial x_t} &= \frac{\partial^2 u_i(x)^T}{\partial x_s \partial x_t}u_j(x) + \frac{\partial u_i(x)^T}{\partial x_s}\frac{\partial u_j(x)}{\partial x_t}\\ &\quad + \frac{\partial u_i(x)^T}{\partial x_t}\frac{\partial u_j(x)}{\partial x_s} + u_i(x)^T\frac{\partial^2 u_j(x)}{\partial x_s \partial x_t}.\end{aligned}$$

Since $\mathcal{A}(x^*)u_j(x^*) = \lambda_j(x^*)u_j(x^*)$ and by (3.8) it follows from the above equations that

$$\frac{\partial^2 \phi_{ij}(x^*)}{\partial x_s \partial x_t} = e_i^T A_{st} e_j + \sum_{k\ne p+1,\ldots,q} (\delta^* - \lambda_k)^{-1}(\tau_{ik}^s \tau_{jk}^t + \tau_{ik}^t \tau_{jk}^s).$$

We obtain that the typical element $\partial^2 L(x^*,\alpha^*)/\partial x_s \partial x_t$ of Hessian matrix $\nabla_{xx}^2 L(x^*,\alpha^*)$ of the Lagrangian can be written as

$$\frac{\partial^2 L(x^*,\alpha^*)}{\partial x_s \partial x_t} = \sum_{i=1}^{p} e_i^T A_{st} e_i + \sum_{i,j=p+1}^{q} \beta_{ij} e_i^T A_{st} e_j + \sum_{i=1}^{p}\sum_{k\neq i} \frac{2\tau_{ik}^s \tau_{ik}^t}{\lambda_i - \lambda_k} + \sum_{i,j=p+1}^{q} \left(\beta_{ij} \sum_{k\neq p+1,\ldots,q} \frac{\tau_{ik}^s \tau_{jk}^t + \tau_{ik}^t \tau_{jk}^s}{\delta^* - \lambda_k} \right). \tag{3.11}$$

All terms in the above equations are calculated at x^*. Also if some of the eigenvalues $\lambda_1, \ldots, \lambda_p$ have multiplicity more than one, then the third term in the right-hand side of (3.11) should be corrected by writing in the corresponding projection matrices.

Now the standard second-order necessary conditions (e.g., [7]), for the constrained problem (3.2), and hence for the original problem (1.1), are

$$y^T \nabla^2_{xx} L(x^*,\alpha^*) y \geq 0 \text{ for all } y \in T_{x^*} V(p,q). \tag{3.12}$$

The corresponding second-order sufficient conditions for x^* to be a local minimizer of $g(x)$ over $V(p,q)$ are given by

$$y^T \nabla^2_{xx} L(x^*,\alpha^*) y > 0 \text{ for any nonzero } y \in T_{x^*} V(p,q). \tag{3.13}$$

We show now that under the additional conditions that the matrices B and $I_{q-p} - B$ are *positive* definite, conditions (3.13) are sufficient for x^* to be a locally optimal solution of the original problem (1.1).

THEOREM 3.1. *Suppose that $\mathcal{A}\pitchfork_{x^*} W(p,q)$, that the first-order necessary conditions* (3.4) *hold and that the matrices B and $I_{q-p} - B$ are positive definite. Then conditions* (3.13) *are sufficient for x^* to be a locally optimal solution of the problem* (1.1).

Proof. Let us make the following observations. The function $f(x)$ can be represented as the composition $f(x) = h(\mathcal{A}(x))$ of the convex function $h(A) = \sum_{i=1}^{c} \lambda_i(A)$, defined on the space $\mathcal{S}_n$, and the smooth mapping $\mathcal{A}(x)$. We have

$$\mathcal{A}(x+y) = \mathcal{A}(x) + \nabla\mathcal{A}(x)y + R(x,y).$$

By continuity of the second-order derivatives of $\mathcal{A}(x)$, the remainder term $R(x,y)$ in the above Taylor expansion is of order $O(\|y\|^2)$ uniformly in x. That is, for any compact, convex set $S \subset \mathbb{R}^m$ there is a positive constant K, independent of x and y, such that $\|R(x,y)\| \leq K\|y\|^2$ for all x and y with $x, x+y \in S$. Since $h(A)$ is Lipschitz continuous on any bounded subset of $\mathcal{S}_n$, we obtain

$$f(x+y) = h(\mathcal{A}(x) + \nabla\mathcal{A}(x)y) + r(x,y),$$

where the term $r(x,y)$ is of order $O(\|y\|^2)$ uniformly in x in a bounded subset of $\mathbb{R}^m$. Moreover, by convexity of the function h we have that

$$h(\mathcal{A}(x) + \nabla\mathcal{A}(x)y) \geq h(\mathcal{A}(x)) + h'(\mathcal{A}(x), \nabla\mathcal{A}(x)y),$$

where $h'(A,D)$ denotes the directional derivative of $h(\cdot)$ at A in the direction D. By the chain rule for directional derivatives

$$h'(\mathcal{A}(x), \nabla\mathcal{A}(x)y) = f'(x,y),$$

and hence

$$f(x+y) \geq f(x) + f'(x,y) + r(x,y). \tag{3.14}$$

Now since $f(x)$ is a composition of the convex function h and the smooth mapping $\mathcal{A}$, its directional derivatives can be written in the form

$$f'(x,y) = \max_{z \in \partial f(x)} z^T y,$$

where $\partial f(x)$ is a convex, compact set called the generalized gradient of f at x. It can be shown by methods of convex analysis (cf. [14]) that

$$\partial f(x^*) = \left\{ \sum_{i=1}^{p} v_{ii}(x^*) + \sum_{i,j=p+1}^{q} \gamma_{ij} v_{ij}(x^*) : \sum_{i=p+1}^{q} \gamma_{ii} = c - p,\ \Gamma \in \mathcal{P}_{q-p} \right\}, \tag{3.15}$$

where $\Gamma = [\gamma_{ij}]$ and $\mathcal{P}_r$ denotes the set of $r \times r$ symmetric matrices Σ such that $\Sigma \geq 0$ and $\Sigma \leq I_r$. The first-order necessary conditions mean that $0 \in \partial f(x^*)$. Moreover, since $0 < B < I_{q-p}$, it follows from (3.4) that 0 belongs to the relative interior of the convex set $\partial f(x^*)$.

Consider now the linear space generated by the convex set $\partial f(x^*)$. It follows from (3.15) and (2.4) that this linear space coincides with the orthogonal complement to the tangent space $T_{x^*}V(p,q)$. Consequently we obtain that it follows from the positive definiteness of the matrices B and $I_{q-p} - B$ that $f'(x^*, y) = 0$ if and only if $y \in T_{x^*}V(p,q)$ and that $f'(x^*, y) \geq \alpha \|y\|$ for some $\alpha > 0$ and all y orthogonal to $T_{x^*}V(p,q)$.

Consider a point x in a neighborhood of x^*. Let $\bar{x}$ be a point in $V(p,q)$ closest to x, i.e., $\bar{x} \in V(p,q)$ and $\|x - \bar{x}\| = \operatorname{dist}(x, V(p,q))$. We have then that $x - \bar{x}$ is orthogonal to the tangent space $T_{\bar{x}}V(p,q)$. Note that the generalized gradient $\partial f(\bar{x})$ can be written in a way similar to (3.15) with vectors $v_{ij}(x^*)$ replaced by $v_{ij}(\bar{x})$. Consequently, by the arguments of continuity, we obtain that for all x in a neighborhood of x^*, $f'(\bar{x}, x - \bar{x}) \geq \frac{1}{2}\alpha \|x - \bar{x}\|$. Together with (3.14) this implies that $f(x) > f(\bar{x})$ for all x sufficiently close to x^*. It remains to note that by the second-order conditions (3.13), $f(\bar{x}) > f(x^*)$ for all $\bar{x} \in V(p,q)$ sufficiently close to x^* and hence $f(x) > f(x^*)$. □

4. Optimization algorithms. As we mentioned earlier the main idea behind a smooth (differentiable) approach to the optimization problem (1.1) is to restrict the minimization procedure to the smooth manifold $V(p,q)$ for appropriately chosen p and q. That is, if the algorithm generates a point sufficiently close to $V(p,q)$, then instead of minimizing $f(x)$ over $\mathbb{R}^m$ one solves the corresponding constrained problem of minimization of $g(x)$ subject to $x \in V(p,q)$. The integers p and q can be updated in the process of optimization. Properly constructed such an algorithm will converge to a point x^* satisfying the first-order optimality conditions (3.4). Recall that the associated multipliers β_{ij} are unique provided the transversality condition holds. Therefore the additional condition that the corresponding matrices B and $I_{q-p} - B$ must be nonnegative definite can be easily verified.

Consider the constrained problem (3.2) and let x^k be a current iteration point sufficiently close to $V(p,q)$. The projection matrix $P(x) = E(x)E(x)^T$ onto the space generated by the eigenvectors $e_{p+1}(x), \ldots, e_q(x)$, is then differentiable at x^k and can be locally represented in a way similar to the corresponding representation

specified in the previous section. That is, $P(x) = U^{(k)}(x)U^{(k)}(x)^T$ with the matrix $U^{(k)}(x)$ being a differentiable function of x and such that $U^{(k)}(x)^T U^{(k)}(x) = I_{q-p}$, $U^{(k)}(x^k) = E(x^k)$ and $U^{(k)}(x^k)^T dU^{(k)}(x^k) = 0$. The manifold $V(p,q)$ can be locally defined by (compare with (3.6))

$$U^{(k)}(x)^T \mathcal{A}(x) U^{(k)}(x) = \delta I_{q-p}. \tag{4.1}$$

Note that differentiation (linearization) of (4.1) at $x = x^k$ leads to (3.10) with all quantities calculated at the point x^k.

One step in Overton's algorithm can be described as follows. Solve the quadratic program

$$\min_{d,\mu} \quad d^T \nabla g(x^k) + \frac{1}{2} d^T H^k d \tag{4.2}$$

$$\text{subject to} \quad \text{Diag}\,(0, \lambda_{p+2}(x^k) - \lambda_{p+1}(x^k), \ldots, \lambda_q(x^k) - \lambda_{p+1}(x^k))$$

$$+ \sum_{i=1}^{m} d_i E(x^k)^T A_i(x^k) E(x^k) - \mu I_{q-p} = 0. \tag{4.3}$$

Put $x^{k+1} = x^k + d^{k+1}$, where d^{k+1} is the optimal solution of the above quadratic program. The constraints (4.3) are obtained by linearization of (4.1). Variable $\mu \in \mathbb{R}$ corresponds to $d\delta$, and the current value of δ at x^k is taken $\delta^k = \lambda_{p+1}(x^k)$. The matrix H^k represents the Hessian matrix of the Lagrangian calculated at x^k and can be calculated in a way similar to the calculations specified in the previous section (see (3.11)).

Note that

$$d^T \nabla g(x^k) = \sum_{i=1}^{m} d_i \text{tr}\left[\tilde{E}(x^k)^T A_i(x^k) \tilde{E}(x^k) + \frac{c-p}{q-p} E(x^k)^T A_i(x^k) E(x^k)\right],$$

where $\tilde{E}(x) = [e_1(x), \ldots, e_p(x)]$. For d satisfying (4.3), the second term in the right-hand side of the above equation reduces to $(c-p)\mu - b(x^k)$, where $b(x^k) = \text{tr}[\text{Diag}\,(0, \lambda_{p+2}(x^k) - \lambda_{p+1}(x^k), \ldots, \lambda_q(x^k) - \lambda_{p+1}(x^k))]$. Overton's algorithm [12], [13] was derived for the case $c = 1$ ($p = 0$), and uses μ instead of $d^T \nabla g(x^k)$ in (4.2). Also additional inequality constraints are imposed in [12], [13] to prevent d^{k+1} from having too large norm. A linearization similar to (4.2)–(4.3) was suggested in [27] and [28].

It should be noted that (4.1), as well as their linearization (4.3), are related to the eigenvectors of the matrix $\mathcal{A}(x)$ calculated at the point $x = x^k$. Therefore, although the feasible set $V(p,q)$ of the constrained problem is fixed, at least locally, the corresponding equations (4.1) can change from iteration to iteration. In particular this means that the current value of the Lagrange multipliers, used in the calculation of the Hessian matrix H^k, cannot be taken from the previous iteration and should be calculated at every iteration, say, by the least squares method (cf. [13], [28]). Consequently Overton's algorithm is not the standard Newton's method. The name "sequential Newton method" probably will be more appropriate for that type of algorithm. With a little bit of additional effort it is still possible to show that typically the method has a locally quadratic rate of convergence (cf. [15]). For a detailed discussion of the involved regularity conditions see the Appendix.

In [6], Fan proposed a quadratically convergent algorithm for solving the constraint problem (3.2) (for the case $c = 1$). His algorithm is also applicable to a more general class of problems [11]. We now describe Fan's algorithm in the context of

solving (3.2) for general $c \geq 1$. Another treatment of (3.2) using Fan's algorithm can be found in [11].

Consider the function $\phi(x)$ defined by

$$\phi(x) = \sum_{i,j=p+1}^{q} (\lambda_i(x) - \lambda_j(x))^2.$$

In a sense this function measures a distance between x and the manifold $V(p,q)$. It is clear that in a neighborhood of x^*, $x \in V(p,q)$ if and only if $\phi(x) = 0$. It can be shown that $\phi(x)$ is analytic at x^* [11]. Moreover, its Hessian matrix $\nabla^2\phi(x)$ can be written as the sum of two symmetric matrices $\Phi_1(x)$ and $\Phi_2(x)$, satisfying the following properties: (i) $\Phi_1(x)$ is nonnegative definite, (ii) the range space of $\Phi_1(x)$ coincides with the linear space generated by vectors $v_{ij}(x)$, $p+1 \leq i < j \leq q$; $v_{ii}(x) - v_{qq}(x)$, $i = p+1, \ldots, q-1$; and (iii) $\Phi_2(x) = 0$ if $\phi(x) = 0$ (cf. [11]).

Let us consider matrices $N(x)$ and $R(x)$ formed by a set of orthonormal column vectors such that the column vectors of $N(x)$ and $R(x)$ generate the null and the range space of $\Phi_1(x)$, respectively. Note that it follows that $R(x)^T N(x) = 0$ and $R(x)R(x)^T + N(x)N(x)^T = I_m$. Moreover, because of the above property (ii), the linear space generated by the columns of $N(x)$ coincides with the tangent space $T_x V(p,q)$ provided $x \in V(p,q)$. Consequently the first-order necessary conditions for the constraint problem (3.2) can be written in the form $N(x^*)^T \nabla g(x^*) = 0$.

Now it is not difficult to see that $\nabla\phi(x) = 0$ if $x \in V(p,q)$. Moreover, using the Taylor expansion $\nabla\phi(x+h) = \Phi_1(x)h + o(\|h\|)$, at $x \in V(p,q)$, and the fact that $\Phi_1(x)h \neq 0$ if $h \notin T_x V(p,q)$, we obtain that, in a neighborhood of x^*, $R(x)^T \nabla\phi(x) = 0$ if and only if $x \in V(p,q)$. That is, equations $R(x)^T \nabla\phi(x) = 0$ locally define the manifold $V(p,q)$. We obtain that the optimal solution x^* can be derived as a solution of the following system of m equations

$$N(x)^T \nabla g(x) = 0 \quad \text{and} \quad R(x)^T \nabla\phi(x) = 0. \tag{4.4}$$

One step of Fan's algorithm consists in linearization of the nonlinear system (4.4), at a current iteration point x^k, and consequent updating of x^k by the solution of the obtained system of linear equations.

It can be shown that the matrices $N(x)$ and $R(x)$ can be chosen as smooth functions of x and their differentials can be calculated as

$$dN(x) = -\Phi_1^{\dagger}(x)[d\Phi_1(x)]N(x),$$

$$dR(x) = N(x)N(x)^T[d\Phi_1(x)]\Phi_1^{\dagger}(x)R(x),$$

where $\Phi_1^{\dagger}(x)$ denotes the Moore–Penrose generalized inverse of $\Phi_1(x)$ [11]. By using these formulas it is possible to construct the required linearization of the system (4.4). Take then $x^{k+1} = x^k + h^{k+1}$, where h^{k+1} is the solution of the linearized, at $x = x^k$, system. It follows then from the standard convergence theory for the Newton method (cf. [5]) that if we choose the starting point sufficiently close to the minimizer x^*, then the algorithm converges to x^* quadratically.

5. Sensitivity analysis. Consider now a situation when the mapping $\mathcal{A}$ depends on a parameter vector $\pi \in \Pi$. That is, let Π be an ℓ-dimensional linear space, $F : \mathbb{R}^m \times \Pi \to \mathcal{S}_n$ be a smooth mapping and let $\mathcal{A}_\pi(\cdot) = F(\cdot, \pi)$ be the associated

parametric family of mappings $\mathcal{A}_\pi : \mathbb{R}^m \to \mathcal{S}_n$. We assume that for some $\pi_0 \in \Pi$, $\mathcal{A}(\cdot) = F(\cdot, \pi_0)$, i.e., the considered mapping $\mathcal{A}(x)$ belongs to the specified parametric family. The optimal value and an optimal solution of the considered optimization problem can be viewed as functions of π. Denote the corresponding optimal value function by $\psi(\pi)$ and the optimal solution by $\bar{x}(\pi)$. In this section we discuss how $\psi(\pi)$ and $\bar{x}(\pi)$ vary under small perturbations of the parameter vector π. More specifically we study differentiability properties of $\psi(\pi)$ and $\bar{x}(\pi)$ at $\pi = \pi_0$.

For the sake of simplicity let us consider the problem of minimization of the largest eigenvalue $\lambda_1(x, \pi)$ of the matrix $F(x, \pi)$. This problem can be formulated as the semi-infinite programming problem

$$\begin{array}{l} \min_{x,\lambda} \ \lambda \\ \text{subject to } y^T F(x, \pi) y - \lambda \leq 0, \ y \in \mathbb{R}^n, \ \|y\| = 1. \end{array} \tag{5.1}$$

Let (x^*, λ^*) be an optimal solution of (5.1) for $\pi = \pi_0$. Note that $x^* = \bar{x}(\pi_0)$ is the minimizer of $\lambda_1(\cdot\,, \pi_0)$ and $\lambda^* = \lambda_1(x^*, \pi_0) = \psi(\pi_0)$ is the corresponding optimal value. Note also that $y^T \mathcal{A}(x^*) y = \lambda^*$, $\|y\| = 1$, if and only if y is an eigenvector of $\mathcal{A}(x^*)$ corresponding to the largest eigenvalue.

First-order (Fritz John) necessary conditions for the semi-infinite program (5.1) are well known (e.g., [17]). After some algebraic manipulations (cf. [21]) these conditions can be formulated in the form of (3.4). That is, let q be the multiplicity of the largest eigenvalue of $\mathcal{A}(x^*)$. Then there exist multipliers β_{ij} such that

$$\begin{array}{l} \sum_{i,j=1}^q \beta_{ij} v_{ij}(x^*) = 0, \\ \beta_{ij} = \beta_{ji}, \ i, j = 1, \ldots, q, \text{ and } \sum_{i=1}^q \beta_{ii} = 1, \end{array} \tag{5.2}$$

and the $q \times q$ matrix $B = [\beta_{ij}]$ is nonnegative definite. (Note that here the condition $B \leq I_q$ holds automatically.) Suppose that the minimizer x^* is *unique*. It is known then that, under certain regularity conditions specified below, the optimal value function of the semi-infinite program (5.1) is directionally differentiable at $\pi = \pi_0$ and its directional derivative is given by the maximum of the directional derivatives of the Lagrangian corresponding to (5.1) at $x = x^*$ and taken with respect to the associated set of Lagrange multipliers (see [24], [25], [29] for details). The corresponding formula for the directional derivatives can be written then in the form

$$\psi'(\pi_0; d) = \max_{B \in \mathcal{B}} d^T \nabla_\pi [tr F(x^*, \pi_0) Q B Q^T], \tag{5.3}$$

where Q is the $n \times q$ matrix formed from a set of orthonormal eigenvectors of $\mathcal{A}(x^*) = F(x^*, \pi_0)$ corresponding to the largest eigenvalue and $\mathcal{B}$ is the set of nonnegative definite symmetric matrices satisfying optimality conditions (5.2).

In general, formula (5.3) holds under certain second-order sufficient conditions associated with the program (5.1) (see [24]). Verification of these conditions, however, may be not easy. Nevertheless there are two situations when applicability of formula (5.3) can be easily verified. One such case is when the program (5.1) is *convex*. For instance, let the mapping F be affine in x,

$$F(x, A) = A + x_1 A_1 + \cdots + x_m A_m,$$

and $\mathcal{A}(\cdot) = F(\cdot\,, A_0)$. Then the program (5.1) is convex and formula (5.3) holds, provided the minimizer x^* does exist and is unique and the Slater and the so-called inf-compactness conditions hold; see [25] and [29]. (The inf-compactness condition is

needed to ensure that $\bar{x}(\pi)$ tends to x^* as $\pi \to \pi_0$.) In that case formula (5.3) takes the form

$$\psi'(A_0; D) = \max_{B \in \mathcal{B}} \operatorname{tr} DQBQ^T. \tag{5.4}$$

Another case when (5.3) holds is if the inf-compactness condition is satisfied and the set $\mathcal{B}$ is a singleton, i.e., there is a unique matrix B satisfying (5.2). Recall that the matrix B corresponds to Lagrange multipliers of the problem (5.1) and that B is unique if the corresponding transversality condition holds. In that case $\psi(\pi)$ is differentiable at π_0 and

$$\nabla\psi(\pi_0) = \nabla_\pi[\operatorname{tr} F(x^*, \pi_0)QBQ^T]. \tag{5.5}$$

Consider now the optimal solution $\bar{x}(\pi)$. Suppose that the transversality condition holds and that $\bar{x}(\pi)$ tends to x^* as $\pi \to \pi_0$. It follows then that the Lagrange multipliers matrix B satisfying (5.2) is unique. Suppose further that the matrix B is *nonsingular* and hence is positive definite. By the arguments of continuity we obtain then that the Lagrange multipliers matrix is nonsingular, and hence the largest eigenvalue of $F(\bar{x}(\pi), \pi)$ has multiplicity q, for all π in a neighborhood of π_0. Therefore locally, for all π near π_0, the considered optimization problem is equivalent to the smooth problem of minimization of the function $\sum_{i=1}^q \lambda_i(x, \pi)$ subject to $\lambda_1(x, \pi) = \cdots = \lambda_q(x, \pi)$. It follows by the Implicit Function Theorem that, under the corresponding second-order sufficient conditions, $\bar{x}(\pi)$ is continuously differentiable at π_0 and its differential $d\bar{x}(\pi_0)(\xi)$ is given by the optimal solution of the quadratic program (cf. [8], [22])

$$\begin{array}{lll} \min_d & & d^T H_{xx} d + 2d^T H_{x\pi}\xi + \xi^T H_{\pi\pi}\xi \\ \text{subject to} & & \sum_{s=1}^m d_s e_i^T A_s e_j + \sum_{t=1}^\ell \xi_t e_i^T Z_t e_j = 0, \quad 1 \le i < j \le q, \\ & & \sum_{s=1}^m d_s e_i^T A_s e_i + \sum_{t=1}^\ell \xi_t e_i^T Z_t e_i = \delta, \quad i = 1, \ldots, q. \end{array} \tag{5.6}$$

Here $e_1, \ldots, e_n$ is an orthonormal set of eigenvectors of the matrix $\mathcal{A}(x^*)$, $A_s = \partial F(x^*, \pi_0)/\partial x_s$, $s = 1, \ldots, m$; $Z_t = \partial F(x^*, \pi_0)/\partial \pi_t$, $t = 1, \ldots, \ell$; δ is an additional parameter and H_{xx}, $H_{x\pi}$, $H_{\pi\pi}$ are the respective Hessian matrices of the corresponding Lagrangian calculated at (x^*, π_0) (see (3.11)). Moreover, under the above conditions, the optimal value function $\psi(\pi)$ is twice continuously differentiable at π_0 and $\psi(\pi_0 + \xi) = \psi(\pi_0) + \xi^T\nabla\psi(\pi_0) + \frac{1}{2}\kappa(\xi) + o(\|\xi\|^2)$, where $\kappa(\xi)$ is the optimal value of the program (5.6).

6. Appendix. In this Appendix we discuss regularity conditions required to ensure locally quadratic rate of convergence of a sequential Newton's algorithm.

Suppose we want to minimize a smooth function $f(x)$ over a smooth manifold $V \subset \mathbb{R}^m$. Suppose further that in a neighborhood of the optimal solution point x^* the manifold V can be explicitly defined by a system of smooth equations $g_i(x) = 0$, $i = 1, \ldots, p$. In this case the standard Newton method can be applied. That is, let x^k be a current point generated by the algorithm and α^k be a corresponding vector of Lagrange multipliers. Then the next iteration is calculated as $x^{k+1} = x^k + d^{k+1}$, where d^{k+1} is the optimal solution of the quadratic programming problem

$$\begin{array}{lll} \min & & d^T\nabla f(x^k) + \frac{1}{2} d^T H^k d \\ \text{subject to} & & g_i(x^k) + d^T\nabla g_i(x^k) = 0, \quad i = 1, \ldots, p. \end{array} \tag{6.1}$$

Here $H^k = \nabla^2_{xx} L(x^k, \alpha^k)$ is the Hessian matrix of the Lagrangian

$$L(x, \alpha) = f(x) + \sum_{i=1}^{p} \alpha_i g_i(x).$$

Note that x^{k+1} and the corresponding vector α^{k+1} of Lagrange multipliers can be obtained as a solution of the linear equations

$$F(z^k) + \nabla F(z^k)(z - z^k) = 0, \tag{6.2}$$

where $F(z) = (\nabla_x L(x, \alpha), g(x))$, $g(x) = (g_1(x), \ldots, g_p(x))$ and $z = (x, \alpha)$.

It is well known that if the algorithm starts at a point sufficiently close to the optimal solution x^* and the second-order sufficient optimality conditions hold at x^*, then the algorithm converges quadratically. Consider now a situation when there is an additional complication that the system of equations which defines the manifold V depends on the point x^k and can change from iteration to iteration. That is, let N_{x^*} be a neighborhood of the point x^* and $g_i^{(k)}$, $i = 1, \ldots, p$, be smooth functions associated with the iteration point x^k, such that $V \cap N_{x^*} = \{x \in N_{x^*} : g_i^{(k)}(x) = 0,\ i = 1, \ldots, p\}$. The above Newton procedure is then modified by employing the functions $g_i^{(k)}$ at kth iteration. Let us briefly discuss the obtained algorithm, referred to as the sequential Newton method.

Given a current iteration point x^k and a corresponding Lagrange multipliers vector α^k the algorithm calculates $x^{k+1} = x^k + d^{k+1}$, where d^{k+1} is a solution of the quadratic programming problem (6.1) with the functions $g_i^{(k)}$ replacing the corresponding functions g_i and $H^k = \nabla^2_{xx} L^{(k)}(x^k, \alpha^k)$ being the Hessian matrix of the Lagrangian

$$L^{(k)}(x, \alpha) = f(x) + \sum_{i=1}^{p} \alpha_i g_i^{(k)}(x).$$

Similar to (6.2), x^{k+1} can be obtained as the first component of the solution $z^{k+1} = (x^{k+1}, \alpha^{k+1})$ of the linear equations

$$F^{(k)}(z^k) + \nabla F^{(k)}(z^k)(z - z^k) = 0, \tag{6.3}$$

where $F^{(k)}(z) = (\nabla_x L^{(k)}(x, \alpha), g^{(k)}(x))$. Note, however, that the calculated Lagrange multipliers vector α^{k+1} cannot be used for the next iteration and this is because it is calculated with respect to the functions $g_i^{(k)}$, which can be changed at the next iteration. Note also that by the first-order necessary conditions for every k there is a Lagrange multipliers vector α^{*k} corresponding to the optimal solution x^* such that $\nabla_x L^{(k)}(x^*, \alpha^{*k}) = 0$. Consequently $F^{(k)}(z^{*k}) = 0$, where $z^{*k} = (x^*, \alpha^{*k})$.

Consider the Taylor expansion

$$F^{(k)}(z) = F^{(k)}(z^k) + \nabla F^{(k)}(z^k)(z - z^k) + R^{(k)}(z) \tag{6.4}$$

of $F^{(k)}$ at z^k with the remainder term $R^{(k)}(z)$. It follows from (6.3) and (6.4) that

$$\nabla F^{(k)}(z^k)(z^{k+1} - z^{*k}) = R^{(k)}(z^{*k}). \tag{6.5}$$

Now let us make the following assumptions.

(i) There is a constant γ such that $\|\alpha^k\| \leq \gamma$ and $\|\alpha^{*k}\| \leq \gamma$ for all k.

(ii) The remainder term $R^{(k)}(z)$ is of order $O(\|z - z^k\|^2)$ uniformly in k. That is, there is a constant K such that

$$\|R^{(k)}(z)\| \leq K\|z - z^k\|^2 \tag{6.6}$$

for all x in the neighborhood N_{x^*}, all α such that $\|\alpha\| \leq \gamma$ and all k.

(iii) The matrices $\nabla F^{(k)}(z_k)$ are uniformly bounded from being singular. That is, there is a constant c such that

$$\|\nabla F^{(k)}(z^k)w\| \geq c\|w\| \tag{6.7}$$

for all w and all k.

(iv) The Lagrange multipliers vector α^k is chosen in such a way that $\|\alpha^k - \alpha^{*k}\|$ is of order $O(\|x^k - x^*\|)$. That is, there is a constant κ such that

$$\|\alpha^k - \alpha^{*k}\| \leq \kappa\|x^k - x^*\| \tag{6.8}$$

for all k.

It follows then from (6.5) that

$$c\|z^{k+1} - z^{*k}\| \leq K\|z^{*k} - z^k\|^2.$$

Consequently,

$$\|x^{k+1} - x^*\| \leq \|z^{k+1} - z^{*k}\| \leq c^{-1}K(\|x^k - x^*\|^2 + \|\alpha^k - \alpha^{*k}\|^2)$$

and hence

$$\|x^{k+1} - x^*\| \leq c^{-1}K(1 + \kappa^2)\|x^k - x^*\|^2 \tag{6.9}$$

for all x^k sufficiently close to x^*. We obtain that assumptions (i)–(iv) imply locally quadratic rate of convergence of the algorithm.

A few remarks about the regularity assumptions (i)–(iv) are now in order. Since $L^{(k)}(x, \alpha)$ is linear in α, assumption (ii) is satisfied if the remainder term in the first-order Taylor expansions of $g_i^{(k)}$ and $\nabla g_i^{(k)}$, $i = 1, \ldots, p$, at x^k, is of order $O(\|x - x^k\|^2)$ uniformly in k. This holds, for example, if $\nabla^2 g_i^{(k)}$, $i = 1, \ldots, p$, are Lipschitz continuous in the neighborhood N_{x^*} with the corresponding Lipschitz constant independent of k. Consider now (4.1) defining, locally, the manifold $V(p, q)$. The matrix $U^{(k)}(x)$ is given there by a matrix in the manifold $\mathcal{M}(x)$, defined in (3.5), which minimizes the distance from $E(x^k)$ to $\mathcal{M}(x)$. Again, since $\mathcal{M}(x^*)$ is compact, it follows from the Implicit Function Theorem that $U^{(k)}(x)$ is a smooth function of x and $E = E(x^k)$ for all x and E sufficiently close to x^* and $\mathcal{M}(x^*)$, respectively. It follows then by continuity arguments that the neighborhood where (4.1) define the manifold $V(p, q)$ can be chosen independently of k for all x^k sufficiently close to x^* and that, say the third-order, derivatives of $U^{(k)}(x)$ are bounded in a neighborhood of x^* uniformly in k.

Assumption (iv) suggests that α^k should be sufficiently close to the Lagrange multipliers vector α^{*k} corresponding to the optimal solution point x^*. It is natural to choose α^k by the least squares method, i.e., to calculate α^k as the minimizer of the function $\psi(\alpha) = \|\nabla f(x^k) + \sum_{i=1}^p \alpha_i \nabla g_i^{(k)}(x^k)\|^2$. Such choice of α^k ensures

assumption (iv) provided $\nabla g_i^{(k)}$, $i = 1, \ldots, p$, are Lipschitz continuous on N_{x^*} with the Lipschitz constant independent of k. Finally let us remark that

$$\nabla F^{(k)}(z^k) = \begin{bmatrix} H^k & G^k \\ G^{kT} & 0 \end{bmatrix},$$

where $H^k = \nabla^2_{xx} L^{(k)}(x^k, \alpha^k)$ and $G^k = \nabla g^{(k)}(x^k)$. It is not difficult to show then that the matrix $\nabla F^{(k)}(z^k)$ is nonsingular if G^k has full column rank p and $x^T H^k x > 0$ for any nonzero x such that $G^{kT} x = 0$. Recall that the second-order sufficient conditions here can be formulated in the form $x^T H^{*k} x > 0$ for any nonzero x such that $G^{*kT} x = 0$, where $H^{*k} = \nabla^2_{xx} L^{(k)}(x^*, \alpha^{*k})$ and $G^{*k} = \nabla g^{(k)}(x^*)$, [7].

Acknowledgments. We are indebted to the editor, Michael L. Overton, and anonymous referees for their careful reading and suggestions for improving the paper.

REFERENCES

[1] T.J. ABATZOGLOU, *The minimum norm projection on C^2 manifold in R^n*, Trans. Amer. Math. Soc., 243 (1978), pp. 115–122.

[2] V.I. ARNOLD, *On matrices depending on parameters*, Russian Math. Surveys, 26 (1971), pp. 29–43.

[3] J. CULLUM, W.E. DONATH, AND P. WOLFE, *The minimization of certain nondifferentiable sums of eigenvalues of symmetric matrices*, Math. Programming Study, 3 (1975), pp. 35–55.

[4] M. GOLUBITSKY AND V. GUILLEMIN, *Stable Mappings and their Singularities*, Springer-Verlag, New York, 1973.

[5] J. GOODMAN, *Newton's method for constrained optimization*, Math. Programming, 33 (1985), pp. 162–171.

[6] M.K.H. FAN, *A quadratically convergent local algorithm on minimizing the largest eigenvalue of a symmetric matrix*, Linear Algebra Appl., to appear.

[7] A.V. FIACCO AND G.P. MCCORMICK, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, Wiley, New York, 1968.

[8] A.V. FIACCO, *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*, Academic Press, New York, 1983.

[9] R. FLETCHER, *Semi-definite constraints in optimization*, SIAM J. Control Optim., 23 (1985), pp. 493–513.

[10] T. KATO, *Perturbation Theory for Linear Operators*, Springer-Verlag, New York, 1984.

[11] B. NEKOOIE AND M.K.H. FAN, *A quadratically convergent local algorithm on minimizing sums of the largest eigenvalues of a symmetric matrix*, Comput. Optim. Appl., to appear.

[12] M.L. OVERTON, *On minimizing the maximum eigenvalue of a symmetric matrix*, SIAM J. Matrix Anal. Appl., 9 (1988), pp.256–268.

[13] ———, *Large-scale optimization of eigenvalues*, SIAM J. Optim., 2 (1992), pp. 88–120.

[14] M.L. OVERTON AND R.S. WOMERSLEY, *Optimality conditions and duality theory for minimizing sums of the largest eigenvalues of symmetric matrices*, Math. Programming, to appear.

[15] ———, *Second derivatives for optimizing eigenvalues of symmetric matrices*, preprint, March 1993.

[16] E. POLAK AND Y. WARDI, *Nondifferentiable optimization algorithm for the design of control systems subject to singular value inequalities*, Automatica, 18 (1982), pp. 267–283.

[17] B.N. PSHENICHNYI, *Necessary Conditions for an Extremum*, Marcel Dekker, New York, 1971.

[18] 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.

[19] A. SHAPIRO, *Weighted minimum trace factor analysis*, Psychometrika, 46 (1982), pp. 201–213.

[20] ———, *On the unsolvability of inverse eigenvalue problems almost everywhere*, Linear Algebra Appl., 49 (1983), pp. 27–31.

[21] ———, *Extremal problems on the set of nonnegative definite matrices*, Linear Algebra Appl., 67 (1985), pp. 7–18.

[22] ———, *Sensitivity analysis of nonlinear programs and differentiability properties of metric projections*, SIAM J. Control Optim., 26 (1988), pp. 628–645.

[23] A. SHAPIRO AND J.D. BOTHA, *Dual algorithms for orthogonal Procrustes rotations*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 378–383.

[24] A. SHAPIRO, *On Lipschitzian stability of optimal solutions of parametrized semi-infinite programs*, Math. Oper. Res., 19 (1994), pp. 743–752.

[25] ———, *Directional differentiability of the optimal value function in convex semi-infinite programming*, Math. Programming, Series A, to appear.

[26] J.M.F. TEN BERGE, *Orthogonal Procrustes rotation for two or more matrices*, Psychometrika, 42 (1977), pp. 267–276.

[27] G.A. WATSON, *An algorithm for optimal l_2 scaling of matrices*, IMA J. Numer. Anal, 11 (1991), pp. 481–492.

[28] ———, *Algorithms for minimum trace factor analysis*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1039–1053.

[29] P. ZENCKE AND R. HETTICH, *Directional derivatives for the value-function in semi-infinte programming*, Math. Programming, 38 (1987), pp. 323–340.

SIAM J. OPTIMIZATION
Vol. 5, No. 3, pp. 570–589, August 1995

006

DATA PARALLEL QUADRATIC PROGRAMMING ON BOX-CONSTRAINED PROBLEMS*

MIKE P. MCKENNA†, JILL P. MESIROV†, AND STAVROS A. ZENIOS‡

Abstract. We develop designs for the data parallel solution of quadratic programming problems subject to box constraints. In particular, we consider the class of algorithms that iterate between projection steps that identify candidate active sets and conjugate gradient steps that explore the working space. Using the algorithm of Moré and Toraldo [Report MCS-p77-05 89, Argonne National Laboratory, Illinois, 1989] as a specific instance of this class of algorithms we show how its components can be implemented efficiently on a data-parallel SIMD computer architecture. Alternative designs are developed for both arbitrary, unstructured Hessian matrices and for structured problems.

Implementations are carried out on a Connection Machine CM–2. They are shown to be very efficient, achieving a peak computing rate over 2 Gflops. Problems with several hundred thousand variables are solved within one minute of solution time on the 8K CM–2. Extremely large test problems, with up to 2.89 million variables, are also solved efficiently. The data parallel implementation outperforms a benchmark implementation of interior point algorithms on an IBM 3090-600S vector supercomputer and a successive overrelaxation algorithm on an Intel iPSC/860 hypercube.

Key words. large-scale optimization, conjugate gradient algorithm, data structures

AMS subject classifications. 65K05, 90C08

1. Introduction.

We consider the box-constrained quadratic program (BQP):

$$\text{Minimize} \quad q(x) = \frac{1}{2}x^\top Hx + c^\top x \tag{1}$$

$$\text{Subject to} \quad \ell \le x \le u.$$

H is an $n \times n$ positive definite matrix and c, ℓ, u are given vectors in $\mathbf{R}^n$. We use Ω to denote the feasible set $\{\Omega = x | \ell \le x \le u\}$ and $\nabla q(x)$ to denote the gradient vector.

Models of this form arise in several areas of application, especially in problems from optimal control and engineering. Other significant areas of application include computerized tomography (see Herman [12]), linear least squares problems with bounded variables, and portfolio optimization; see, e.g., the papers in Zenios [23]. References to the many diverse engineering and optimal control applications are given in the introductions of Dembo and Tulowitzki [7] and Moré and Toraldo [19].

Several authors proposed algorithms for solving large scale instances of BQP. A popular approach is to use an active-set algorithm that solves a sequence of subproblems of the form

$$\text{Minimize} \quad q(x^k + d) \tag{2}$$

$$\text{Subject to} \quad d_i = 0 \qquad \text{for all } i \in W_k.$$

Here W_k is the index set of *active constraints*, indicating the set of variables that would remain fixed at one of their bounds. Within the active set framework the

* Received by the editors September 14, 1992; accepted for publication (in revised form) May 3, 1994.

† Thinking Machines Corporation, 245 First Street, Cambridge, Massachusetts 02142.

‡ Operations and Information Management Department, The Wharton School, University of Pennsylvania, Philadelphia, Pennsylvania 19104 and University of Cyprus, Nicosia, Cyprus (zenioss@zeus.cc.ucy.ac.cy). The work of this author was partially supported by National Science Foundation grants CCR–8811135 and SES–91–00216 and Air Force Office of Scientific Research grant 91–0168.

algorithm needs a step that identifies a candidate active set and a step that solves (2). Bertsekas [2] and others, e.g., Dunn [10], Dembo and Tulowitzki [7], Morè and Toraldo [18], [19], and Wright [22], proposed the use of a gradient projection method as the active set identification step. Gradient projection identifies the optimal active set in a finite number of steps under a nondegeneracy assumption, but it requires exact solutions of (2). A Newton-type projection algorithm with superlinear rate of convergence was proposed by Bertsekas and Gafni [3]. The algorithm still requires the exact solution of an equality-constrained quadratic program at each step.

Dembo and Tulowitzki [7] developed a framework that allows inexact minimizations — with increasing level of accuracy — on consecutive working spaces. Furthermore, this algorithm would modify the working space by several constraints at a time. Modifications to their scheme and convergence results were established by Wright [22] who also applied this algorithm with some success to the optimization of augmented Lagrangians. Moré and Toraldo [19] blended gradient projection and conjugate gradient algorithms into a general convergent framework. Their algorithm has finite convergence for nondegenerate problems, and permits inexact minimizations on the working space. Moré [20] showed that this algorithm vectorizes and parallelizes well for structured test problems.

More recent work specializes interior point algorithms for BQP. Han, Pardalos and Ye [11] applied a primal-dual interior point algorithm to solve large instances of BQP. They conducted extensive numerical experiments on an IBM 3090-600S vector supercomputer. Their results indicate that the algorithm requires very few steps and is very efficient. De Leone [15] specialized iterative successive overrelaxation (SOR) methods for BQP, showed that they can be easily implemented on a distributed memory hypercube, and reported encouraging results solving large scale problems on an Intel iPSC/860. Other recent work on the solution of BQP is the paper by Conn, Gould, and Toint [6].

In this paper we study the box-constrained quadratic program, with a view towards data parallel computing on massively parallel architectures. This paradigm of parallel computation uses a large number of processing elements — potentially millions of them — to operate on multiple data elements of the problem concurrently. It has been shown that data parallel computing can be used to efficiently execute row action algorithms for optimization problems with network structures; see Zenios [24] and references therein.

Our objective is to develop massively parallel designs for sparse and unstructured BQP problems. It is shown that gradient projection conjugate gradient (GPCG) type algorithms, like those of Dembo and Tulowitzki [7] and Moré and Toraldo [19] can be mapped very effectively onto a data parallel single instruction multiple data (SIMD) architecture. Alternative implementations on the Connection Machine CM–2 are evaluated empirically by solving the obstacle problem of Ciarlet [5] as given in Dembo and Tulowitzki [7]. An implementation for sparse, unstructured problems on the CM–2 is competitive with structured implementations of interior point algorithms on the IBM 3090-600 and parallel SOR methods on an Intel iPSC/860. Further specializing the implementation the structure of the obstacle problem results in a code that outperforms significantly the competing algorithms/computer architectures.

GPCG has been shown, by the earlier studies, to be a robust and efficient algorithm for solving BQP. For the suite of test problems solved, Moré and Toraldo [18] report:

> "Algorithm GPCG satisfies the convergence criteria in a few iterations (less than 15), and with a reasonable number of function-gradient evaluations and Hessian-vector products per iteration."

Our paper takes this line of research a step further by showing that GPCG can be implemented in such a way that its steps can be executed at a very high computing rate (measured in flops) on a massively parallel computer, and in a way that scales very well with an increasing number of processors. The implementation compares favorably with the two benchmark implementations of Han, Pardalos, and Ye [11] and De Leone [15].

Section 2 gives an overview of the GPCG algorithm and develops the data parallel implementation. Readers may first wish to consult Appendix A that introduces concepts of data parallel computing before they study §§2.2–2.3. Results of the computational experiments are reported in §3 and concluding remarks are given in §4.

2. The quadratic programming algorithm for box-constrained problems. The GPCG algorithm proceeds as follows.

Phase 1. Execute gradient projection (GP) iterates until a candidate active set is identified or the objective value is decreased sufficiently.

Phase 2. Execute conjugate gradient (CG) iterates to solve the locally unconstrained problem on the candidate active set identified in Phase 1. The conjugate gradient algorithm need only solve the unconstrained problem approximately.

During both phases of the algorithm we employ a projected linesearch. This has the advantage that the active set can be modified by adding or dropping from it more than one constraint at a time. Both phases of the algorithm and the projected linesearch are described in detail below.

2.1. Gradient projection step. The GP algorithm starts from some iterate $y^0 = x^k$ with the active set denoted by $\mathcal{A}(y^0)$. We use $\mathcal{A}(y)$ to denote the set of indices of the active constraint variables, i.e.,

$$\mathcal{A}(y) = \{i \mid y_i = \ell_i \text{ or } y_i = u_i,\ i = 1, 2, \ldots, m\}.$$

The algorithm then generates a sequence $\{y^j\}$ until a point y^J is found that satisfies

$$\mathcal{A}(y^J) = \mathcal{A}(y^{J-1}) \text{ or} \tag{3}$$

$$q(y^{J-1}) - q(y^J) \leq \eta \cdot \max\{q(y^{j-1}) - q(y^j) \quad | \quad 1 \leq j \leq J\}. \tag{4}$$

$\eta < 1$ is a user-specified parameter that determines the termination criteria for the gradient projection algorithm. Once the y^J that satisfies (3) or (4) is identified, the algorithm proceeds to use conjugate gradient to explore the face of the polytope defined by $\mathcal{A}(y^J)$.

The gradient projection iterative step is defined by

$$y^{j+1} = p_\Omega[y^j - \alpha_j \nabla q(y^j)], \tag{5}$$

where $y^0 = x^k$. Here, $p_\Omega[z]$ is the projection of a point z into the feasible set Ω:

$$p_\Omega[z] = \max\{\ell, \min\{u, z\}\}. \tag{6}$$

The max and min in the projection operator are taken elementwise. The parameter $\alpha_j > 0$ is computed using the linesearch of §2.3 such that $q(y^{j+1}) < q(y^j)$.

2.2. Conjugate gradient step. Now let $x^k = y^J$ and $\mathcal{A}(x^k)$ be the current active set. The following problem is locally unconstrained in the *free* variables, i.e., variables that do not belong to the set $\mathcal{A}(x^k)$:

$$\text{Minimize} \quad \{q(x^k + d) \mid d_i = 0 \ \text{ for all } i \in \mathcal{A}(x^k)\}. \tag{7}$$

If $\mathcal{A}(x^k) = \mathcal{A}(x^*)$, i.e., if the optimal solution to BQP lies in the same face as x^k, then the solution of (7) solves the original quadratic program.

Problem (7) can be expressed in terms of the free variables, denoted by w. We assume that there are m_k free variables; that is, the cardinality of $\mathcal{A}(x^k)$ is $(n - m_k)$. Let H_k be the $m_k \times m_k$ matrix obtained from H by removing rows and columns $i \in \mathcal{A}(x^k)$ and let $g_k = (c + Hx^k)_k \in \mathbf{R}^{m_k}$ be the subvector of the gradient of $q(x^k)$ obtained by removing components of the gradient vector such that $i \in \mathcal{A}(x^k)$. (H_k and g_k are called the *reduced Hessian* and *reduced gradient*, respectively.) The unconstrained minimization problem on the current active set is

$$\text{Minimize} \quad q_k(w) = \frac{1}{2} w^\top H_k w + g_k^\top w. \tag{8}$$

The CG algorithm starts with some $w^0 \in \mathbf{R}^{m_k}$ and generates a sequence of conjugate vectors $w^0, w^1, \ldots$ until an iterate w^J, $J \leq m_k$, is produced that satisfies

$$q_k(w^{J-1}) - q_k(w^J) \leq \epsilon \cdot \max\{q_k(w^{j-1}) - q_k(w^j) \ \mid \ 1 \leq \ell \leq J\}. \tag{9}$$

This test detects whether the CG algorithm is making sufficient progress. ϵ is a user-specified termination parameter. The inverse of the diagonal of the reduced Hessian, $\operatorname{diag}(H_k)$, is used as a preconditioner in the CG algorithm.

Upon termination of CG we obtain the descent direction d as follows: Set $d_i = 0$ if $i \in \mathcal{A}(x^k)$ and set d_i equal to the element of w^J that corresponds to the ith free variable. The algorithm now takes a step:

$$x^{k+1} = p_\Omega[x^k + \alpha_k d]. \tag{10}$$

Again $\alpha_k > 0$ is a steplength parameter computed using the projected linesearch of §2.3 such that $q(x^{k+1}) < q(x^k)$.

At this point, the algorithm may execute a gradient projection step, starting from x^{k+1} in order to identify a new active set. However, if x^{k+1} appears to be in the optimal face, then we repeat the conjugate gradient algorithm starting from x^{k+1}. That is, if the current active set is such that for all $i \in \mathcal{A}(x^{k+1})$ either $(\nabla q(x^{k+1}))_i \geq 0$ and $x_i = \ell_i$ or $(\nabla q(x^{k+1}))_i \leq 0$ and $x_i = u_i$, then the conjugate gradient is restarted. If for some i with $x_i = \ell_i$ we have $(\nabla q(x^{k+1}))_i < 0$, then the ith active variable could be released from the lower bound. Similarly, variables that satisfy $x_i = u_i$ and $(\nabla q(x^{k+1}))_i > 0$ can be released from the upper bound. In either case, the active set would change and the algorithm continues with gradient projection steps.

2.3. Projected linesearch. Both gradient projection and the conjugate gradient algorithm produce a descent direction d. In the former case d is the negative gradient, in the latter case d is obtained from the conjugate direction that satisfies (9). Both algorithms need to determine a step $\alpha_k > 0$ such that $\phi_k(\alpha) = q(p_\Omega[x^k + \alpha d])$ is sufficiently reduced. Sufficient reduction is achieved when the Armijo condition is satisfied, Luenberger [16], i.e.,

$$\phi_k(\alpha) \leq \phi_k(0) + \mu \phi_k'(0)\alpha, \qquad \mu \in (0, 1/2). \tag{11}$$

Under this condition, ϕ_k will decrease if $\phi_k'(0) < 0$.

We describe here the linesearch used to compute an acceptable α_k. The development follows Moré and Toraldo [19], which is based on the projected linesearch of Dembo and Tulowitzki [7]. The linesearch algorithm will produce a sufficient decrease in $\phi_k(\alpha)$ while identifying the set of constraints that will become active (possibly more than one). It is possible that no constraints will become active, in which case the linesearch just takes a single Newton step.

First, we identify the minimum step length $\tilde{\beta}$ beyond which one or more constraints become active. This is computed as

$$\tilde{\beta} = \min_j \left\{ \frac{u_j - x_j^k}{d_j} \text{ if } d_j > 0, \ \frac{\ell_j - x_j^k}{d_j} \text{ if } d_j < 0 \right\}. \tag{12}$$

(The stepsizes for which some constraint becomes active are termed *breakpoints*. $\tilde{\beta}$ is the smallest breakpoint.)

The first trial point of the linesearch is computed as the Newton step

$$\alpha_k^0 = -\frac{\phi_k'(0)}{\phi_k''(0)}. \tag{13}$$

It can be easily verified that $\alpha_k^0 = 1$ for the conjugate gradient step, and

$$\alpha_k^0 = \frac{\|r_k\|^2}{r_k^\top H_k r_k}$$

for the projected gradient step. Here r_k denotes the projection onto Ω of the gradient vector $\nabla q(x^k)$, given by

$$(r_k)_i = \begin{cases} (\nabla q(x))_i & \text{if } \ell_i < x_i < u_i, \\ \min\{(\nabla q(x))_i, 0\} & \text{if } x_i = \ell_i, \\ \max\{(\nabla q(x))_i, 0\} & \text{if } x_i = u_i. \end{cases} \tag{14}$$

If $\alpha_k^0 \leq \tilde{\beta}$, then the linesearch algorithm terminates. The point $x^k + \alpha_k^0 d$ produces a sufficient decrease in $\phi_k(\alpha)$ and does not change the active set. If $\alpha_k^0 > \tilde{\beta}$ the sufficient decrease condition (11) is violated, and the linesearch proceeds as follows.

THE PROJECTED LINESEARCH ALGORITHM:

WHILE $\phi_k(\alpha_k^{\ell+1}) > \phi_k(0) + \mu\phi_k'(0)\alpha_k^{\ell+1}$ DO:

Step I. Quadratic interpolation of $\phi_k(0), \phi_k'(0), \phi_k(\alpha_k^\ell)$ to get the minimizer α_k^* as follows:

For the conjugate gradient algorithm

$$\alpha_k^* = -\frac{(\alpha_k^\ell)^2}{2} \cdot \frac{r_k w^J}{\phi_k(\alpha_k^\ell), -\phi_k(0) - \alpha_k^\ell r_k w^J},$$

where r_k is the reduced gradient.

For the projected gradient algorithm

$$\alpha_k^* = \frac{(\alpha_k^\ell)^2}{2} \cdot \frac{\|r_k\|^2}{\phi_k(\alpha_k^\ell) - \phi_k(0) - \alpha_k^\ell \|r_k\|^2},$$

where r_k is the projected gradient.

Step II. Backtracking:

$$\alpha_k^{\ell+1} = \max\left\{\tilde{\beta}, \max\left\{\frac{\alpha_k^\ell}{100}, \min\left\{\alpha_k^*, \frac{\alpha_k^\ell}{2}\right\}\right\}\right\}.$$

END DO.

Step I performs a standard quadratic interpolation of the piecewise quadratic function ϕ_k and analytically computes its minimizer a_k^*. If the minimizer is smaller than $\tilde{\beta}$ then the new iterate $(x^k+\alpha_k^*)$ does not change the active set, and the projection $p_\Omega[x^k+\alpha_k^* d]$ does not effect any changes to the projected point. The quadratic interpolation is a good estimate and the calculated minimizer is acceptable. Otherwise, the projection operator is effective and the "effective" direction given by $(p_\Omega[x^k+\alpha_k^* d]-x^k)$ may not be sufficiently steep. If this is the case the algorithm backtracks from the estimated step α_k^* to a smaller value towards $\tilde{\beta}$ as suggested in Step II, until an "effective" direction is obtained which is sufficiently steep. At the extreme case the algorithm backtracks all the way to $\tilde{\beta}$, in which case the "effective" direction is exactly the direction obtained by the conjugate gradient or projected gradient algorithm. In this case only one active constraint will change. The backtracking linesearch compromises between finding a new iterate with large decrease in the objective against finding a new iterate that significantly changes the active set. Several investigations (Bertsekas [2], Dembo and Tulowitzki [7], and Ahlfeld et al. [1]) have shown backtracking linesearches, such as the one described here, to be very efficient in practice.

3. Data parallel designs. We discuss now data parallel implementations of GPCG on the Connection Machine CM–2. Alternative designs are considered for problems when the matrix H is (i) sparse without any detectable structure; (ii) uniformly sparse; and (iii) structured, as obtained from the obstacle problem. The implementations are designed in such a way as to facilitate efficient utilization of the processing elements during all phases of the algorithm: the projected linesearch, computation of projected gradients, and the conjugate gradient solver. At this point, readers who are not familiar with concepts of data level parallelism and the CM–2 should consult the Appendix for a description of virtual processing (VP) sets and geometries.

The GP algorithm and the linesearch operate on the full problem of size $n \times n$. The conjugate gradient (CG) algorithm operates on the reduced problem of size $m_k \times m_k$. In either case, we need to operate on the coefficients of the Hessian matrix (either the full or the reduced matrix), as well as the current iterate x^k.

The most general mapping of the BQP to the CM processors requires four VP sets. One pair of VP sets holds the full matrix H of size $n \times n$ and a full vector of size n. We denote these sets by H and v, respectively. The second pair holds the reduced matrix H_k of size $m_k \times m_k$ and a reduced vector of size m_k. We denote these sets by RH and Rv, respectively. All VP sets are set up as one-dimensional geometries.

Both H and H_k are stored row-wise. The nonzero entries of each row are stored in a contiguous set of processors that are designated as a *segment* (see Appendix A). This representation is akin to the data structures used in the row-wise representation of sparse matrices (Blelloch [4]). One significant difference, however, is that even if the matrix is symmetric, we assign processors to both the upper and lower triangular elements. This representation deviates from current sparse-matrix practices, but utilizes more efficiently the CM processors.

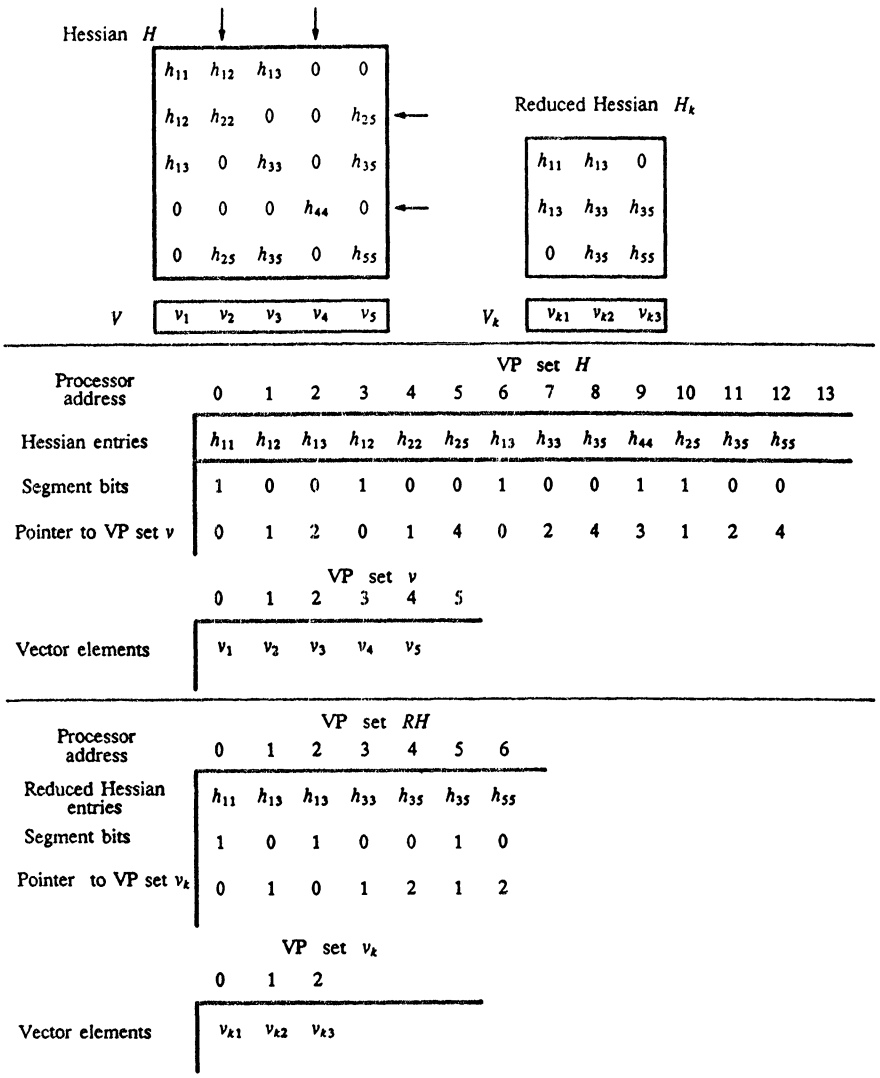


FIG. 1. *Data parallel representation of the sparse Hessian (H) and reduced Hessian (H_k) on the Connection Machine.*

Each processor of the H (or RH) set holds the numerical value for an element, h_{ij}, of the corresponding matrix and its column number, j. The column number is actually a pointer to the processor in the v (or Rv) set that holds the corresponding v_j. Thus the appropriate vector elements can be fetched by the matrix elements for the execution of matrix-vector multiplications. Figure 1 illustrates the representation of H, H_k, v, and v_k.

We discuss parallel implementations of the linesearch and CG algorithms. The GP algorithm merely requires the calculation of the gradient vector and its projection (cf. (5)). These calculations are also used in the linesearch procedure so the data parallel implementation of GP is not treated separately.

3.1. Data parallel projected linesearch. We consider now the data parallel implementation of the projected linesearch using the virtual processor configuration

illustrated in Fig. 1. The implementation is efficient since at every step of the linesearch algorithm the required calculations can be performed simultaneously by multiple processors. If there are as many virtual processors as there are variables then calculations of the algorithm, such as the projection operator or the evaluation of the reduced gradient norm, can be executed in just one step.

Consider, for example, the projected linesearch for the GP step. Each processor in the VP set v corresponds to a variable x_j. It has local memory fields to hold the bounds u_j and ℓ_j, the search direction d_j, the cost coefficients c_j, and the current iterate x_j^k. The linesearch algorithm (see §2.3) first needs to evaluate the breakpoints and identify the smallest one, $\tilde{\beta}$. The breakpoint β_j for the jth variable is calculated by the jth virtual processor of the set v. A global-min operation identifies $\tilde{\beta}$, which is then broadcast to all processors. The reduced gradient r_k — needed to calculate a_k^0 — is evaluated by applying the projection operator (14) componentwise at each processor. A local multiplication of $(r_k)_j$ by itself — executed by the jth virtual processor for all j simultaneously — and a scan-with-add calculation, computes the norm $\|r_k\|^2$.

The only remaining nontrivial step of the linesearch is the evaluation of $\phi_k(\alpha_k^\ell)$ at each trial point α_k^ℓ. The projection $y \leftarrow p_\Omega[x^k + \alpha_k^{\ell+1} d]$ is again computed componentwise by multiple processing elements. Once the vector y is computed, we need to compute $\phi_k(\alpha_k^\ell) = q(y) = \frac{1}{2} y^\top H y + c^\top y$. The inner product $c^\top y$ is computed on the vector VP set v using local multiplication $c_j \cdot y_j$ followed by a global-add operation. The product Hy requires the communication of data from the vector VP set v where elements of y are stored to the matrix VP set H. Recall that each VP, h_{ij}, of H has a pointer to the VP, v_j, of v corresponding to its column number. Hence a *get* operation will move vector elements y_j to the VP of H corresponding to entries of the column j of matrix H. A local multiplication followed by a segmented-spread-add operation will then complete the matrix vector product, Hy. The results are stored in set v for later use by the first VP in each segment of the H set — which corresponds to the diagonal entries h_{jj} — using a send operation to the jth VP of the v set. A local multiplication with the local copy of y_j followed by a global-add completes the computation $y^\top H y$.

3.2. Data parallel conjugate gradient solvers. We consider now three alternative designs for implementing the conjugate gradient algorithm. This is the most time-consuming part of GPCG. We need efficient ways to execute matrix-vector products $H_k v_k$ on the reduced space and inner products $v_k^\top v_k$. The vector v_k is dense and is treated as such in all designs. The reduced Hessian matrix H_k is sparse. The preconditioner used in the CG algorithm is the inverse of the diagonal of the reduced Hessian matrix, which is a dense vector. The implementation of the preconditioner does not pose any difficulties.

The first design (§3.2.1) assumes that H_k has no detectable structure and carries out the implementation at a high level using virtual processing. The second design (§3.2.2) makes the assumption that the number of nonzero elements in H_k is uniform across rows. The implementation is carried out using the hypercube model of the CM–2 more explicitly. Finally (§3.2.3), we exploit the special structure of the Hessian matrix arising from the obstacle model. The algorithm is implemented using a two-dimensional nearest-neighbor communication grid. Alternative implementations of a CG algorithm for finite element optimization on an SIMD machine are reported in Dixon and Ducksbury [8].

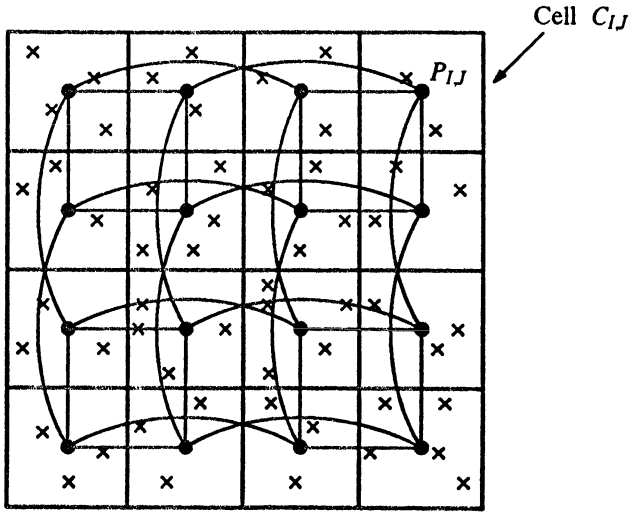


FIG. 2. *Overlay of a sparse matrix on the Connection Machine processors: each processor holds (roughly) the same number of nonzero elements.*

3.2.1. General matrices. This design uses the sparse representation for H_k illustrated in Fig. 1. The computation of matrix-vector products of the form $H_k v_k$ is identical to the computation of Hv as already explained in the context of the linesearch algorithm. The only difference from the description of §3.2 is that we work on the reduced VP sets RH and Rv, instead of the full matrix set H and the full vector v. Inner vector products are similarly computed on the set Rv.

3.2.2. Uniformly sparse matrices. In this section we discuss our approach for matrix-vector multiplication when the matrix is *uniformly sparse* (i.e., all rows have roughly the same number of nonzero elements, as illustrated by the x in Fig. 2). For this implementation we use the *slicewise* programming model of the CM–2 (see Appendix A). Thus, for the purpose of this section, a "processor" is the ensemble of a floating point unit and the 32 local memories of the corresponding single bit processors.

Let the number of processors $P = 2^d$ be an even power of two, so that $\sqrt{P} = 2^{d/2}$ is an integer power of two. To store the matrix, we treat the machine's processors as a $2^{d/2} \times 2^{d/2}$ grid, where each processor $P_{I,J}$ of the grid resides at location $2^{d/2}I + J$ of the hypercube. Note that, in this configuration, row I of the grid consists of the processors $2^{d/2}I + J$ where J varies from 0 through $2^{d/2} - 1$; i.e., row I occupies a $d/2$-dimensional subcube of the hypercube. Similarly, a column of the grid also occupies a subcube.

Let the full vector be of size n, and let the sparse matrix hold n_c nonzero coefficients. If we overlay the $2^{d/2} \times 2^{d/2}$ grid over the sparsity pattern (as in Fig. 2), then each grid cell $C_{I,J}$ "captures" roughly n_c/P coefficients. For our matrix-vector multiplication we let each processor $P_{I,J}$ hold the coefficients that are captured by

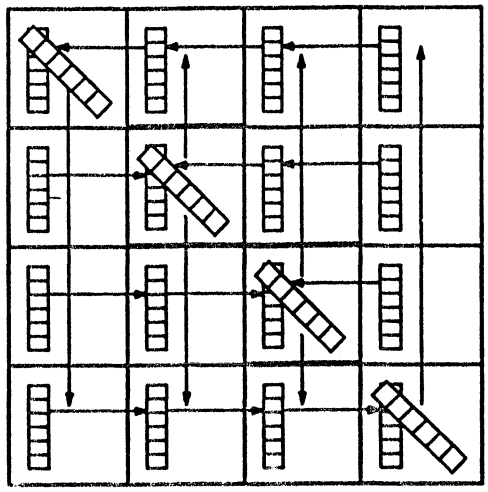


FIG. 3. *Spreading a vector along the diagonal of a matrix to compute the matrix-vector product, and spreading partial results back to the diagonal to accumulate the result vector.*

cell $C_{I,J}$.[1] Let the matrix row and column coordinates vary from 0 through $n-1$, and let the processors' grid coordinates vary from 0 through $2^{d/2}-1$. Processor $P_{I,J}$ holds the nonzero coefficients $m_{i,j}$ such that

$$\lceil n/2^{d/2} \rceil I \le i < \lceil n/2^{d/2} \rceil (I+1) \quad \text{and} \quad \lceil n/2^{d/2} \rceil J \le j < \lceil n/2^{d/2} \rceil (J+1).$$

Within each processor, we sort the coefficients in row major order and place them contiguously in memory (as in Fig. 1).

The ordering of the coefficients is similar to the ordering used in the matrix-vector multiplication procedure for unstructured matrices. Of course there is an important difference: In the context of the uniformly sparse matrix-vector multiplier, the ordering describes how the coefficients for cell $C_{I,J}$ are laid out within the memory of a floating point node, processor $P_{I,J}$. In the context of the router-based matrix-vector multiplier for unstructured matrices, the ordering defines how the coefficients are laid out across the bit serial processors.

A vector v involved in the uniform sparse matrix-vector multiplication is stored in the diagonal processors of the grid (as shown in Fig. 3). Each diagonal processor $P_{J,J}$ holds the vector elements $v_{\lceil n/2^{d/2} \rceil J}, \ldots, v_{\lceil n/2^{d/2} \rceil (J+1)-1}$. Let v_J denote the portion of the vector v stored in processor $P_{J,J}$.

We can now describe the matrix-vector multiplication. We want to perform the sums

$$v'_i = \sum_{j=0}^{n-1} h_{ij} v_j \text{ for } i = 0, \ldots, n-1,$$

[1] When the matrix is not uniformly sparse we may still be able to roughly capture n_c/P coefficients per cell $C_{I,J}$ and therefore process the same number of nonzero coefficients per processor $P_{I,J}$. This can be achieved by randomly permuting the rows and columns of the matrix. If dense rows/columns are grouped together with very sparse rows/columns it is possible that each cell $C_{I,J}$ will hold roughly the same number of nonzero coefficients.

and we want each portion

$$v'_I = (v'_{\lceil n/2^{d/2}\rceil I}, \ldots, v_{\lceil n/2^{d/2}\rceil (I+1)-1})$$

of the result vector to be placed in the diagonal processor $P_{I,I}$.

For the first step of the matrix-vector multiplication, each diagonal processor $P_{J,J}$ broadcasts subvector v_J vertically to the processors $P_{0,J}, \ldots, P_{2^{d/2}-1,J}$ of grid column J (as shown by the vertical arrows in Fig. 3). More precisely, the processors of grid column J act in concert to distribute the elements of subvector v_J throughout column J. The processors of grid column J occupy a $d/2$-dimensional subcube of the hypercube. Therefore the processors in column J can use a *one-to-all broadcast* algorithm (Lennart and Ho [14]) to distribute the elements of subvector v_J among themselves in $O(\lceil n/2^{d/2}\rceil + d/2) = O(\lceil n/\sqrt{P}\rceil + \log(P))$ time. After the broadcast operation, each processor $P_{I,J}$ holds the coefficients and vector elements that are required for the matrix-vector multiplication in cell $C_{I,J}$.

In the second step of the matrix-vector multiplication, each processor $P_{I,J}$ calculates locally the partial sums

$$v'_{i,J} = \sum_{j=\lceil n/2^{d/2}\rceil J}^{\lceil n/2^{d/2}\rceil (J+1)-1} h_{ij} v_j \quad \text{for } i = \lceil n/2^{d/2}\rceil I, \ldots, \lceil n/2^{d/2}\rceil (I+1) - 1.$$

The partial sums are calculated in roughly $O(n_c/P)$ time. Let $v'_{I,J}$ denote the resulting vector $(v'_{\lceil n/2^{d/2}\rceil I,J}, \ldots, v'_{\lceil n/2^{d/2}\rceil (I+1)-1,J})$ of partial sums in processor $P_{I,J}$. In Fig. 3, each vector $v'_{I,J}$ of partial sums is shown as an array in cell $C_{I,J}$.

Now consider an element v'_i of the desired result vector v'. Observe that

$$\begin{aligned} v'_i &= \sum_{j=0}^{n-1} h_{ij} v_j \\ &= \sum_{J=0}^{2^{d/2}-1} \sum_{j=\lceil n/2^{d/2}\rceil J}^{\lceil n/2^{d/2}\rceil (J+1)-1} h_{ij} v_j \\ &= \sum_{J=0}^{2^{d/2}-1} v'_{i,J}. \end{aligned}$$

The equation generalizes to portions v'_I of the solution vector:

$$v'_I = \sum_{J=0}^{2^{d/2}-1} v'_{I,J}.$$

For the third and last step of the matrix-vector multiplication, we implement the latter equation by summing the subvectors $v'_{I,J}$ across each grid row I (as shown by the horizontal arrows in Fig. 3). Each row I places the result subvector v'_I in processor $P_{I,I}$. The processors of grid row I occupy a $d/2$-dimensional subcube of the hypercube. Therefore the processors in row I can use either the algorithm described in McKenna and Zenios [17] or a time-reversed variant of the one-to-all broadcast in Lennart and Ho [14] to sum the subvectors in $O(\lceil n/2^{d/2}\rceil + d/2) = O(\lceil n/\sqrt{P}\rceil + \log(P))$ time.

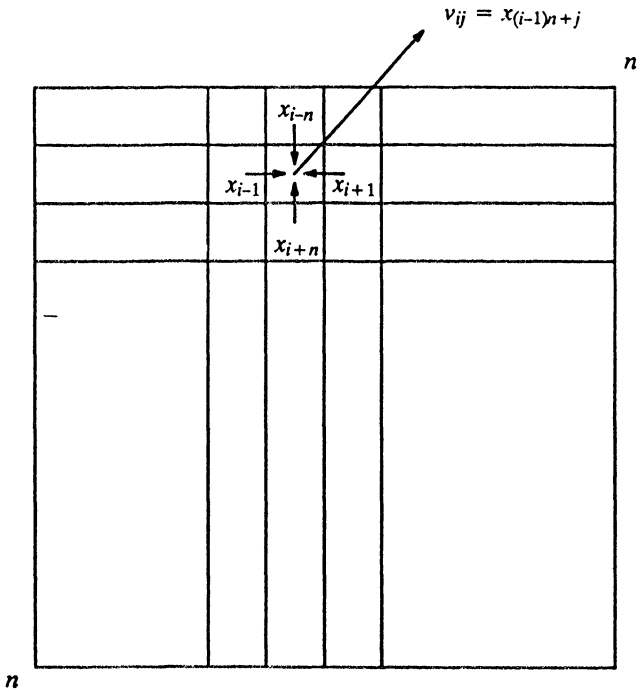


FIG. 4. *Computing Hx for the obstacle problem.*

This third step completes the matrix-vector multiplication. The three steps of the multiplication require a total of $O(n_c/P+\lceil n/\sqrt{P}\rceil+\log(P))$ time. If $n_c = \Omega(P\log(P))$ (i.e., the number of coefficients is asymptotically at least a multiple of $P\log(P)$), then n_c/P dominates $\log(P)$, so that the $\log(P)$ term drops out of the execution time. If $n_c/n = \Omega(\sqrt{P})$ (i.e., the number of coefficients per matrix row is asymptotically at least a multiple of $\sqrt{P}$), then n_c/P dominates $\lceil n/\sqrt{P}\rceil$, so that the $\lceil n/\sqrt{P}\rceil$ term also drops out. In this final case, the execution time is $O(n_c/P)$. This execution time is optimal in that the best possible execution time for a single processor is divided over the P parallel processors.

3.2.3. Structured matrices: the obstacle problem. We consider now the implementation of the algorithm for the solution of the *obstacle problem* of Ciarlet [5]. The H matrix of the obstacle problem is obtained from a difference approximation to the Laplacian operator of some potential function defined over a two-dimensional grid of size $n \times n$. The grid is subdivided into *pixels*. Within each pixel with coordinates (i,j) we have a uniform potential $v_{i,j}$ that needs to be estimated by solving the quadratic programming problem. (Boundary conditions are specified by the presence of obstacles on this grid, and those give rise to the upper and lower bounds of the quadratic program.) Using a lexicographic ordering of the variables we get the vector $x \in \mathbf{R}^{n^2}$ such that $x_\ell = v_{i,j}$ for $\ell = (i-1)n + j$. The sparsity pattern of H is determined by the interaction of x_ℓ with the elements of vector x in all adjacent pixels. Hence, the matrix is pentadiagonal, i.e., it has five bands, with nonzero entries along the diagonal and two nonzero entries above and two nonzero entries below the diagonal. Further details on the structure of the matrix are given in §4.1.

The product Hx can be computed using the two-dimensional grid of size $n \times n$ without the need to construct the matrix H which is of dimension $n^2 \times n^2$ (see Fig. 4). The result vector $y = Hx$, of dimension n^2 is stored on the two-dimensional grid, such that y_ℓ is stored at (i,j), for $\ell = (i-1)n + j$. The result vector is computed by $y_\ell = 4v_{i,j} - v_{i-1,j} - v_{i+1,j} - v_{i,j-1} - v_{i,j+1}$. This is a standard calculation for pentadiagonal matrices, referred to as a five-point stencil. The vector components $v_{i-1,j}, v_{i+1,j}, v_{i,j-1}, v_{i,j+1}$ are fetched first to virtual processor (i,j) using nearest neighbor grid communications (NEWS). The vector y can be computed using four NEWS communications and five floating point operations. Special routines on the CM–2 take advantage of the full communication bandwidth of the hypercube, and were used to fetch the data.

The inner product of two vectors $x^\top x$ can also be computed directly on the two-dimensional grid. Each processor computes the product $x_\ell x_\ell$ and a global-add operation completes the inner product.

4. Computational results. The GPCG algorithm was implemented on a Connection Machine CM–2. The implementations for the unstructured sparse and uniformly sparse, matrices were carried out using PARIS release 6.1 in double precision arithmetic (52 bits in the mantissa). The implementation for the structured matrices arising from the obstacle problem was done in CM Fortran. For all three implementations the algorithm was executed using a SUN front-end and a CM–2 with 64-bit floating point accelerators and 32K byte local memories. Operations on the CM–2 accounted for 83–98% of the total execution time. The number of processing elements used depends on the implementation and the problem size, and is reported together with the computational results. The various parameters of the algorithm were set as follows:

Termination parameter for projected gradient (cf. (4)) $\eta = 0.25$.
Termination parameter for conjugate gradient (cf. (9)) $\epsilon = 0.1$.
Sufficient decrease parameter for the linesearch (cf. (11)) $\mu = 0.1$.
The GPCG algorithm terminates when $\|P_\Omega \nabla q(x)\| \leq 10^{-5} \|\nabla q(x^0)\|$, where x^0 is the starting point.

The performance of the GPCG algorithm for different problem characteristics has already been studied extensively in Moré and Toraldo [19]. In general we found that our implementation behaves (almost) identically to the results reported in Moré and Toraldo with respect to number of steps, number of function/gradient evaluations, and number of matrix-vector products. Hence, we start with an implementation that is as efficient as the algorithm implemented serially in previous studies. The objective of our experiments is to establish the efficiency of the parallel implementations as measured in solution times. The test problems are identical to those used in Han, Pardalos, and Ye [11]. Hence, we can compare the massively parallel implementation of the GPCG algorithm with the vector implementation of interior point algorithms on the IBM 3090-600S vector supercomputer. We also compare our results with those obtained by DeLeone [15] using parallel SOR on an Intel iPSC/860.

The first question we want to address with our experiments is the efficiency of the alternative parallel implementations. This question is addressed by examining two performance metrics: computing the rate with which calculations are executed, i.e., the flop rate of the algorithm, and examining the change in solution time when solving bigger problems on proportionally larger machines. The second question we address is: How does the massively parallel implementation of GPCG compare with implementations of competing algorithms on alternative parallel architectures and

vector supercomputers? This question is addressed by comparing the solution time of our code with benchmark results obtained using other state-of-the-art algorithms on other computer platforms, namely, a vector supercomputer and a distributed memory hypercube.

4.1. Test problems. We discuss now the characteristics of the test problems used in the computational experiments. The code was tested on both randomly generated problems, and on problems derived from an engineering application.

Dense problems. Dense test problems are generated according to the problem generator of Moré and Toraldo [19]. The matrix H is defined by

$$H = QBQ, \text{ where } Q = I - \frac{2}{\mid v \mid^2} vv^\top. \tag{15}$$

The elements of $v \in \mathbf{R}^n$ are randomly generated in the interval $(-1, 1)$. I is the identity matrix and B is defined as $\text{diag}(b)$, where the diagonal elements are given by

$$\log b_i = \left(\frac{i-1}{n-1}\right) \text{cond}\ , \ i = 1, 2, \ldots, n,$$

where the parameter cond specifies the condition number of H.

The number of active constraints at the optimal solution $\text{na}(x^*)$ is also a user-specified parameter. We first determine the optimal active set by choosing $\text{na}(x^*)/n$ indices of variables that belong to the optimal active constraint set $\mathcal{A}(x^*)$. Then we determine the amount of degeneracy by specifying the magnitude of the gradient vector components at the optimal solution. Let $\mid d_i \mid = 10^{-t(i)\text{deg}}$ for $i \in \mathcal{A}(x^*)$, where $t(i)$ are randomly generated numbers in (0,1) and deg is a user-specified parameter. The sign of the gradient vector components is also chosen at random, hence half of the active variables have positive gradient at the solution, and the other half have negative. We now choose the optimal point x^* based on the active set $\mathcal{A}(x^*)$ and the signs of d_i:

If $i \neq \mathcal{A}(x^*)$ choose $x_i^* \in (0, 1)$ and set $d_i = 0$.
Else if $d_i > 0$, $x_i^* = 0$.
Else $x_i^* = 1$.

Finally we set $c = -Hx^* + d$. We can easily verify that x^* is the optimal solution of a BQP with input data H and c and with the prescribed characteristics.

The dense test problems for our experiments were generated with the following input parameters: cond = 6, deg = 6, $\text{na}(x^*) = n/2$. These parameters are identical to those used by Han, Pardalos, and Ye [11] to generate test problems of size up to 800×800.

Sparse problems. Sparse test problems were generated using the obstacle problem of Ciarlet [5] as explained by Dembo and Tulowitzki [7], and were subsequently used by Moré and Toraldo [19] and Han, Pardalos, and Ye [11]. The matrix H for the obstacle problem is pentadiagonal. Entries on the diagonal are equal to 4. The off-diagonal entries $h_{i,i+1}, h_{i,i-1}, h_{i,i+n}, h_{i,i-n}$ are all equal to -1, and remaining entries are 0. The linear component of the cost function c is given by

$$c = -\left(\frac{1}{m+1}\right)^2, \text{ where } m = \sqrt{n}.$$

TABLE 1
Performance of the data parallel implementations. The flops rates for the 64K CM–2 *are extrapolated.*

Routine	Mflops	
	8K CM–2	64K CM–2 (peak)
Matrix-vector (unstructured matrices)	4.94	39.5
Matrix-vector (uniformly sparse matrices)	113.8	910.4
Matrix-vector (structured matrices)	207	1656
Conjugate gradient (unstructured matrices)	10.6	84.8
Conjugate gradient (uniformly sparse matrices)	104.2	833.6
Conjugate gradient (structured matrices)	264	2112

We generated two sets of obstacle problems. For Problem I we generate the bounds as follows:

$$l_i = (\sin(9.2\alpha_i) \times \sin(9.3\gamma_i))^3, \tag{16}$$

$$u_i = (\sin(9.2\alpha_i) \times \sin(9.3\gamma_i))^2 + 0.02. \tag{17}$$

For Problem II the bounds are generated by

$$l_i = (\sin(3.2\alpha_i) \times \sin(3.3\gamma_i)), \tag{18}$$

$$u_i = 2000. \tag{19}$$

The coefficients for both problems are given by

$$\alpha_i = \left(i - \lfloor \frac{i-1}{m} \rfloor \times m\right) \times (1/m + 1)\,, \qquad i = 1, 2, \ldots, n.$$

$$\gamma_i = \lceil \frac{i}{m} \rceil \times (1/m + 1)\,, \qquad i = 1, 2, \ldots, n.$$

4.2. Efficiency of the parallel implementations. The implementation of the algorithm for structured matrices is the most efficient, followed by the implementation for uniformly sparse matrices and the implementation for unstructured matrices. All implementations achieve high flops rates (up to 2.1 Gflops) in the conjugate gradient solver. The flops rates for the matrix-vector multiplication and the conjugate gradient solver are summarized in Table 1.

To illustrate the *scalability* of the algorithm we solved a series of problems of different sizes on an 8K, 16K, and 32K CM–2 using the structured implementation. Table 2 gives the solution time for each problem size. We also report the number of PG steps. For the largest problem sizes, we observe that the solution time is reduced by a factor slightly greater than three when moving from an 8K system to a 32K system. We also note that the time per iteration increases very slowly when the size of the problem increases proportionally to the size of the machine. For example, the 10,000 variable problem takes 0.298 sec/iteration on the 8K CM–2. The 40,000 variable problem takes 0.320 sec/iteration on the 32K CM–2.

TABLE 2
Scalability of the algorithm with increasing number of processing elements. Execution times are in CM *seconds.*

Size n	PG itns.	8K CM–2	16K CM–2	32K CM–2
10000	8	2.39	2.18	1.99
40000	10	5.03	4.27	3.20
90000	11	11.12	7.15	4.81
115600	13	15.34	10.71	6.66
160000	13	22.97	15.11	8.89
250000	13	39.86	22.23	12.45
360000	15	59.87	36.36	19.47

4.3. Comparisons. In this section we compare the results of the massively parallel implementations with the interior point algorithm implemented by Han, Pardalos, and Ye [11] using vectorization on the IBM 3090-600S. Their implementation exploits the structure of the quadratic matrix by using optimized subroutines from the ESSL library. Hence, a fair comparison can be made with our implementation for structured matrices. In the following tables we summarize the results for the sparse, unstructured implementations as well. We also compare with the results reported by De Leone [15] using a parallel implementation of SOR on an Intel iPSC/860 hypercube.

In order to evaluate the performance of each implementation viz a viz the performance of the underlying hardware platform, we cite the performance characteristics of each machine. The CM–2 has a cycle time of (approx.) 140 nsec. Each node has a peak computing rate of 14 Mflops. The IBM 3090 with vector units has a clock cycle of (approx.) 15 nsec. Each processor has a peak computing rate of 140 Mflops. A single node of the Intel iPSC/860 has a peak computing rate of 40 Mflops. In benchmark testing, with the solution of a dense system of linear equations, the single-processor IBM 3090 achieved a computing rate of 16 Mflops and the iPSC/860 4.5 Mflops. Both benchmark results are for the solution of 100×100 systems using LINPACK, and without any tuning of the codes; Dongarra [9].

Table 3 compares results for the dense problems.[2] This is an unfair exercise for GPCG since the sparse implementations are specially tailored to sparse data. Much better performance could be obtained by a code written specifically for dense problems. We chose to include these timings because even the uniformly sparse code is seen to be very efficient, especially for the larger problem size ranges. The explanation for the high performance of the uniformly sparse code is simple: Since the matrix is dense it has the same number of nonzero entries per row, and this number is large. Hence, a large number of nonzero entries are packed and processed at each processor as explained in §3.2.2. The cost of communication is amortized as evidenced from the high flops rates reported in Table 1. The uniformly sparse code outperforms for this class of problems the dense code of Han, Pardalos, and Ye even when the latter is vectorized on an IBM 3090-600S vector supercomputer (Table 3).

Tables 4 and 5 compare results for the two obstacle problems when using PGCG on the 8K CM–2, SOR on an Intel iPSC/860, and the interior point algorithm on an IBM 3090-600S. Results are consistent across the two tables: The implementation of

[2] In order to eliminate noise in our experiments due to variations in the random number generators, we generated five instances of two of the test problems. The performance of the algorithm was identical for all five instances. No changes were observed in the number of PG steps and only minor changes in the number of CG steps. Solution times varied by less than 1%. The results in Table 3 are for the solution of a single problem instance.

TABLE 3
Solving dense problems with the massively parallel implementations of GPCG *on the* CM–2 *and the implementations of interior point algorithms on the* IBM 3090-600S. *Results reported in* CM *and* CPU *seconds respectively.* NEM = *not enough memory.* NA = *not available.*

Size n	Connection Machine CM–2 (8K)		IBM 3090	
	Unstructured	Uniformly sparse	Scalar	Vectorized
100	13.88	1.24	1.57	0.20
200	23.71	1.54	11.83	0.85
400	88.48	2.26	199.12	4.84
500	92.62	2.64	267.20	9.07
800	134.35	7.18	NA	34.46
1024	NEM	5.69	NA	NA

GPCG that exploits problem structure is one to two orders of magnitude faster than the two competing algorithms/computer architectures. (Both competing implementations exploit the problem structure as well.) Even more interesting is the observation that the unstructured implementation of GPCG is still faster than the algorithm on the IBM 3090, and is slower than the algorithm on the iPSC/860 only by a factor of 2–5. These results support the claim that, for the solution of BQP problems, the massively parallel Connection Machine CM–2 can outperform by a large factor the vector supercomputer and distributed memory hypercube.

Of course it is possible that even better performance could be achieved on the CM–2 with a data parallel implementation of either one of the other two algorithms. However, implementations of the GPCG on either one of the other two architectures would not be competitive with the CM–2 implementation: none of the other computer architectures could execute the algorithm with the flop rates achieved on the CM–2 (Table 1). For example, the peak computing rate of the IBM 3090-600S structured linear algebra subroutines is around 70 Mflops; compare this number to the 2.1 Gflops achieved with the structured solver on the full-size CM–2. Overall we observe that, without the results of this paper, the large scale problems would take from 15 minutes to more than an hour to be solved, while with our approach the same problems are solved within 1–2 minutes. We also solved a test problem with 2.89 million variables, which is an order of magnitude larger than any other problem reported in the literature. The solution time for this problem was under 10 minutes on the 32K CM–2.

The benchmark results reported in Tables 4 and 5 are meant to be indicative of the size of the problems one can solve using alternative combinations of algorithms/parallel architectures. Such benchmarks provide one performance metric to guide users in selecting an algorithm or a computer architecture for a specific application.

5. Conclusions. We have reported on alternative approaches for the implementation of a GPCG algorithm for BQP on data-level parallel computers. The implementations were shown to be very competitive with state-of-the-art implementations of interior point methods on vector supercomputers and SOR methods on distributed memory hypercubes. An implementation of the algorithm that takes advantage of the structure of the obstacle test problems achieves computing rates of 2.1 Gflops and is substantially more efficient than competing methodologies. It is also very interesting to note that even the implementation for sparse but unstructured problems remains quite competitive.

TABLE 4

Solving the obstacle Problem I *with the massively parallel implementations of* GPCG *on the* CM–2, *with the parallel* SOR *of De Leone on the* Intel iPSC/860 *and the vector implementation of interior point algorithms of Han, Pardalos, and Ye on the* IBM 3090-600S. *Results reported in* CM *and* CPU *seconds, respectively.* * = *Solved on* 32K CM–2. NEM = *not enough memory.* NA = *not available.*

Size n	Connection Machine CM–2 (8K)			Intel iPSC	IBM 3090
	Unstructured	Unif. sparse	Structured		
10000	12.76	15.82	2.39	10.5	16.3
40000	68.36	107.09	5.03	49.1	131.1
90000	286.63	NEM	11.12	112.8	437.6
115600	377.03	NEM	15.34	156.7	700.3
160000	490.20	NEM	22.97	245.9	1035.8
250000	NEM	NEM	39.86	518.9	2110.5
360000	NEM	NEM	59.87	1047.6	4090.3
2890000	NEM	NEM	*482.00	NA	NA

TABLE 5

Solving the obstacle Problem II *with the massively parallel implementations of* GPCG *on the* CM–2, *with the parallel* SOR *of De Leone on the* Intel iPSC/860 *and the vector implementation of interior point algorithms of Han, Pardalos, and Ye on the* IBM 3090-600S. *Results reported in* CM *and* CPU *seconds respectively.* NEM = *not enough memory.*

Size n	Connection Machine CM–2 (8K)			Intel	IBM 3090
	Unstructured	Unif. sparse	Structured		
10000	29.34	36.69	3.62	5.9	25.4
40000	157.67	223.51	8.72	53.4	203.9
90000	525.51	NEM	19.27	130.8	699.9
115600	1534.06	NEM	72.86	201.3	1018.7
160000	1082.26	NEM	40.31	394.9	1534.7
250000	NEM	NEM	73.05	911.8	3141.9
360000	NEM	NEM	110.55	1873.4	5312.4

Appendix A. The Connection Machine environment. We briefly introduce the characteristics of the Connection Machine CM–2 (Hillis [13]) that are relevant to our parallel implementations. See also [21]. The CM–2 is a fine-grain SIMD (i.e., single instruction stream, multiple data stream) system. Its basic hardware component is an integrated circuit with sixteen processing elements (PEs) and a *router* that handles general communication. A fully configured CM has 4,096 chips for a total of 65,536 PEs. The 4,096 chips are interconnected as a 12-dimensional hypercube. Each processor has 32K bytes of local memory, and for each cluster of 32 PEs a floating point accelerator handles floating point arithmetic. Thus, a CM–2 with $P = 2^d$ processors can be viewed as a $d - 5$ dimensional hypercube with 2^{d-5} floating point processors each with correspondingly larger local memories. This is often referred to as the *slicewise* version of the machine, since the data is stored in slices across the associated 32 local memories.

The CM–2 provides the mechanism of virtual processors VPs that allows one PE to operate in a serial fashion on multiple copies of data. VPs are specified by segmenting the local memory of each PE and allowing physical processors to operate serially and in lock step fashion on these segments. The number of segments is called the VP *ratio* (i.e., ratio of virtual to physical PEs). Looping by the PE over all the memory segments is executed in linear time. The set of virtual processors associated with each element of a data set is called a VP *set.*

The CM–2 supports two addressing mechanisms for communication. The *send* ad-

dress is used for general purpose communications via the routers. The NEWS address describes the position of a VP in an n-dimensional grid that optimizes communication performance. The *send* address indicates the location of the PE (hypercube address) that supports a specific VP and the relative address of the VP in the VP set that is currently active. The NEWS address is an n-tuple of coordinates that specifies the relative position of a VP in an n-dimensional Cartesian-grid geometry. A *geometry* is an abstract description of such an n-dimensional grid. Once a geometry is associated with the currently active VP set, a relative addressing mechanism is established among the processors in the VP set. Each processor has a relative position in the n-dimensional geometry and NEWS allows the communication across the north, east, west, and south neighbors of each processor, and enables the execution of operations along the axes of the geometry. Such operations are efficient since the n-dimensional geometry can be mapped onto the underlying hypercube in such a way that adjacent VPs are mapped onto vertices of the hypercube connected with a direct link. Parallel primitives can be invoked to execute operations along some axis of the geometry (using NEWS addresses), operate on an individual processor using *send* addresses, or to translate NEWS to *send* addresses.

Parallel primitives that are relevant to our implementation are the *scans* and *spreads* of Blelloch [4]. The $\otimes$-scan primitive, for an associative, binary operator $\otimes$, takes a sequence $\{x_0, x_1, \ldots, x_n\}$ and produces another sequence $\{y_0, y_1, \ldots, y_n\}$ such that $y_i = x_0 \otimes x_1 \otimes \cdots \otimes x_i$. For example, *add-scan* takes as an argument a parallel variable (i.e., a variable with its ith element residing in a memory field of the ith VP) and returns at VP i the value of the parallel variable summed over $j = 0, \ldots, i$. A scan can be applied only to preceding processors (e.g., sum over $j = 0, \ldots, i-1$) or it can be performed in reverse. The $\otimes$-spread primitive, for an associative, binary operator $\otimes$, takes a sequence $\{x_0, x_1, \ldots, x_n\}$ and produces another sequence $\{y_0, y_1, \ldots, y_n\}$ such that $y_i = x_0 \otimes x_1 \otimes \cdots \otimes x_n$. For example, *add-spread* takes as an argument a parallel variable residing in the memories of n active data processors and returns at VP i the value of the parallel variable summed over $j = 0, \ldots, n$. A variation of a scan primitive, denoted as *segmented-$\otimes$-scan*, allows its operation within *segments* of a parallel variable. It takes as arguments a parallel variable and a set of segment bits that specify a partitioning of the VP set into contiguous segments.

REFERENCES

[1] D. P. AHLFELD, R. S. DEMBO, J. M. MULVEY, AND S. A. ZENIOS, *Nonlinear programming on generalized networks*, ACM Transactions on Mathematical Software, 13 (1987), pp. 350–368.

[2] D. P. BERTSEKAS, *On the Goldsten–Levitin–Polyak gradient projection method*, IEEE Trans. Automatic Control, AC-21(2), 1976.

[3] D. P. BERTSEKAS AND M. E. GAFNI, *Projected Newton methods and optimization of multicommodity flows*, IEEE Trans. Automatic Control, AC-28(12), Dec. 1983.

[4] G. E. BLELLOCH, *Vector Models for Data-Parallel Computing*, The MIT Press, Cambridge, MA, 1990.

[5] P. G. CIARLET, *The Finite Element Method for Elliptique Problems*, North-Holland, Amsterdam, 1978.

[6] A. R. CONN, N. I. M GOULD, AND PH. L. TOINT, *Testing a class of methods for solving minimization problems with simple bounds on the variables*, Math. Comp., 50 (1988), pp. 399–430.

[7] R. S. DEMBO AND U. TULOWITZKI, *Local convergence analysis for successive inexact quadratic programming methods*, Working Paper, Series B, No. 78, School of Organization and Management, Yale University, New Haven, CT, Oct. 1984.

[8] L. C. W. DIXON AND P. G. DUCKSBURY, *Finite elements optimization on the* DAP, Computa-

tional Physics Communications, 37 (1984), pp. 187–193.
[9] J. J. DONGARRA, *Performance of various computers using standard linear equations software*, Supercomputing Rev., July 1990, pp. 49–56.
[10] J. C. DUNN, *On the convergence of projected gradient processes to singular critical points*, J. Optim. Theory Appl., 55 (1987), pp. 203–216.
[11] C-G. HAN, P. M. PARDALOS, AND Y. YE, *Computational aspects of an interior point algorithm for quadratic programming problems with box constraints*, in Large Scale Numerical Optimization, T. Coleman and Y. Li, eds., Society for Industrial and Applied Mathematics, Philadelphia, PA, 1990.
[12] G. T. HERMAN, *Image Reconstruction from Projections: The Fundamentals of Computerized Tomography*, Academic Press, New York, 1980.
[13] W. D. HILLIS, *The Connection Machine*, Scientific American, June 1987.
[14] S. LENNART AND C-T. HO, *Optimum broadcasting and personalized communication on hypercubes*, IEEE Trans. on Computers, 38 (1989), pp. 1249–1268.
[15] R. DE LEONE, *Private communication*, 1991.
[16] D. G. LUENBERGER, *Linear and Nonlinear Programming*, Addison-Wesley, Reading, MA, 1984.
[17] M. MCKENNA AND S. A. ZENIOS, *An optimal parallel implementation of a quadratic transportation algorithm*, in Fourth SIAM Conf. Parallel Processing for Scientific Computing, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1990, pp. 357–363.
[18] J. H. MORÉ AND G. TORALDO, *Algorithms for bound constrained quadratic programming problems*, Numer. Math., 55 (1989), pp. 377–400.
[19] ———, *On the solution of large quadratic programming problems with bound constraints*, Report MCS-p77-0589, Argonne National Laboratory, Argonne, IL, 1989.
[20] J. J. MORÉ, *On the performance of algorithms for large-scale bound constrained problems*, in Large Scale Numerical Optimization, T. Coleman and Y. Yi, eds., Society for Industrial and Applied Mathematics, 1990, pp. 32–45.
[21] THINKING MACHINES CORPORATION, *Connection Machine* CM–200 *Series Technical Summary*, Cambridge, MA, 1991.
[22] S. WRIGHT, *Implementing proximal point methods for linear programming*, J. Optim. Theory Appl., 65 (1990), pp. 531–554.
[23] S. A. ZENIOS, ED., *Financial Optimization*. Cambridge University Press, Cambridge, 1993.
[24] S. A. ZENIOS, *Data parallel computing for network-structured optimization problems*, Comput. Optim. Appl., 3 (1994), pp. 199–242.

SIAM J. OPTIMIZATION
Vol. 5, No. 3, pp. 590–640, August 1995

007

A SEQUENTIAL QUADRATIC PROGRAMMING ALGORITHM USING AN INCOMPLETE SOLUTION OF THE SUBPROBLEM*

WALTER MURRAY† AND FRANCISCO J. PRIETO‡

Abstract. We analyze sequential quadratic programming (SQP) methods to solve nonlinear constrained optimization problems that are more flexible in their definition than standard SQP methods. The type of flexibility introduced is motivated by the necessity to deviate from the standard approach when solving large problems. Specifically we no longer require a minimizer of the QP subproblem to be determined or particular Lagrange multiplier estimates to be used. Our main focus is on an SQP algorithm that uses a particular augmented Lagrangian merit function. New results are derived for this algorithm under weaker conditions than previously assumed; in particular, it is not assumed that the iterates lie on a compact set.

Key words. nonlinearly constrained minimization, sequential quadratic programming, quasi-Newton method, large-scale optimization

AMS subject classifications. 49D37, 65K05, 90C30

1. Introduction.

The problem of interest is the following:

$$\text{NP} \qquad \begin{array}{ll} \underset{x\in\Re^n}{\text{minimize}} & F(x) \\ \text{s.t.} & c(x) \geq 0, \end{array}$$

where $F : \Re^n \to \Re$ and $c : \Re^n \to \Re^m$. Since we shall not assume that second derivatives are known, computing x^*, a point satisfying the *first-order Karush–Kuhn–Tucker* (KKT) *conditions* for NP is the best that can be achieved. Such points are feasible and satisfy the following conditions:

$$\nabla F(x^*) = \nabla c(x^*)^T \lambda^*, \qquad \lambda_j^* c_j(x^*) = 0 \quad j = 1, \ldots, m \tag{1.1}$$

for some nonnegative multiplier vector $\lambda^* \in \Re^m$. Whenever the term "KKT point" is used in the following sections, it will mean a point satisfying the first-order KKT conditions for NP. Despite this theoretical limitation, we prefer some KKT points to others to try and satisfy our real purpose of finding a minimizer. For example, if the initial estimate is feasible we do not wish to converge to a nearby KKT point if at that point the objective function is higher.

We use the term *stationary point* to denote a point that is feasible and satisfies (1.1) for some multiplier vector $\lambda \in \Re^m$ that is not necessarily nonnegative.

Typically SQP algorithms generate a sequence of points $\{x_k\}$ converging to a solution, by solving at each point, x_k, a quadratic program (QP) of the form

* Received by the editors January 1, 1990; accepted for publication (in revised form) March 28, 1994. This research was supported by National Science Foundation grant DDM-9204208, Department of Energy grant DE-FG03-92ER25117, Office of Naval Research grant N00014-90-J-1242, and the Bank of Spain.

† Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, California 94305-4022 (walter@sol-walter.stanford.edu).

‡ Department of Statistics and Econometrics, Universidad Carlos III de Madrid.

$$\text{QP} \qquad \begin{array}{ll} \underset{p \in \Re^n}{\text{minimize}} & \nabla F(x_k)^T p + \frac{1}{2} p^T H_k p \\ \text{s.t.} & c(x_k) + \nabla c(x_k) p \geq 0, \end{array}$$

for some positive definite matrix H_k. Let p_k (referred to as the search direction) denote the unique solution to QP. We define $x_{k+1} \equiv x_k + \alpha_k p_k$, where the steplength α_k is chosen to achieve a reduction in a merit function.

SQP algorithms are viewed by many as the best approach to the solution of NP when n is small (< 200). As the size of the problem grows, usually so does the relative importance of the effort to solve QP when compared to the total effort. Indeed, for many large problems the effort to solve QP dominates the total effort.

When the minimizer of QP is used to define the search direction, it is not necessary in any theoretical discussion of an SQP algorithm to define *how* the QP subproblem is solved. Most implementations of SQP methods currently available use an active-set method to solve the QP subproblem. For a comprehensive survey of active-set methods see [18], [13], and [17]. The potential number of iterations to solve a QP using an active-set method grows exponentially with n. In practice the number of iterations grows much more slowly than exponential (if this was not the case active-set methods would be hopelessly inefficient). Nonetheless, the number of iterations required to solve a large QP is usually large. In any implementation of an SQP method it is necessary to limit the number of iterations allowed to solve a given QP subproblem. If the QP solution process is terminated prematurely the SQP algorithm may break down. It is in part for this reason that the development of SQP methods for large-scale problems has been inhibited. Even for small problems there are occasions when the number of QP iterations is excessive. Since the definition of "small" continues to increase as computers become more powerful we can expect the cost of solving the subproblems to grow in importance.

In the algorithms presented here we have endeavored to improve the efficiency of SQP methods by circumventing the need to determine the minimizer of QP. We show that a suitable search direction may be computed from information available at *any* stationary point of QP. Stationary points occur as iterates within most active-set methods to solve QP and for such methods the number of iterations to determine a stationary point increases only linearly with the size of the problem. Consequently, the search direction may be found by applying an active-set method to QP and terminating the procedure early.

It may be thought that by expending much less effort to compute the search direction, the number of iterations for the outer algorithm may increase. However, it has been observed that large numbers of QP iterations are required only when x_k is a poor approximation to x^*, that is, when the QP subproblem does not model the nonlinear problem well. We hypothesize that a search direction based on the minimizer of such subproblems is little better than using information at a stationary point. Our preliminary results reported in §6 support this hypothesis.

Not solving the QP subproblem also implies that we do not know the QP multipliers, which are often used to estimate the multipliers of NP. In general, SQP methods usually use some specific estimate of the NP multipliers in the definition of the method and hence in the proof of convergence. When solving large problems specific definitions of multiplier estimates are not always computationally attractive. In our analysis we allow for flexibility in how multipliers are defined by requiring only that the multiplier estimates satisfy certain conditions.

1.1. Incomplete solutions for QP subproblems. There have been other proposals to define the search direction for an SQP algorithm other than as the minimizer of the QP subproblem. In Dembo and Tulowitzki [9] an algorithm is analyzed for which the search direction p_k has the property that

$$\|p_k - p_k^*\| = o(\|p_k\|),$$

where p_k^* denotes the minimizer for the kth QP subproblem, (unless stated otherwise all norms in the paper are ℓ_2-norms).

We follow a different approach and define a search direction for which the effort to compute it has a guaranteed bound. A different algorithm, but using the same approach, was suggested by Gurwitz and Overton [20]. However, no global convergence results were given for their algorithm.

In the course of solving a QP an active-set method generates iterates that are stationary points. We show that such points may be used to construct a suitable search direction. The step to the stationary point is not generally an adequate search direction. However, if the stationary point is not a minimizer then there exist nonoptimal multipliers. We show how an auxiliary direction may be constructed using information about the nonoptimal multipliers. This auxiliary direction, when combined with the step to the stationary point, gives a suitable search direction.

Terminating the QP algorithm prior to obtaining a solution impacts the SQP algorithm in a number of critical ways. Not only is the search direction different, but also the QP multipliers will not be available. The merit function of principal interest requires the definition of a search direction in the space of the multipliers. In the past, this search direction has been defined using the QP multipliers. The fact that such multipliers are positive was crucial in the analysis of these algorithms. The consequences of terminating the QP solution process early are therefore far reaching.

The remainder of this paper is organized as follows. Section 2 describes the form of the general algorithm, and the definition of the search direction. Section 3 studies the convergence properties of the algorithm; it is shown that such an algorithm is globally convergent. In §4 we show that the algorithm converges superlinearly. We also show that the penalty parameter used in the merit function is bounded. Section 5 considers the use of alternative merit functions. Finally, §6 presents numerical results obtained from an implementation that uses the merit function of principal interest.

2. Description of the algorithm. The search direction we propose could be used with most of the merit functions analyzed in the literature. However, our primary interest is the following merit function:

$$L_A(x, \lambda, s, \rho) = F(x) - \lambda^T(c(x) - s) + \tfrac{1}{2}\rho(c(x) - s)^T(c(x) - s), \tag{2.1}$$

where $s \geq 0$ are slack variables, and the scalar ρ is known as the penalty parameter.

This merit function was suggested by Gill et al. [16] and is used in the SQP code NPSOL. It is similar to merit functions proposed by Wright [34] and Schittkowski [32]. Although our primary interest is this specific merit function, we also show (§5) how the ideas discussed can be extended to the use of other merit functions. The reason for our focus on this merit function is due to the success in practice of NPSOL. The merit function is also used in a new SQP code, LSSQP [10], designed to solve large problems.

The search is performed on an expanded space, including the Lagrange multiplier estimates λ, and the slack variables s. The symbols p, ξ, and q will be used to denote the components of the search direction on the corresponding subspaces. In this case, the value of the merit function as a function of the steplength will be denoted by

$$\phi(\alpha; x, p, \lambda, \xi, s, q, \rho) \equiv L_A(x + \alpha p, \lambda + \alpha\xi, s + \alpha q, \rho). \tag{2.2}$$

The explicit reference to the parameters will be omitted in what follows. The derivative of ϕ with respect to α is denoted by ϕ'. Also, $\phi_k(\alpha)$ and $\phi'_k(\alpha)$ will be used to indicate the values of ϕ and ϕ' evaluated at $(x_k, p_k, \lambda_k, \xi_k, s_k, q_k, \rho_k)$.

The following conventions will be used in the rest of the paper:

$$g_k \equiv \nabla F(x_k), \qquad A_k \equiv \nabla c(x_k), \qquad c_k \equiv c(x_k),$$

and the symbols $\hat{A}_k$ and $\hat{c}_k$ will be used with the same meaning as A_k and c_k, but restricted to the set of active constraints at the given point. The term *active constraint* will be used to designate a constraint that is satisfied exactly at the current point ($c_j(x) = 0$ in NP, or $a_j^T p = -c_j$ in QP), and the set of all constraints active at a given point will be referred to as the *active set* at the point.

The objective function for the QP subproblem will be denoted by $\psi_k(p)$,

$$\psi_k(p) \equiv g_k^T p + \tfrac{1}{2} p^T H_k p. \tag{2.3}$$

Sometimes, ψ will denote the function of one variable $\psi_k(\gamma) \equiv \psi_k(p + \gamma d)$.

For any vector v, the notation v^- will be used to denote the vector whose jth element is defined as

$$v_j^- \equiv -\min(0, v_j).$$

Also, the symbol e denotes the vector $(1, \ldots, 1)^T$, and symbols of the form β_{abc} denote fixed scalars related to properties of the problem, or the implementation of the algorithm, where "abc" identifies the specific scalar represented.

Finally, throughout the paper we will use the symbol $\|u\|$ to denote the ℓ_2-norm of the vector u, unless we explicitly indicate that a different norm is being considered.

2.1. The algorithm. We first present an outline of the algorithm. Given H_0 positive definite, x_0 and λ_0, select $\rho_{-1} \geq 0$, $0 < \sigma < \eta < 1$, $\beta_c \geq \|c^-(x_0)\|_\infty$, $\beta_\mu \geq \|\lambda_0\|$ and $\beta_\rho > 0$.

ALGORITHM ETSQP

$k \leftarrow 0$
repeat
Obtain the search direction p_k from the QP subproblem

$$\begin{array}{ll} \min_p & \psi_k(p) \equiv g_k^T p + \frac{1}{2} p^T H_k p \\ \text{s.t.} & A_k p + c_k \geq 0 \end{array}$$

Compute μ_k, an estimate of λ^* such that $\|\mu_k\| \leq \beta_\mu$
$\xi_k \leftarrow \mu_k - \lambda_k$
if $\rho_{k-1} = 0$
Compute s_k from $(s_k)_j = \max(0, (c_k)_j)$
else
Compute s_k from $(s_k)_j = \max(0, (c_k)_j - (\lambda_k)_j / \rho_{k-1})$
end if
$q_k \leftarrow A_k p_k + c_k - s_k$
if $\phi_k'(0) \leq -\frac{1}{2} p_k^T H_k p_k$
$\rho_k \leftarrow \rho_{k-1}$
else

$$\rho_k \leftarrow \max\Big(2\rho_{k-1}, \frac{\psi_k(p_k) + (2\lambda_k - \mu_k)^T (c_k - s_k)}{\|c_k - s_k\|^2}, \beta_\rho\Big)$$

end if
if $\phi_k(1) \leq \phi_k(0) + \sigma \phi_k'(0)$
$\hat{\alpha} \leftarrow 1$
else
Select $\hat{\alpha} \in (0,1)$ to satisfy

$$\phi_k(\hat{\alpha}) \leq \phi_k(0) + \sigma \hat{\alpha} \phi_k'(0), \quad |\phi_k'(\hat{\alpha})| \leq -\eta \phi_k'(0)$$

end if
while $c(x_k + \hat{\alpha} p_k) \not\geq -\beta_c e$ **or** $\phi_k(\hat{\alpha}) > \phi_k(0) + \sigma \hat{\alpha} \phi_k'(0)$ **do**
$\hat{\alpha} \leftarrow \hat{\alpha}/2$
end do
$\alpha_k \leftarrow \hat{\alpha}$

$$\begin{pmatrix} x_{k+1} \\ \lambda_{k+1} \end{pmatrix} \leftarrow \begin{pmatrix} x_k \\ \lambda_k \end{pmatrix} + \alpha_k \begin{pmatrix} p_k \\ \xi_k \end{pmatrix}$$

Compute g_{k+1}, A_{k+1} and c_{k+1}
Update H_k to form H_{k+1}
$k \leftarrow k + 1$
until *convergence*

The following are some comments on the steps of the algorithm.
(i) At each point x_k, we form the QP subproblem

$$\underset{p \in \Re^n}{\text{minimize}} \; g_k^T p + \frac{1}{2} p^T H_k p \tag{2.4a}$$

$$\text{subject to } A_k p \geq -c_k, \tag{2.4b}$$

and determine a stationary point for QP, that is, a point $\tilde{p}_k$ satisfying

$$g_k + H_k \tilde{p}_k = A_k^T \pi_k, \tag{2.5a}$$

$$A_k \tilde{p}_k + c_k \geq 0, \qquad \pi_k^T (A_k \tilde{p}_k + c_k) = 0, \tag{2.5b}$$

for some vector $\pi_k \in \Re^m$ (the QP multipliers at $\tilde{p}_k$).

From information available at the stationary point we construct a search direction p_k and μ_k an estimate of λ^*. The precise conditions that p_k and μ_k need to satisfy are given later in this section. If $p_k = 0$, we set $\lambda_k = \mu_k$ and terminate. Otherwise, we compute the search direction in the space of the multiplier estimates ξ_k as

$$\xi_k = \mu_k - \lambda_k. \tag{2.6}$$

(ii) The slack variables s_k are computed from

$$(s_k)_j = \begin{cases} \max(0, (c_k)_j) & \text{if } \rho_{k-1} = 0, \\ \max\Big(0, (c_k)_j - \dfrac{(\lambda_k)_j}{\rho_{k-1}}\Big) & \text{otherwise.} \end{cases} \tag{2.7}$$

These values minimize the merit function (2.1) at $(x_k, \lambda_k, \rho_{k-1})$ with respect to the slack variables. The slack variables s_k appear in the merit function (2.1) as part of the term $c_k - s_k$. From (2.7), this term takes the value

$$(c_k)_j - (s_k)_j = \begin{cases} \min(0, (c_k)_j) & \text{if } \rho_{k-1} = 0, \\ \min\Big((c_k)_j, \dfrac{(\lambda_k)_j}{\rho_{k-1}}\Big) & \text{otherwise.} \end{cases} \tag{2.8}$$

The following inequality will be useful in the analysis of the algorithm:

$$\|c_k^-\| \leq \|c_k - s_k\|. \tag{2.9}$$

To simplify the notation in the justification of this result, we drop the subscript k.

If $c_j - s_j = c_j$ then clearly $|c_j - s_j| = |c_j| \geq |c_j^-|$.

If $c_j - s_j \neq c_j$ and $c_j \geq 0$, then $c_j^- = 0 \leq |c_j - s_j|$. Otherwise, $c_j - s_j \neq c_j$ and $c_j < 0$. From (2.8) we get $c_j - s_j < c_j < 0$, and hence $|c_j - s_j| > |c_j| \geq |c_j^-|$. We have shown $|c_j^-| \leq |c_j - s_j|$ under all circumstances, implying (2.9).

(iii) The search direction in the space of the slack variables q_k is set to the vector of slack variables for the QP subproblem, i.e.,

$$q_k = A_k p_k + c_k - s_k. \tag{2.10}$$

For a linear constraint this choice keeps the corresponding slack variable at its optimum value.

(iv) The penalty parameter will not be modified if the condition

$$\phi_k'(0) \leq -\tfrac{1}{2} p_k^T H_k p_k, \tag{2.11}$$

is satisfied, where $\phi_k(\alpha)$ is defined in (2.2). Otherwise, we define the penalty parameter as

$$\rho_k = \max\big(2\rho_{k-1}, \hat{\rho}_k, \beta_\rho\big), \tag{2.12}$$

where β_ρ is some positive constant,

$$\hat{\rho}_k \equiv \frac{\psi_k(p_k) + (2\lambda_k - \mu_k)^T(c_k - s_k)}{\|c_k - s_k\|^2}, \tag{2.13}$$

and ψ_k was defined in (2.3). It will be shown that the definition (2.12) ensures that (p_k, ξ_k, q_k) is a sufficient descent direction for the merit function, in the sense that condition (2.11) holds for this value of the penalty parameter.

(v) The steplength $\alpha_k > 0$ is computed to reduce $\phi_k(\alpha)$ while keeping the constraint violation bounded. The termination conditions for the linesearch are as follows: If

$$\phi_k(1) - \phi_k(0) \leq \sigma\phi_k'(0), \tag{2.14}$$

set $\hat{\alpha} = 1$. Otherwise, find an $\hat{\alpha} \in (0, 1)$ such that

$$\phi_k(\hat{\alpha}) - \phi_k(0) \leq \sigma\hat{\alpha}\phi_k'(0), \tag{2.15a}$$

$$\phi_k'(\hat{\alpha}) \geq \eta\phi_k'(0), \tag{2.15b}$$

where $0 < \sigma < \eta < 1$.

If the condition

$$c(x_k + \hat{\alpha}p_k) \geq -\beta_c e \tag{2.16}$$

holds, we define $\alpha_k = \hat{\alpha}$; otherwise we compute α_k by performing a backtracking linesearch from $\hat{\alpha}$ until (2.15aa) and (2.16) are both satisfied. It will be shown later that such a steplength always exists, and that Algorithm ETSQP is well defined. This definition of the steplength ensures that $c(x_k) \geq -\beta_c e$ for all k. A more sophisticated algorithm could be used to determine α_k when (2.16) does not hold. However, we anticipate such events will be rare.

(vi) Finally, x_k and λ_k are updated from

$$\begin{pmatrix} x_{k+1} \\ \lambda_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ \lambda_k \end{pmatrix} + \alpha_k \begin{pmatrix} p_k \\ \xi_k \end{pmatrix}. \tag{2.17}$$

2.2. The definition of the search direction. At each iteration of Algorithm ETSQP an inner iteration is performed to determine the search direction by solving the QP subproblem (2.4a) using an active-set method. The following is an outline of a suitable algorithm to determine the search direction. The outer iteration subscript has been omitted, and the subscript i refers to the inner iterations.

We assume that positive constants β_p, β_b, γ_M have been defined ($\beta_b \leq 1$).

ALGORITHM SD

Compute p_0 satisfying:

$$Ap_0 + c \geq 0, \qquad \|p_0\| \leq \beta_p \|c^-\|, \qquad g^T p_0 \leq \beta_p \|c^-\|$$

Form $\hat{A}_0$, the active-set matrix at p_0, as the set of all rows in A corresponding to
active QP constraints at p_0
$i \leftarrow 0$
repeat

Compute $\bar{p}_i$ from $\begin{pmatrix} H & \hat{A}_i^T \\ \hat{A}_i & 0 \end{pmatrix} \begin{pmatrix} \bar{p}_i \\ -\hat{\pi} \end{pmatrix} = \begin{pmatrix} -g - Hp_i \\ 0 \end{pmatrix}$

$\gamma_i \leftarrow \min\left(1, \inf_j \left\{ -\dfrac{c_j + a_j^T p_i}{a_j^T \bar{p}_i} \mid a_j^T \bar{p}_i < 0 \right\}\right)$

$p_{i+1} \leftarrow p_i + \gamma_i \bar{p}_i$

Set $\hat{A}_{i+1}$ to be the active-set matrix at p_{i+1}

$i \leftarrow i + 1$

until p_i is a stationary point. $\tilde{p} \leftarrow p_i$

if $\hat{\pi} \geq 0$

$p \leftarrow \tilde{p}$

else

Set $v_r \leftarrow 1$ if $\hat{\pi}_r \leq \beta_b \min_j \hat{\pi}_j$, otherwise set $v_r \leftarrow 0$

Compute $\bar{d}$ by solving: $\min\{\bar{d}^T \bar{d} \mid \hat{A}_i \bar{d} = v\}$

$d \leftarrow \bar{d}/\|\bar{d}\|$

$\gamma \leftarrow \min\left(-\dfrac{(g + H\tilde{p})^T d}{d^T H d}, \inf_j \left\{ -\dfrac{c_j + a_j^T \tilde{p}}{a_j^T d} \mid a_j^T d < 0 \right\}, \gamma_M\right)$

if $\|\tilde{p} + \gamma d\| > \|\tilde{p}\|$

$p \leftarrow \tilde{p} + \gamma d$

else

$p \leftarrow \tilde{p}$

end if

end if

Some comments on this procedure are presented below.

(i) An initial feasible point p_0 of the QP subproblem is obtained. When the minimizer of the QP is used as the search direction, then, given the uniqueness of p, the choice of p_0 is irrelevant. If we determine the search direction from a stationary point that is not a minimizer, the sequence of stationary points that we compute depends directly on the value of p_0. We wish to define the initial point in such a manner that *all* stationary points are satisfactory points at which to terminate the solution process. It will be seen that the following conditions on p_0 are sufficient to ensure our objective.

For some constant $\beta_p > 0$,

$$\|p_0\| \leq \beta_p \|c^-\| \quad \text{and} \quad g^T p_0 \leq \beta_p \|c^-\|. \tag{2.18}$$

(ii) A sequence of feasible descent steps are taken, for example, by first computing the unique step $\bar{p}_i$ to the minimizer of the QP on the current working set as the least-

length solution of the system of equations

$$\begin{pmatrix} H & \hat{A}_i^T \\ \hat{A}_i & 0 \end{pmatrix} \begin{pmatrix} \bar{p}_i \\ -\hat{\pi} \end{pmatrix} = \begin{pmatrix} -g - Hp_i \\ 0 \end{pmatrix}, \tag{2.19}$$

where p_i is the current estimate. A step γ_i is taken, where γ_i is obtained as either one or the step to the nearest constraint,

$$\gamma_i = \min\Big(1, \inf_j \Big\{ -\frac{c_j + a_j^T p_i}{a_j^T \bar{p}_i} \mid a_j^T \bar{p}_i < 0 \Big\}\Big). \tag{2.20}$$

The QP algorithm may be terminated at *any* stationary point $\tilde{p}$. (Algorithm SD is terminated at the first stationary point.) It will be seen in the proofs that to always use $\tilde{p}$ as the search direction will not in general ensure convergence.

(iii) If $\tilde{p}$ is the minimizer of the QP subproblem the search direction p is defined as $p \equiv \tilde{p}$, else

$$p \equiv \begin{cases} \tilde{p} + \tilde{\gamma} d & \text{if } \|\tilde{p}\| < \|\tilde{p} + \tilde{\gamma} d\|, \\ \tilde{p} & \text{otherwise,} \end{cases} \tag{2.21}$$

where the vector d and the scalar $\tilde{\gamma}$ are computed with the following properties:

d satisfies $\hat{A}_i d \geq 0$, and $\|d\|_\infty = 1$.

The rate of descent along d is "sufficiently" large. By this we mean d satisfies

$$\tilde{g}^T d \leq \beta_d \tilde{g}^T d^*, \tag{2.22}$$

where $0 < \beta_d \leq 1$, $\tilde{g} = H\tilde{p} + g$ and d^* solves

$$\begin{array}{ll} \min_d & d^T \tilde{g} \\ \text{s.t.} & \hat{A}_i d \geq 0, \\ & \|d\|_\infty \leq 1. \end{array} \tag{2.23}$$

There are many procedures for computing a suitable vector d. For example, if the singular values of $\hat{A}_i$ are bounded above and below and $\hat{A}_i$ has full row rank then a suitable d may be computed as follows. Define a vector v to satisfy

$$v_j = \begin{cases} 1 & \text{if } \hat{\pi}_j < 0, \\ 0 & \text{otherwise.} \end{cases}$$

We then compute $\bar{d}$ as the least-length solution of $\hat{A}_i y = v$ and define d as

$$d \equiv \bar{d} / \|\bar{d}\|_\infty.$$

For this direction d we have

$$\tilde{g}^T d = \hat{\pi}^T \hat{A}_i d = \hat{\pi}^T v \leq \frac{1}{\|\bar{d}\|_\infty} \min_j \hat{\pi}_j. \tag{2.24}$$

Under the assumptions made on $\hat{A}_i$ it follows $\|\bar{d}\|_\infty$ is bounded. We shall now show d is a "sufficient" descent direction. Let u^* denote the solution of the problem

$$\begin{array}{ll} \min_u & \hat{\pi}^T u \\ \text{s.t.} & u \geq 0, \quad \|u\|_\infty \leq 1. \end{array}$$

We have $\hat{\pi}^T u^* \geq m \min_j \hat{\pi}_j$. Define $\bar{u} \equiv \hat{A}_i d^*$. We get

$$\tilde{g}^T d^* = \hat{\pi}^T \hat{A}_i d^* = \hat{\pi}^T \bar{u} \geq m \|\bar{u}\| \min_j \hat{\pi}_j,$$

since $\bar{u} \geq 0$. If $\|\hat{A}_i\|$ is bounded it follows $\|\bar{u}\|$ is bounded. From (2.24) and the above inequality we get

$$\tilde{g}^T d = \hat{\pi}^T \hat{A}_i d = \hat{\pi}^T v \leq \frac{1}{\|\bar{d}\|_\infty} \min_j \hat{\pi}_j \leq \frac{1}{m\|\bar{u}\|\|\bar{d}\|_\infty} \tilde{g}^T d^*.$$

Lemma 2.1 presents some properties of the solutions of (2.23). These properties are based on the observation that the cost vector and the coefficients of each constraint can be normalized without affecting the feasible region or the solutions of the problem. Since we are concerned with sequences we reintroduce the outer subscript. Define $\hat{g}_k \equiv \tilde{g}_k / \|\tilde{g}_k\|$ and a matrix B_k whose jth row is the normalized jth row of $\hat{A}_i$. The problem

$$\begin{array}{ll} \min_d & \hat{g}_k^T d \\ \text{s.t.} & B_k d \geq 0, \\ & \|d\|_\infty \leq 1, \end{array} \tag{2.25}$$

has the same feasible region and the same solutions as (2.23). We tacitly assume no row of $\hat{A}_i$ is a zero vector, otherwise it could be omitted from both problems. Likewise, if $\|\tilde{g}_k\| = 0$ it implies $\tilde{p}_k$ is the minimizer of the QP.

LEMMA 2.1. *Given a subsequence of iterates $\{x_k\}$, generated by Algorithm* ET-SQP *and such that for all of them $\tilde{p}_k \neq p_k^*$, the directions d_k^* obtained as solutions of* (2.23) *at each point satisfy $\tilde{g}^T d_k^* < 0$ and $\|d_k^*\|_\infty = 1$. Furthermore, if $\tilde{g}_k^T d_k^* \to 0$ along the subsequence, then either $\tilde{g}_k \to 0$ or for any limit $(\hat{g}, B)$ of the sequence $\{(\hat{g}_k, B_k)\}$, defined as in* (2.25), *it holds that $\hat{g} = B^T \nu$, with $\nu \geq 0$.*

Proof. Since $p_k^* - \tilde{p}_k$ is a feasible descent direction of (2.23) at $d = 0$ it follows that $d = 0$ is not optimal, and the solutions of (2.23) satisfy $\tilde{g}_k^T d_k^* < 0$ and $\|d_k^*\|_\infty = 1$.

Consider now the sequence of problems of the form (2.25) and the problem obtained from a limit of the sequence $\{(\hat{g}_k, B_k)\}$. The feasible regions of all problems are compact convex polytopes; if we denote the vertices of the polytope corresponding to problem k by $\{d_k^l\}$, where the index l takes a finite number of different values, it holds that for each l, $d_k^l \to d^l$, a vertex for the polytope corresponding to the feasible region of the limit problem (assume without loss of generality that the convergent subsequence has been chosen so that the number of vertices is the same for all problems in the subsequence).

Any feasible point of the limit problem, d, can be written as a convex combination of the vertices d^l, $d = \sum_l \zeta_l d^l$. We can then construct for any feasible d a sequence $\{d^k\}$, where each point d^k is defined as $d^k \equiv \sum_l \zeta_l d_k^l$, having the properties that d^k is feasible for the kth problem (2.25), and $d^k \to d$.

If $\tilde{g}_k \not\to 0$ then $\tilde{g}_k^T d_k^* \to 0$ implies $\hat{g}_k^T d_k^* \to 0$ and it must hold that $d = 0$ is an optimal solution of the limit problem, implying that there exists a vector $\nu \geq 0$ satisfying $\hat{g} = B^T \nu$. □

Note that $\tilde{g}_k^T d_k^* \to 0$, if and only if $\tilde{g}_k^T d_k \to 0$, where d_k is a sufficient descent direction.

The scalar $\tilde{\gamma}$ is given by

$$\tilde{\gamma} = \min(\bar{\gamma}, \hat{\gamma}, \gamma_M), \tag{2.26}$$

where γ_M is a specified upper bound on the steplength,

$$\bar{\gamma} = \inf_j \Big\{ -\frac{c_j + a_j^T \tilde{p}}{a_j^T d} \mid a_j^T d < 0 \Big\}, \tag{2.27}$$

is the largest feasible step from $\tilde{p}$ along d, and

$$\hat{\gamma} = -\frac{(g + H\tilde{p})^T d}{d^T H d}, \tag{2.28}$$

is the step to the minimizer of $\psi(\tilde{p} + \gamma d)$.

2.3. The multiplier estimates. Equation (2.6) defining the search direction on the multiplier space ξ_k requires the computation of an estimate μ_k for the Lagrange multipliers. The estimates $\{\mu_k\}$ are then used to update $\{\lambda_k\}$, the Lagrange multiplier estimate used in the merit function. To allow flexibility in algorithm design we have chosen to specify conditions on the multipliers estimates μ_k rather than give explicit definitions.

It will be shown that the following conditions on μ_k are sufficient to ensure that the algorithm is globally convergent.

MC1. The estimates μ_k are uniformly bounded in norm, that is $\|\mu_k\| \leq \beta_\mu < \infty$.

MC2. The complementarity condition $\mu_k^T(A_k p_k + c_k) = 0$ is satisfied at all iterations.

We may satisfy these conditions by choosing $\mu_k = 0$. Condition MC2 is made for convenience; condition MC1 and the form in which the multiplier estimates are updated imply that $\{\lambda_k\}$ are uniformly bounded.

LEMMA 2.2. *If condition* MC1 *holds, then* $\|\lambda_k\| \leq \beta_\mu$ *for all* k.

Proof. The proof is by induction. We select β_μ to satisfy $\|\lambda_0\| \leq \beta_\mu$. From (2.17),

$$\lambda_{k+1} = \lambda_k + \alpha_k(\mu_k - \lambda_k), \quad k \geq 0. \tag{2.29}$$

Using norm inequalities and $0 < \alpha_k \leq 1$, we have

$$\|\lambda_{k+1}\| \leq \alpha_k\|\mu_k\| + (1 - \alpha_k)\|\lambda_k\| \leq \alpha_k\beta_\mu + (1 - \alpha_k)\beta_\mu = \beta_\mu,$$

as required. □

2.4. Second-order information. We choose the matrices $\{H_k\}$ to be positive definite and bounded, with bounded condition number. In practice, such matrices may be generated (see [15]) by updating a quasi-Newton approximation to the Hessian of the Lagrangian function or the Hessian of the augmented Lagrangian function in each iteration together with certain safeguards (for example, if the factors of H_k are updated, by enforcing bounds on the size of the elements, and ensuring sufficiently positive diagonal elements). These conditions can be written as follows:

HC1. $\beta_{lvH} < \infty$ is the largest eigenvalue of $\{H_k\}$.

HC2. $\beta_{svH} > 0$ is the smallest eigenvalue of $\{H_k\}$.

3. Global convergence results. The results in this section establish global convergence properties for Algorithm ETSQP, under certain assumptions on the problem NP. We first introduce these assumptions, and then, under the condition that they hold, we prove the following results:

(i) The iterates $\{x_k\}$ lie on a compact set.

- In Lemma 3.1 we show that the quantities associated with the algorithm are well defined at all points.
  In Lemma 3.2 it is shown that if $\|x_k\|$ is large then $\|p_k\|$ cannot be arbitrarily small.
- In Lemma 3.3 we show that p computed using Algorithm SD satisfies

$$\psi(p) \equiv g^T p + \tfrac{1}{2} p^T H p \leq -\beta_1 p^T H p + \beta_2 \|c - s\|,$$

  where β_1 and β_2 are positive constants.
- Lemma 3.4 proves that the sequence $\{x_k\}$ lies on a compact set.
- Lemma 3.5 shows that the sequence $\{p_k\}$ also remains bounded.

(ii) The sequence $\{\|p_k\|\}$ dominates the sequence $\{\|x_k - x^*\|\}$, where x^* denotes a KKT point closest to x_k. The main implication of this result is that $\|p_k\| \to 0$ is sufficient to ensure that $x_k \to x^*$, a KKT point of NP.
- It is shown in Lemma 3.6 that the KKT points for problem NP are isolated.
- Lemma 3.7 shows that if $\|x_k - x^*\| \to 0$ along a subsequence then along the same subsequence $\|\pi_k - \lambda^*\| \to 0$.
- Lemma 3.8 introduces another preliminary result, proving that if $p_k \to 0$ along a subsequence then along this subsequence $\|x_k - x^*\| \to 0$, where x^* is a KKT point for NP nearest to x_k. Moreover, for large enough k, p_k is the minimizer of the QP subproblem, and the correct active set at x^* is identified.
- The proof that $\|p_k\|$ dominates $\|x_k - x^*\|$ is given in Lemma 3.9.

(iii) Bounds on the growth of the penalty parameter ρ_k. We cannot prove that ρ_k will remain bounded in the algorithm without stronger conditions on the multiplier estimate μ_k, but we can show that its growth is bounded by certain quantities related with the algorithm, and that is enough to prove convergence.
- We show in Lemma 3.10 that at all the iterations where the penalty parameter is modified the following bounds hold,

$$\rho_k \|c_k - s_k\| \leq N \quad \text{and} \quad \rho_k \|p_k\|^2 \leq N.$$

- In Lemmas 3.11 and 3.12 we show that similar inequalities hold at all iterations.

(iv) The steplength α_k is bounded away from zero if we are not close to a solution.
- We first need a bound on the second derivatives of $\phi(\alpha)$. In Lemma 3.13 we prove that $\phi_k(\alpha_k) \leq N$ for some positive constant N.
- In Lemma 3.14 we show that, if $\|p_k\|$ is large enough, there exists a value $\bar{\alpha} > 0$ independent of the iteration such that $\alpha_k \geq \bar{\alpha}$.

(v) In Theorem 3.15 we show that $x_k \to x^*$.

(vi) Finally, we prove that $\lambda_k \to \lambda^*$.
- This result requires stronger conditions on the multiplier estimate μ_k than just MC1 and MC2. We start by introducing a third condition MC3.
- Lemma 3.16 strengthens the result in Lemma 3.14 showing that, under the new conditions on the multipliers, α_k is uniformly bounded away from zero.
- In Theorem 3.17 we show that $\lambda_k \to \lambda^*$.

3.1. Assumptions. Some of the following assumptions make use of the concepts of stationary points and KKT points at infinity. We will say that NP has a stationary point at infinity if there exist sequences $\{x_k\}$ and $\{\eta_k\}$ such that $\|x_k\| \to \infty$ and/or $\|\eta_k\| \to \infty$, and

$$c_k^- \to 0, \quad A_k^T \eta_k - g_k \to 0, \quad \eta_k^T c_k \to 0.$$

As before let B_k denote a matrix whose rows are the normalized rows of A_k and $\hat{g}_k$ denote the normalized gradient vector. Define ν_k so that $A_k^T \eta_k - g_k = (B_k^T \nu_k - \hat{g}_k)\|g_k\|$. If in addition to the preceding conditions we have $\nu \geq 0$, where ν indicates the limit of some subsequence $\{\nu_k\}$, we then say there is a KKT point at infinity.

Finally, we will say that strict complementarity does *not* hold at some stationary point at infinity if for the preceding sequences and some constraint j we have

$$(c_k)_j \to 0 \text{ and } (\eta_k)_j \to 0.$$

We make the following assumptions.

A1. For some constant $\beta_c > 0$, the global minimum of the problem

$$\begin{array}{ll} \underset{x \in \Re^n}{\text{minimize}} & F(x) \\ \text{s.t.} & c(x) \geq -\beta_c e, \end{array}$$

is bounded below.

A2. There exist no KKT points at infinity for problem NP.

A3. F, c_j, and their first and second derivatives are continuous and uniformly bounded in norm on a compact set.

A4. The Jacobian corresponding to the active constraints at all KKT points has full rank.

A5. A feasible point p_{k_0} exists to all the QP subproblems, satisfying

$$\|p_{k_0}\| \leq \beta_p \|c_k^-\| \quad \text{and} \quad g_k^T p_{k_0} \leq \beta_p \|c_k^-\|$$

for some constant $\beta_p > 0$.

A6. Strict complementarity holds at all stationary points of NP, including stationary points at infinity, if they exist.

A7. The reduced Hessian of the Lagrangian function is nonsingular at all KKT points. The larger the value of β_c, the stronger is assumption A1. There will be problems, for example $F(x) = f(x)^T f(x)$, where it is known a priori that Assumption A1 holds with $\beta_c = \infty$. Also, if A1 does not hold with $\beta_c = 0$ then it is possible for any reasonable algorithm to diverge.

Assumption A5 imposes conditions on the initial point for the QP. It is possible that no point satisfies these conditions; this would be the case for example if one of the QP subproblems generated by the algorithm is not feasible. Nevertheless, by introducing an additional variable it is possible to construct a modified problem for which satisfying the conditions on p_{k_0} is trivial. Consider the problem

$$\begin{array}{ll} \underset{(x,\bar{x}) \in \Re^{n+1}}{\text{minimize}} & \mathcal{F}(x,\bar{x}) \equiv (1-\omega)F(x) + \omega\bar{x} \\ \text{s.t.} & c(x) + \bar{x}e \geq 0 \quad \text{and} \quad \bar{x} \geq 0, \end{array} \tag{3.1}$$

where $\bar{x} \in \Re$ and $\omega \in [0,1]$. The KKT points for this problem are also KKT points for NP if NP is feasible and ω is sufficiently close to one. The modified problem is

always feasible, and the corresponding QP subproblem takes the form

$$\begin{array}{ll} \underset{(p,\bar{p})\in\Re^{n+1}}{\text{minimize}} & (1-\omega)g_k^T p + \omega\bar{p} + \frac{1}{2}\begin{pmatrix} p^T & \bar{p} \end{pmatrix} H_k \begin{pmatrix} p \\ \bar{p} \end{pmatrix} \\ \text{s.t.} & c_k + A_k p + \bar{x}_k e + \bar{p}e \geq 0, \\ & \bar{x}_k + \bar{p} \geq 0. \end{array}$$

For this QP subproblem the point

$$p_0 = \begin{pmatrix} p \\ \bar{p} \end{pmatrix} = \begin{pmatrix} 0 \\ \|(c_k + \bar{x}_k e)^-\|_\infty \end{pmatrix}$$

is feasible since we can ensure that $\bar{x}_k \geq 0$. Therefore there always exists a feasible point that satisfies A5 with $\beta_p = 1$ since $\|p_0\| = \|(c_k + \bar{x}_k e)^-\|_\infty$ and

$$\nabla \mathcal{F}_k^T p_0 = \begin{pmatrix} (1-\omega)g_k^T & \omega \end{pmatrix} \begin{pmatrix} p \\ \bar{p} \end{pmatrix} = \omega\|(c_k + \bar{x}_k e)^-\|_\infty \leq \|(c_k + \bar{x}_k e)^-\|_\infty,$$

implying that the conditions on p_{k_0} in Assumption A5 are trivial to satisfy for (3.1).

3.2. Existence of the iterates. We start by showing that all the quantities associated with the algorithm are well defined. In particular, we show that the choice of penalty parameter ensures (2.11) is satisfied and that the steplength exists.

LEMMA 3.1. *Under Assumptions* A3, A5 *and conditions* HC1, HC2, *the procedures given in the algorithm to compute the values of the penalty parameter* ρ_k *and the steplength* α_k *are well defined.*

Proof. We drop the subscript k denoting the iteration number, to simplify the notation.

Consider the gradient of the merit function L_A, defined in (2.1), with respect to x, λ, and s,

$$\nabla L_A(x,\lambda,s) = \begin{pmatrix} g(x) - A(x)^T\lambda + \rho A(x)^T(c(x) - s) \\ -(c(x) - s) \\ \lambda - \rho(c(x) - s) \end{pmatrix}. \tag{3.2}$$

It follows from (2.6), (2.10), and (2.2) that $\phi'(0)$ is given by

$$\begin{aligned} \phi'(0) &= p^T g - p^T A^T\lambda + \rho p^T A^T(c - s) - (c - s)^T\xi + \lambda^T q - \rho q^T(c - s) \\ &= p^T g + (2\lambda - \mu)^T(c - s) - \rho\|c - s\|^2, \end{aligned} \tag{3.3}$$

where g, A, and c are evaluated at x.

If $\|c - s\| = 0$, from (2.9) and (2.18) we have $p_0 = 0$, and since $\psi(p) = p^T g + \frac{1}{2}p^T H p \leq \psi(p_0) = 0$ it follows that

$$\phi'(0) = p^T g \leq -\tfrac{1}{2}p^T H p,$$

implying that ρ does not need to be modified.

If $\|c - s\| > 0$, we obtain from (3.3) that for $\rho = \hat{\rho}$ (defined in (2.13))

$$\phi'(0) = g^T p + (2\lambda - \mu)^T(c - s) - \hat{\rho}\|c - s\|^2 = -\tfrac{1}{2}p^T H p,$$

which implies the desired descent condition (2.11) is satisfied for all $\rho \geq \hat{\rho}$.

An immediate consequence of (2.11) and the properties of H_k is the following bound on the directional derivative:

$$\phi_k'(0) \leq -\tfrac{1}{2}\beta_{svH}\|p_k\|^2. \tag{3.4}$$

It follows from the procedure to increase the value of the penalty parameter (see (2.12)) that $\rho_k \to \infty$ if and only if the parameter is increased an infinite number of times.

We also need to prove that the value of α_k introduced in the algorithm is well defined. We show that if condition (2.14) is not satisfied, a steplength $\hat{\alpha}_k \in (0,1)$ that satisfies conditions (2.15) always exists (see, for example, Moré and Sorensen [23]).

Define the functions

$$\chi(\alpha) \equiv \phi(\alpha) - \phi(0) - \sigma\alpha\phi'(0),$$
$$\zeta(\alpha) \equiv \phi'(\alpha) - \eta\phi'(0),$$

and note that from $\sigma < \eta$ and $\phi'(0) < 0$, implied by (2.11), we have

$$\chi'(\alpha) = \phi'(\alpha) - \sigma\phi'(0) < \phi'(\alpha) - \eta\phi'(0) = \zeta(\alpha) \tag{3.5}$$

for any α.

If (2.14) does not hold,

$$\phi(1) - \phi(0) > \sigma\phi'(0) \quad \Rightarrow \quad \chi(1) > 0,$$

and we also have $\chi(0) = 0$. From these two results and the mean-value theorem, there will be a point $\breve{\alpha} \in [0,1]$ such that $\chi'(\breve{\alpha}) > 0$, and from (3.5), $\zeta(\breve{\alpha}) > 0$.

From $\phi'(0) < 0$ we have $\zeta(0) < 0$, and the continuity of ζ (Assumption A3) will imply the existence of a zero of ζ in $(0, \breve{\alpha})$. Let $\hat{\alpha}$ denote the smallest point in $(0, \breve{\alpha})$ such that $\zeta(\hat{\alpha}) = 0$, that is,

$$\phi'(\hat{\alpha}) = \eta\phi'(0), \tag{3.6}$$

and condition (2.15b) is satisfied at $\hat{\alpha}$.

From $\zeta(0) < 0$ we must have

$$\zeta(\alpha) < 0 \quad \forall\alpha \in [0,\hat{\alpha}) \quad \Leftrightarrow \quad \phi'(\alpha) < \eta\phi'(0) \quad \forall\alpha \in [0,\hat{\alpha}), \tag{3.7}$$

implying that condition (2.15b) is not satisfied for any point in $[0,\hat{\alpha})$.

Finally, from (3.5) and (3.7), we have

$$\chi'(\alpha) < 0 \quad \forall\alpha \in [0,\hat{\alpha}),$$

and this together with $\chi(0) = 0$ implies $\chi(\hat{\alpha}) < 0$, that is,

$$\phi(\hat{\alpha}) - \phi(0) < \sigma\hat{\alpha}\phi'(0), \tag{3.8}$$

showing that $\hat{\alpha}$ satisfies both conditions (2.15) simultaneously.

We still need to consider condition (2.16). For the function $h(\alpha) \equiv c(x+\alpha p)+\beta_c e$ we have from (2.4b)

$$h'(0) = Ap \geq -c.$$

If $-\frac{1}{2}\beta_c \geq c_j \geq -\beta_c$, we have $h_j(0) \geq 0$ and $h_j'(0) \geq \frac{1}{2}\beta_c > 0$; if $c_j \geq -\frac{1}{2}\beta_c$ then $h_j(0) \geq \frac{1}{2}\beta_c > 0$ and in any case there exists a value $\tilde{\alpha} > 0$ such that $h_j(\alpha) \geq 0$ (implying $c_j(x + \alpha p) \geq -\beta_c$) for all j and all $\alpha \in [0,\tilde{\alpha}]$, implying that for $\alpha \in [0, \min(\hat{\alpha},\tilde{\alpha})]$ both conditions (2.15a) and (2.16) hold simultaneously. □

This lemma implies that all the quantities associated with the algorithm are well defined.

3.3. Boundedness of the iterates. To prove global convergence we show first that if Assumptions A1 and A2 hold, all points in the sequence $\{x_k\}$ generated by the algorithm lie on a compact set. We start by showing that for $\|x_k\|$ large enough we cannot have $\|p_k\|$ arbitrary small.

LEMMA 3.2. *Under Assumptions* A2 *and* A6 *and condition* HC1, *there exist positive constants* M *and* ϵ *such that* $\|x_k\| \geq M \Rightarrow \|p_k\| \geq \epsilon$.

Proof. Assume this result does not hold. Then, for any positive constants M and ϵ we can find iterates such that $\|x_k\| \geq M$ and $\|p_k\| < \epsilon$, and we could construct a sequence $\{x_k\}$, and its associated sequence $\{p_k\}$, along which $\|x_k\| \to \infty$ and $\|p_k\| \to 0$. For this sequence, from $\|p_k\| \to 0$ and (2.4b), we must have $\|c_k^-\| \to 0$. Also, from the definition of p_k, (2.21), it must hold that $\|\tilde{p}_k\| \to 0$, and from (2.5a) and MC1, we must have

$$\|A_k^T \pi_k - g_k\| = \|H_k \tilde{p}_k\| \to 0.$$

Since $\|p_k\| \to 0$ and $\|\tilde{p}_k\| \to 0$, using (2.21) and $\|d_k\| = 1$, we also have either $\tilde{\gamma}_k \to 0$ or $\tilde{\gamma}_k = 0$ for k large enough. It then follows from (2.26) that either $\min(\bar{\gamma}_k, \hat{\gamma}_k) \to 0$ or $\hat{\gamma}_k = \bar{\gamma}_k = 0$ for k large enough. If $\bar{\gamma}_k \to 0$ along a subsequence, then (2.27) implies for some constraint j that $(\pi_k)_j \to 0$ and $c_j(x_k) \to 0$, but this would contradict Assumption A6. If $\hat{\gamma}_k \to 0$ along a subsequence, then from (2.28) and Lemma 2.1 we get $\nu \geq 0$ in the limit, where ν is now defined as a limit point of $\{\nu_k\}$, where $B_k^T \nu_k = \hat{g}_k$.

The properties of this sequence,

$$\|x_k\| \to \infty, \quad \|c_k^-\| \to 0, \quad \|A_k^T \pi_k - g_k\| \to 0,$$

together with $\tilde{p}_k \to 0$ and $\nu \geq 0$, imply that there exists a KKT point at infinity, which violates Assumption A2, so the lemma must hold. □

Another result we need for the compactness proof is a bound on the value of the QP objective function at the incomplete solution for the QP.

LEMMA 3.3. *Under Assumption* A5 *and conditions* HC1, HC2, *for* p *computed by Algorithm* SD *there exist constants* $\beta_1 > 0$ *and* $\beta_2 > 0$ *such that*

$$\psi(p) \equiv g^T p + \tfrac{1}{2} p^T H p \leq -\beta_1 p^T H p + \beta_2 \|c - s\|.$$

Proof. The result will be shown by considering first the initial point for the QP, p_0, and then the descent achieved in each QP iteration.

By definition

$$\psi(p_0) = -\tfrac{1}{2} p_0^T H p_0 + g^T p_0 + p_0^T H p_0.$$

Since $\|p_0\| \leq \beta_p \|c^-\|$ and $g^T p_0 \leq \beta_p \|c^-\|$ (Assumption A5), condition HC1 on H implies

$$\psi(p_0) \leq -\tfrac{1}{2} p_0^T H p_0 + \beta_p \|c^-\| + \beta_{lvH} \beta_p^2 \|c^-\|^2. \tag{3.9}$$

Consider the quadratic function $b\gamma + \frac{1}{2} c\gamma^2$, where $b < 0$ and $c > 0$; then for all $\gamma \in [0, -b/c]$ (between 0 and the minimizer), we have

$$\gamma \leq -\frac{b}{c} \quad \Rightarrow \quad \gamma(b + c\gamma) \leq 0 \quad \Rightarrow \quad b\gamma + \tfrac{1}{2} c\gamma^2 \leq -\tfrac{1}{2} c\gamma^2. \tag{3.10}$$

The change in the QP objective function at any intermediate QP iteration i can be represented as

$$\psi(p_{i+1}) - \psi(p_i) = \tfrac{1}{2}\gamma_i^2 d_i^T H d_i + \gamma_i (g + Hp_i)^T d_i, \tag{3.11}$$

where d_i is used to denote the QP step obtained from (2.19) or the final step d defined in (2.22), and γ_i is a feasible steplength bounded by the steplength to the minimizer along v_i, as defined in (2.20) or (2.26). We have $d_i^T H d_i > 0$ (from condition HC2) and $(g + Hp_i)^T d_i < 0$ (from (2.22)), implying that we can apply the bound (3.10) to (3.11) to obtain

$$\psi(p_{i+1}) - \psi(p_i) \leq -\tfrac{1}{2}\gamma_i^2 d_i^T H d_i. \tag{3.12}$$

If we have taken N iterations to compute p (the search direction), by adding the inequalities (3.12) for $i = 0, \ldots, N$ and using (3.9) we obtain

$$\begin{aligned} \psi(p) &= \psi(p_0) + \sum_{i=1}^{N} (\psi(p_i) - \psi(p_{i-1})) \\ &\leq -\tfrac{1}{2}\left(p_0^T H p_0 + \sum_{i=1}^{N} \gamma_i^2 d_i^T H d_i \right) + \beta_p \|c^-\| + \beta_{lvH} \beta_p^2 \|c^-\|^2. \end{aligned} \tag{3.13}$$

We can use the convexity of the function $p^T H p$ (implied by property HC2) to write

$$p_0^T H p_0 + \sum_{i=1}^{N} \gamma_i^2 d_i^T H d_i \geq \frac{1}{N+1} \left(p_0 + \sum_{i=1}^{N} \gamma_i d_i \right)^T H \left(p_0 + \sum_{i=1}^{N} \gamma_i d_i \right) = \frac{1}{N+1} p^T H p.$$

This result together with (3.13) implies

$$\psi(p) \leq -\frac{1}{2(N+1)} p^T H p + \beta_p \|c^-\| + \beta_{lvH} \beta_p^2 \|c^-\|^2. \tag{3.14}$$

Since $c^- \geq \beta_c e$ the desired result follows from this inequality and (2.9). □

We can now prove the main result of this section.

LEMMA 3.4. *Under Assumptions* A1, A2, A3, A5, *and* A6, *and conditions* MC1, HC1 *and* HC2, *the sequence* $\{x_k\}$ *generated by the algorithm lies on a compact set.*

Proof. First we show the set of points at which the penalty parameter is modified lies on a compact set. If ρ_k remains bounded it follows from the manner the penalty parameter is modified, (2.12), that there is only a finite set of such points. Therefore we need only study the case when $\rho_k \to \infty$. Consider the iterations k where the penalty parameter is modified. From condition MC1 and the boundedness of the multiplier estimates λ_k (Lemma 2.2), we have

$$\|2\lambda_k - \mu_k\| \leq 2\|\lambda_k\| + \|\mu_k\| \leq 3\beta_\mu. \tag{3.15}$$

This result, together with Lemma 3.3 and the definition of the penalty parameter (2.13), gives

$$\begin{aligned} \rho_k \|c_k - s_k\|^2 &\leq g_k^T p_k + \tfrac{1}{2} p_k^T H_k p_k + (2\lambda_k - \mu_k)^T (c_k - s_k) \\ &\leq (\beta_1 + 3\beta_\mu) \|c_k - s_k\| - \beta_1 p_k^T H_k p_k. \end{aligned} \tag{3.16}$$

As we have assumed $\rho_k \to \infty$, (3.16) implies $\|c_k - s_k\| \to 0$, and from (2.9) also $\|c_k^-\| \to 0$.

From Lemma 3.3 and (3.15) we have

$$\omega_k \equiv g_k^T p_k + (2\lambda_k - \mu_k)^T (c_k - s_k) \tag{3.17a}$$

$$\leq -\tfrac{1}{2} p_k^T H_k p_k - \beta_1 p_k^T H_k p_k + (\beta_1 + 3\beta_\mu)\|c_k - s_k\|. \tag{3.17b}$$

If $\|p_k\| \geq \epsilon > 0$ along an infinite subsequence, then it follows from $\|c_k - s_k\| \to 0$ and MC2 that there exists an index K such that for all $k \geq K$ in the subsequence,

$$(\beta_p + 3\beta_\mu)\|c_k - s_k\| \leq \beta_1 p_k^T H_k p_k.$$

From (3.17b) we obtain the following bound on ω_k,

$$\omega_k \leq -\tfrac{1}{2} p_k^T H_k p_k, \tag{3.18}$$

for $k \geq K$. From (3.17a) and the bounds (3.18) and (3.3), we have for sufficiently large k

$$\phi_k'(0) = \omega_k - \rho_k \|c_k - s_k\|^2 \leq \omega_k \leq -\tfrac{1}{2} p_k^T H_k p_k.$$

This last inequality implies that ρ_k is not modified for all $k \geq K$, which contradicts our assumption that the penalty parameter was modified an infinite number of times.

We have shown that $\|p_k\| \to 0$ along the subsequence at which the penalty parameter is modified. The boundedness of $\|x_k\|$ along this subsequence follows from Lemma 3.2.

We now consider those points corresponding to iterations where the penalty parameter is not modified. From condition (2.16) on the linesearch and Assumption A1, we have $F(x_k) \geq \beta_F > -\infty$ for all k. Also, from Lemma 2.2 $\|\lambda_k\|$ is bounded, implying that

$$L_A(x_k, \lambda_k, s_k, \rho_k) \geq \beta_F - \max\left(\frac{\beta_\mu^2}{2\beta_\rho}, m\beta_\mu\beta_c\right) > -\infty. \tag{3.19}$$

Since $\|x_k\|$ is bounded when $\rho_k \neq \rho_{k-1}$ and $L_A(x_k, \lambda_k, s_k, \rho_k)$ is *reduced* when $\rho_k = \rho_{k-1}$ it follows that $L_A(x_k, \lambda_k, s_k, \rho_k)$ is bounded. Moreover, for a sequence of iterations for which ρ_k is not changed the reduction in $L_A(x_k, \lambda_k, s_k, \rho_k)$ is bounded. Let I denote the index at which ρ_k is modified and let $I \leq k \leq K$ denote the iterates for which ρ_k remains fixed. It follows from the above reasoning that there exists N such that

$$\phi_I - \phi_K = \sum_{k=I}^{K} (\phi_k - \phi_{k+1}) \leq N, \tag{3.20}$$

where to simplify the notation we have used $\phi_k \equiv \phi_k(0)$.

From the termination condition for the linesearch (2.15a), (3.4) and (3.20), we also have

$$\tfrac{1}{2}\sigma\beta_{svH} \sum_{k=I}^{K} \alpha_k \|p_k\|^2 \leq \sum_{k=I}^{K} (\phi_k - \phi_{k+1}) \leq N. \tag{3.21}$$

This result implies that $\alpha_k\|p_k\|$ is bounded. Hence if $\|x_k\|$ is not bounded there must exist sets of iterates with indices, say $s_l < k \leq r_l$ for $l = 1, 2, \ldots$, such that $\|x_{s_l}\| \leq M$, $\|x_k\| > M$ for M large enough, $\lim_{l\to\infty} r_l = \infty$, and $\lim_{l\to\infty} \|x_{r_l}\| \to \infty$. It follows that if M is chosen so that $M > \max\{\|x_I\|\}$ then ρ_k is constant in the interval $s_l \leq k \leq r_l$. The existence of an index such that $\|x_{s_l}\| \leq M$ is assured since we have $\|x_I\| \leq M$ and at least one index in the interval for which $\|x_k\| > M$. From these assumptions and definitions it follows that

$$\sum_{k=s_l}^{r_l-1} \alpha_k\|p_k\| \geq \|x_{r_l} - x_{s_l}\| \to \infty. \tag{3.22}$$

It follows from Lemma 3.2 that $\|p_k\| > \epsilon$ for $s_l + 1 \leq k \leq r_l$. From (3.22) we get

$$\sum_{j=s_l}^{r_l-1} \alpha_j\|p_j\|^2 > \epsilon \sum_{j=s_l+1}^{r_l-1} \alpha_j\|p_j\| + \alpha_{s_l}\|p_{s_l}\|^2 \to \infty,$$

but this contradicts (3.21), implying that the points generated by the algorithm must lie on a compact set. □

To complete this section, we show that the search direction computed from the QP subproblem is bounded.

LEMMA 3.5. *Under the assumptions of Lemma* 3.4, *the sequence* $\{p_k\}$ *is bounded.*

Proof. We drop the subscript k in the proof.

As all the steps taken in the solution of the QP subproblem are descent steps, we have from (2.3),

$$\psi(p_0) \geq \psi(p) = g^T p + \tfrac{1}{2}p^T H p = \tfrac{1}{2}\|H^{\frac{1}{2}}p + H^{-\frac{1}{2}}g\|^2 - \tfrac{1}{2}g^T H^{-1} g,$$

implying from HC2 and $\|a\| \leq \|a+b\| + \|b\|$,

$$\sqrt{\beta_{svH}}\|p\| \leq \|H^{\frac{1}{2}}p\| \leq \|H^{-\frac{1}{2}}g\| + \|H^{\frac{1}{2}}p + H^{-\frac{1}{2}}g\| \leq \|H^{-\frac{1}{2}}g\| + \sqrt{2\psi(p_0) + g^T H^{-1} g}.$$

The boundedness of $\|p\|$ follows from this result, Lemma 3.4, conditions HC1 and HC2 and the bound (3.9). □

It is tacitly assumed in the remaining proofs that the Assumptions A1–A7 and conditions MC1, MC2, HC1, and HC2 hold.

3.4. The sequence of search directions $\{p_k\}$. In this section we relate the behavior of the sequence $\{x_k - x^*\}$, where x^* denotes a KKT point closest to x_k, to that of the sequence $\{p_k\}$. In particular, we show that $\|p_k\| \to 0$ implies $x_k \to x^*$, and so it is enough to prove that $\|p_k\| \to 0$ to establish global convergence.

Although the KKT point x^* introduced above may not be unique, the assumptions made on the problem, and more specifically Assumption A7, imply that if $\|x_k - x^*\|$ is sufficiently small then x^* is unique, as the following lemma shows. This result allows us to work with a well-defined sequence $\{x_k - x^*\}$, at least close to a KKT point; it will also imply that the limit point of the sequence generated by the algorithm is unique.

LEMMA 3.6. *The* KKT *points for problem* NP *are isolated.*

Proof. Assume that the result does not hold, and let x^* denote a KKT point for NP that is not isolated, that is, for any $\epsilon > 0$ there exists a KKT point $y_\epsilon \neq x^*$

satisfying $\|x^* - y_\epsilon\| < \epsilon$. Consequently, there exists a sequence $\{y_k\}$ such that y_k is a KKT point for all k, $y_k \neq x^*$ and $y_k \to x^*$.

For sufficiently small $\|x^* - y_k\|$ the active sets at y_k and x^* must be the same; otherwise we would have for some constraint j that $c_j(x^*) = 0$ with both $c_j(y_k) > 0$ and $(\lambda_k)_j = 0$ along some subsequence, where λ_k is the multiplier vector at y_k. From Assumptions A3 and A4 and (1.1) we have $\lambda_k \to \lambda^*$, the multiplier vector at x^*, but this would imply $c_j(x^*) = \lambda_j^* = 0$, contradicting Assumption A6.

Let Z_k denote a basis for the null-space of $\nabla \hat{c}(y_k)$, the Jacobian of the active constraints at y_k, and Z^* denote the corresponding basis at x^*. Among all possible bases, Z_k is selected to have continuous first derivatives in a ball around x^*. It follows from Assumption A4 and the fact the active set is constant that such bases exist.

For any element of the sequence y_k and for x^* we have from (1.1)

$$Z_k^T \nabla F(y_k) = 0 \qquad \text{and} \qquad Z^{*T} \nabla F(x^*) = 0.$$

The Taylor series expansion of $Z_k^T \nabla F(y_k)$ around x^* gives

$$\begin{aligned} 0 = Z_k^T \nabla F(y_k) &= Z_k^T (\nabla F(y_k) - \nabla c(y_k)^T \lambda^*) \\ &= Z^{*T} (\nabla F(x^*) - \nabla c(x^*)^T \lambda^*) + (\nabla Z(x^*)(\nabla F(x^*) - \nabla c(x^*)^T \lambda^*) \\ &\quad + Z^{*T} \nabla^2 L(x^*, \lambda^*))(y_k - x^*) + o(\|y_k - x^*\|), \end{aligned} \tag{3.23}$$

where $L(x, \lambda)$ is the Lagrangian function of NP. Using (1.1) in (3.23), and dividing by $\|y_k - x^*\|$ gives

$$Z^{*T} \nabla^2 L(x^*, \lambda^*) \delta_k = o(1), \qquad \text{where} \qquad \delta_k = \frac{y_k - x^*}{\|y_k - x^*\|}. \tag{3.24}$$

Let $\hat{c}$ denote the subset of constraints active at x^* and y_k. If ϵ is sufficiently small then δ_k satisfies

$$\hat{c}(y_k) = 0 = \nabla \hat{c}(x^*)(y_k - x^*) + o(\|y_k - x^*\|) \quad \Rightarrow \quad \nabla \hat{c}(x^*) \delta_k = o(1). \tag{3.25}$$

Finally, for any convergent subsequence of the bounded sequence $\{\delta_k\}$, with limit δ, we have from (3.24) and (3.25),

$$Z^{*T} \nabla^2 L(x^*, \lambda^*) \delta = 0, \qquad \nabla \hat{c}(x^*) \delta = 0,$$

contradicting Assumption A7. □

This result, together with Assumption A2, implies that the number of KKT points lying on any compact region is finite. The distinctness and finiteness of the KKT points implies the existence of $\epsilon^* > 0$ such that for any two KKT points, say x_1^* and x_2^*, we have $\|x_1^* - x_2^*\| > 2\epsilon^*$. It follows that if $\|x_k - x^*\| < \epsilon^*$, where x^* is a KKT point nearest to x_k, then x^* is unique.

The next result presents some properties of the QP multipliers that will be useful for the analysis of the convergence and rate of convergence of the algorithm.

LEMMA 3.7. *Given a sequence of iterates* $\{x_k\}$ *and the associated sequence of search directions* $\{p_k\}$ *such that* $x_k \to x^*$, *a* KKT *point for* NP *with multiplier vector* λ^* *and* $p_k \to 0$, *then*

$$\|\pi_k - \lambda^*\| \to 0,$$

where π_k are the QP *multipliers at the stationary point $\tilde{p}_k$. Furthermore,*

$$\|\pi_k - \lambda^*\| = O(\|\tilde{p}_k\|),$$

if $\|x_k - x^*\| \leq K\|\tilde{p}_k\|$ for some constant K.

Proof. We first show that for any constraint j such that $c_j(x^*) = \delta_1 > 0$ we must have $(\pi_k)_j = 0$ for large enough k. If $p_k \to 0$ it follows from (2.21) that $\tilde{p}_k \to 0$. Consequently, it follows from Assumption A3 that for k sufficiently large $\|\tilde{p}_k\| \leq \delta_1/(4\delta_2)$, where $\delta_2 = \|a_j^*\| > 0$. For k large enough we have

$$(a_k)_j^T \tilde{p}_k + (c_k)_j \geq \tfrac{1}{2}\delta_1 > 0,$$

implying that the multiplier for this constraint is zero.

Let $\hat{A}^*$ and $\hat{A}_k$ denote the corresponding Jacobian matrices restricted to the active set at x^* and let $\hat{\lambda}^*$ and $\hat{\pi}_k$ denote their respective multipliers. From (1.1) and (2.5a) we have

$$\begin{aligned} \hat{A}^{*T}\hat{\lambda}^* &= g^*, \\ \hat{A}_k^T\hat{\pi}_k &= g_k + H_k\tilde{p}_k, \end{aligned}$$

implying

$$\hat{A}^{*T}(\hat{\lambda}^* - \hat{\pi}_k) = g^* - g_k - H_k\tilde{p}_k - (\hat{A}^* - \hat{A}_k)^T\hat{\pi}_k. \tag{3.26}$$

From Assumption A4 that $\hat{A}^*$ has full rank and Assumption A3 it follows that $\hat{A}_k$ will also have full rank for large enough k, implying that $\hat{\pi}_k$ is bounded in norm, and these results together with (2.21), $p_k \to 0$ and HC1 yield $\pi_k \to \lambda^*$.

Using Taylor series expansions in (3.26), we obtain

$$\hat{A}^{*T}(\hat{\pi}_k - \hat{\lambda}^*) = \nabla^2 L(x^*, \hat{\pi}_k)(x_k - x^*) + H_k\tilde{p}_k + o(\|x_k - x^*\|), \tag{3.27}$$

where $L(x, \lambda)$ denotes the Lagrangian function for NP. The required result follows from (3.27), the condition we have imposed on the sequences $\{\tilde{p}_k\}$ and $\{x_k - x^*\}$, the boundedness of $\|\hat{\pi}_k\|$, Assumptions A3 and A4 and condition HC1. □

We now analyze the sequence of search directions $\{p_k\}$. The following result shows that as $p_k \to 0$ we get close to KKT points of NP and we only need to consider values p_k obtained as the minimizers for the corresponding subproblems. We complete this result by showing that a small value of $\|p_k\|$ also implies that the correct active set at x^* is identified, in the sense that the active QP constraints at p_k correspond to the active NP constraints at x^*.

LEMMA 3.8. *If along a subsequence $p_k \to 0$ then along this subsequence $\|x_k - x^*\| \to 0$, where x^* is a* KKT *point nearest to x_k. For k large enough, x^* is unique, p_k is the* QP *minimizer and the correct active set at x^* is identified.*

Proof. A subsequence such that $p_k \to 0$ exists if and only if a subsequence exists such that $p_k \to 0$ and the active set at p_k is constant. Let $\{r\}$ denote the sequence of indices for such a subsequence.

From the definition (2.21) of p_r it follows immediately that $A_r p_r + c_r \geq 0$. From $p_r \to 0$ and Assumption A3 it must hold that $c_r^- \to 0$ and $\tilde{p}_r \to 0$.

From (2.5) we have

$$A_r^T\pi_r - g_r - H_r\tilde{p}_r = 0 \quad \text{and} \quad \pi_r^T(A_r\tilde{p}_r + c_r) = 0. \tag{3.28}$$

Since $\tilde{p}_r \to 0$ it follows that

$$A_r^T \pi_r - g_r \to 0, \quad \pi_r^T c_r \to 0 \quad \text{and} \quad c_r^- \to 0. \tag{3.29}$$

We now show that for large enough r that p_r must have been computed as the minimizer for the QP. It follows from $p_r \to 0$ and $\|d_r\| = 1$ that either there exists K such that for all $r > K$ we have $\gamma_r = 0$ or $\gamma_r \to 0$ (see (2.26)). If we assume the latter it follows that

$$\min(\bar{\gamma}_r, \hat{\gamma}_r) \to 0.$$

(i) If $\bar{\gamma}_r \to 0$ along a subsequence, then from (2.27) along this subsequence we will have for some constraint j

$$\nabla c_j(x_r)^T(\tilde{p}_r + \bar{\gamma}_r d_r) + c_j(x_r) = 0 \quad \text{and} \quad (\pi_r)_j = 0,$$

where $(\pi_r)_j = 0$ follows from the fact that the QP constraint j is limiting the step, and so it cannot be active at $\tilde{p}_r$. These equations imply

$$c_j(x_r) \to 0 \quad \text{and} \quad (\pi_r)_j = 0,$$

contradicting Assumption A6.

(ii) If $\hat{\gamma}_r \to 0$ along a subsequence, then from (2.28),

$$\frac{\psi_r'(0)}{d_r^T H_r d_r} \to 0,$$

which implies from condition HC1 and $\|d_r\| = 1$ that $\psi_r'(0) = (H_r\tilde{p}_r + g_r)^T d_r \to 0$. If the condition number of $\hat{A}_i$ along the subsequence is bounded, condition (2.24) will hold and for some constraint j we have $(\pi_r)_j < 0$, $(\pi_r)_j \to 0$ and $\nabla c_j(x_r)^T\tilde{p}_r + c_j(x_r) = 0$, giving

$$c_j(x_r) \to 0 \quad \text{and} \quad (\pi_r)_j \to 0,$$

again contradicting Assumption A6. Otherwise, from Lemma 2.1 in the limit we have that $\nabla c(x^*)^T\lambda^* = \nabla F(x^*)$ with $\lambda^* \geq 0$, implying that x^* is a KKT point with a rank-deficient Jacobian matrix for the active constraints, violating Assumption A4.

We conclude therefore that $\gamma_r = 0$ for $r > K$ and this together with (3.28) implies p_r is the minimizer of the QP subproblem. For r large enough $\pi_r \geq 0$, which together with (3.29) and Assumption A3 implies $\|x_r - x^*\| \to 0$, where x^* is the nearest KKT point to x_r. For r large enough x^* is unique.

Finally, we prove that for r large enough the active set of the QP coincides with the active set of NP at x^*. First note that for r large enough the active set of the QP must be a subset of the constraints active at x^*, otherwise p_r is a step to a nonactive constraint implying $\|p_r\| > \epsilon > 0$. Assume that for the subsequence we have $\nabla c_j(x_r)p_r + c_j(x_r) > 0$ and $c_j(x^*) = 0$. From (2.5b) we must have $(\pi_r)_j = 0$, implying from Lemma 3.7 that $\lambda_j^* = 0$, but this violates Assumption A6, and for r large enough the correct active set is known. □

This result shows that there is an $\epsilon > 0$ such that if $\|p_k\| < \epsilon$, then p_k is the solution of the QP subproblem, and the correct active set is known.

We have just shown that if $p_k \to 0$ along a subsequence, then $x_k \to x^*$. To show $p_k \to 0$, we need a stronger result, giving a relationship between the rates of convergence of the sequences $\{x_k - x^*\}$ and $\{p_k\}$.

LEMMA 3.9. *If x^* denotes a* KKT *point closest to x_k, then there exists a constant M such that*

$$\|x_k - x^*\| \leq M\|p_k\|.$$

Proof. If $\|p_k\| > \epsilon$ for all k then the result holds trivially since $\|x_k\|$ and $\|x^*\|$ are both bounded. Again let $\{r\}$ denote the indices of a subsequence such that $p_r \to 0$ and the active set at p_r is constant. From Lemma 3.8, for this subsequence we have $\|x_r - x^*\| \to 0$. We assume for the rest of this proof that r is large enough so that x^* is unique, p_r is the minimizer of the QP and the correct active set has been identified.

Let $\hat{c}$, $\hat{A}$, and $\hat{\pi}$ denote the corresponding quantities restricted to the constraints in the active set. From Assumption A4 we know that $\hat{A}^*$ has full row rank, and we assume that r is large enough so that $\hat{A}_r$ also has full rank.

Let Z_r denote a basis for the null space of $\hat{A}_r$, with uniformly bounded norm and continuous first derivatives. From the optimality conditions for p_r, (2.5), we get

$$h(x) \equiv \begin{pmatrix} Z_r^T H_r \\ \hat{A}_r \end{pmatrix} p_r = -\begin{pmatrix} Z_r^T g_r \\ \hat{c}_r \end{pmatrix} = -\begin{pmatrix} Z_r^T (g_r - \hat{A}_r^T \hat{\lambda}^*) \\ \hat{c}_r \end{pmatrix}. \tag{3.30}$$

Since $h(x^*) = 0$, we have from the Taylor series expansion that

$$h_j(x_r) = S_j((\theta_r)_j)(x_r - x^*),$$

where $S_j((\theta_r)_j) = \nabla h_j(x^* + (\theta_r)_j(x_r - x^*))$ and $0 < (\theta_r)_j \leq 1$. We have therefore

$$\begin{pmatrix} Z_r^T g_r \\ \hat{c}_r \end{pmatrix} = -S(\theta_r)(x_r - x^*). \tag{3.31}$$

From (3.23) we get

$$S(0) = \begin{pmatrix} {Z^*}^T \nabla^2 L(x^*, \lambda^*) \\ \hat{A}(x^*) \end{pmatrix},$$

and Assumptions A4 and A7 imply that $S(0)$ is nonsingular. It follows that for sufficiently large values of r, $S(\theta_r)$ is also nonsingular. It then follows from (3.31) that for some positive constant M_1,

$$\|x_r - x^*\| \leq M_1(\|Z_r^T g_r\| + \|\hat{c}_r\|). \tag{3.32}$$

From Assumption A3, property HC1 and (3.30) it follows that

$$M_2\|p_r\| \geq \|Z_r^T g_r\| + \|\hat{c}_r\|, \tag{3.33}$$

for some positive constant M_2.

Since the subsequence $\{p_k\}$ such that $p_k \to 0$ is composed of a finite number of subsequences for which $p_r \to 0$ and the active set at p_r is constant, the required result follows from (3.32) and (3.33). □

3.5. Bounds on the penalty parameter. The conditions we have imposed on the algorithm (and more specifically on the multiplier estimate) are not sufficient to ensure that the penalty parameter is bounded. However, bounds on ρ_k are related to the behavior of different quantities in the algorithm, and in particular to $\|p_k\|$ and

$\|c_k - s_k\|$. The following lemmas introduce bounds on the size of ρ_k in terms of these quantities. We start by presenting the results for those iterations where the penalty parameter is modified, and then we extend the results to general iterations.

The notation k_l is used in all that follows to indicate iterations at which the value of the penalty parameter needs to be modified.

LEMMA 3.10. *For any iteration k_l in which the value of ρ is modified,*

$$\rho_{k_l}\|c_{k_l} - s_{k_l}\| \leq N \quad \text{and} \quad \rho_{k_l}\|p_{k_l}\|^2 \leq N,$$

for some constant N.

Proof. All quantities in the proof refer to iteration k_l, and so this subscript is dropped.

From the definition of $\hat{\rho}$, (2.13), and Lemma 3.3 we get

$$\begin{aligned}\hat{\rho}\|c - s\|^2 &= g^T p + \tfrac{1}{2}p^T H p + (2\lambda - \mu)^T(c - s) \\ &\leq -\beta_1 p^T H p + \beta_2\|c - s\| + (2\lambda - \mu)^T(c - s) \leq (\beta_2 + \|2\lambda - \mu\|)\|c - s\|,\end{aligned}$$

where β_1 and β_2 are positive constants. From (3.15) and the above result we obtain the first bound in the Lemma,

$$\hat{\rho}\|c - s\| \leq 3\beta_\mu + \beta_2. \tag{3.34}$$

If the penalty parameter needs to be modified, condition (2.11) cannot hold for $\tilde{\rho} \equiv \rho_{k_l - 1}$, and (3.3) implies

$$\phi'(0) = g^T p + (2\lambda - \mu)^T(c - s) - \tilde{\rho}\|c - s\|^2 > -\tfrac{1}{2}p^T H p.$$

It follows that

$$g^T p + \tfrac{1}{2}p^T H p + (2\lambda - \mu)^T(c - s) > 0. \tag{3.35}$$

Replacing in (3.35) the bound for $g^T p + \frac{1}{2}p^T H p$ given in Lemma 3.3 we obtain

$$(2\lambda - \mu)^T(c - s) + \beta_2\|c - s\| > \beta_1 p^T H p,$$

which together with Lemma 2.2 implies

$$\frac{3\beta_\mu + \beta_2}{\beta_1}\|c - s\| > p^T H p. \tag{3.36}$$

From condition HC2 we have $\|p\|^2 \leq (1/\beta_{svH})\, p^T H p$. If we multiply both sides of this inequality by $\hat{\rho}$ and use (3.36) to bound $p^T H p$, we obtain

$$\hat{\rho}\|p\|^2 \leq \hat{\rho}\,\frac{1}{\beta_{svH}}\, p^T H p \leq \frac{3\beta_\mu + \beta_2}{\beta_1 \beta_{svH}}\,\hat{\rho}\|c - s\| \leq \frac{(3\beta_\mu + \beta_2)^2}{\beta_1 \beta_{svH}},$$

where the last inequality follows from (3.34). The second desired bound then follows from $2\hat{\rho} \geq \rho$. □

We now extend these results to all iterations. To simplify notation, we shall use I and K to denote k_l and k_{l+1} respectively. Thus, the penalty parameter is increased at x_I and x_K in order to satisfy condition (2.11), and remains fixed at ρ_I for iterations $I, \ldots, K - 1$.

LEMMA 3.11. *There exists a constant M such that for all l,*

$$\rho_{k_l} \sum_{k=k_l}^{k_{l+1}-1} \|\alpha_k p_k\|^2 < M. \tag{3.37}$$

Proof. For $I \leq k \leq K-1$, property (2.15a) imposed by the choice of α_k, and the fact that the penalty parameter is not increased, imply that

$$\phi_k - \phi_{k+1} \geq -\sigma \alpha_k \phi'_k.$$

Summing these inequalities for $k = I$ to $K-1$, $0 \leq \alpha_k \leq 1$ together with (3.4) gives

$$\tfrac{1}{2}\sigma \beta_{svH} \sum_{k=I}^{K-1} \|\alpha_k p_k\|^2 \leq \phi_I - \phi_K. \tag{3.38}$$

Consider the term $\rho_I(\phi_I - \phi_K)$. From (2.1) and (2.2),

$$\rho\phi = \rho F - \rho\lambda^T(c-s) + \tfrac{1}{2}\rho^2\|c-s\|^2.$$

This equation, together with the boundedness of $\rho_I\|c_I - s_I\|$ and $\rho_I\|c_K - s_K\|$ (implied by $\rho_K > \rho_I$ and Lemma 3.10), and that of the multiplier estimates (Lemma 2.2), implies that for some $M_1 > 0$,

$$\rho_I(\phi_I - \phi_K) \leq M_1 + \rho_I(F_I - F_K). \tag{3.39}$$

Consider now iterations for which $\|p_I\| \leq \epsilon$, so that Lemma 3.8 applies and p_I has been obtained as the minimizer for the subproblem (for all other iterations Lemma 3.10 implies that ρ_I is bounded, and the result follows from Assumption A3, (3.39), and (3.38)).

Expanding F_K and c_K about x_I, we get

$$F_K - F_I = (x_K - x_I)^T g_I + O(\|x_I - x_K\|^2), \tag{3.40a}$$

$$c_K - c_I = A_I(x_K - x_I) + O(\|x_I - x_K\|^2). \tag{3.40b}$$

From Lemma 3.9 we have

$$\|x_I - x^*\| \leq M_p\|p_I\| \quad \text{and} \quad \|x_K - x^*\| \leq M_p\|p_K\|. \tag{3.41}$$

As p_I was obtained as the solution of the QP, condition (2.5a) must hold with multiplier vector $\pi_I \geq 0$. This condition together with (3.40aa), (3.40ab), and (3.41) implies

$$F_I - F_K = (c_I - c_K)^T\pi_I + O\Big(\max(\|p_I\|^2, \|p_K\|^2)\Big). \tag{3.42}$$

Using again (2.5),

$$c_I{}^T\pi_I = -p_I{}^T A_I{}^T\pi_I = -g_I{}^T p_I - p_I{}^T H_I p_I.$$

Since ρ is increased at iteration I, we must have that condition (2.11) cannot hold at that iteration, implying

$$\phi'_I(0) = g_I{}^T p_I + (2\lambda_I - \mu_I)^T(c_I - s_I) - \rho_{I-1}\|c_I - s_I\|^2 > -\tfrac{1}{2}p_I{}^T H_I p_I.$$

The previous two results imply

$$\rho_I {\pi_I}^T c_I < -\rho_I \tfrac{1}{2} {p_I}^T H_I p_I + \rho_I (2\lambda_I - \mu_I)^T (c_I - s_I) - \rho_I \rho_{I-1} \|c_I - s_I\|^2,$$

and this, together with the positive-definiteness of H_I (condition HC2), the boundedness of the multipliers (condition MC1 and Lemma 2.2) and Lemma 3.10, gives

$$\rho_I {c_I}^T \pi_I < \rho_I (2\lambda_I - \mu_I)^T (c_I - s_I) \leq M_2, \tag{3.43}$$

for some $M_2 > 0$.

Consider now the term ${c_K}^T \pi_I$ in (3.42). From $\pi_I \geq 0$ we must have

$$-\rho_I c_K^T \pi_I \leq \rho_I c_K^{-T} \pi_I$$

and from (2.9) we have $\|c_K^-\| \leq \|c_K - s_K\|$. Using $\rho_I < \rho_K$ and Lemma 3.10, we conclude that there exists a constant M_3 such that

$$-\rho_I {c_K}^T \pi_I < M_3. \tag{3.44}$$

Finally, consider the third term on the right-hand side of (3.42). It follows from Lemma 3.10 and the relation $\rho_I < \rho_K$ that there exists M_4 and M_5 such that

$$\rho_I \|p_I\|^2 < M_4 \quad \text{and} \quad \rho_I \|p_K\|^2 < M_5,$$

and hence for some constant M_6

$$\rho_I O\Big(\max(\|p_I\|^2, \|p_K\|^2)\Big) < M_6. \tag{3.45}$$

Combining (3.43), (3.44), and (3.45), we obtain the bound

$$\rho_I (F_I - F_K) < M_2 + M_3 + M_6,$$

which, together with (3.39) and (3.38) implies the desired result. □

LEMMA 3.12. *There exists a constant M such that, for all k,*

$$\rho_k \|c_k - s_k\| \leq M. \tag{3.46}$$

Proof. As in the preceding Lemma, let $I = k_l$ and $K = k_{l+1}$. From Lemma 3.10, (3.46) is immediate for $k = I$ and $k = K$.

To verify a bound for $k = I+1, \ldots, K-1$ we analyze some intermediate iterations k and $k+1$. We drop the iteration subscript; also let quantities evaluated at x_{k+1} be denoted with a tilde.

From (2.8), $\rho_I(\tilde{c}_j - \tilde{s}_j) = \min(\rho_I \tilde{c}_j, \tilde{\lambda}_j)$. Consider the following two cases:

(i) If $\rho_I \tilde{c}_j \geq -|\tilde{\lambda}_j|$, then

$$\rho_I |\tilde{c}_j - \tilde{s}_j| \leq |\tilde{\lambda}_j|. \tag{3.47}$$

(ii) Assume now that $\rho_I \tilde{c}_j < -|\tilde{\lambda}_j|$. Expanding the jth constraint function around x_k gives

$$\tilde{c}_j = c_j + \alpha a_j^T p + O(\|\alpha p\|^2).$$

Rewriting the previous expression, we obtain:

$$\tilde{c}_j = (1 - \alpha) c_j + \alpha (a_j^T p + c_j) + O(\|\alpha p\|^2). \tag{3.48}$$

Adding and subtracting $(1-\alpha)s_j$ on the right-hand side of (3.48) gives

$$\tilde{c}_j = (1-\alpha)(c_j - s_j) + (1-\alpha)s_j + \alpha(a_j^T p + c_j) + O(\|\alpha p\|^2). \tag{3.49}$$

Since s_j, $a_j^T p + c_j$, α and $1-\alpha$ are all nonnegative, we get

$$(1-\alpha)s_j + \alpha(a_j^T p + c_j) \geq 0,$$

and using this bound in (3.49) we obtain

$$\tilde{c}_j \geq (1-\alpha)(c_j - s_j) + O(\|\alpha p\|^2). \tag{3.50}$$

Since we assume $\rho_I \tilde{c}_j < -|\tilde{\lambda}_j|$ we have $\tilde{c}_j = \tilde{c}_j - \tilde{s}_j \leq 0$. Using this bound and $1-\alpha \leq 1$ in (3.50) we get the following inequality:

$$-\tilde{c}_j = |\tilde{c}_j| = |\tilde{c}_j - \tilde{s}_j| \leq -(1-\alpha)(c_j - s_j) + O(\|\alpha p\|^2) \leq |c_j - s_j| + O(\|\alpha p\|^2).$$

Multiplying both sides by ρ_I gives

$$\rho_I|\tilde{c}_j - \tilde{s}_j| \leq \rho_I|c_j - s_j| + \rho_I O(\|\alpha p\|^2). \tag{3.51}$$

For a given iteration $k \leq K-1$ and constraint j we have one of the following two situations.

(i) For some iteration l, $I < l \leq k$, $\rho_I(c_l)_j \geq -|(\lambda_l)_j|$. If we add (3.51) for iterations $r = l, \ldots, k-1$, and use (3.47), we get

$$\begin{aligned}\rho_I|(c_k)_j - (s_k)_j| &\leq \rho_I|(c_l)_j - (s_l)_j| + \rho_I O\left(\sum_{r=l}^{k-1} \|\alpha_r p_r\|^2\right) \\ &\leq |(\lambda_l)_j| + \rho_I O\left(\sum_{r=l}^{k-1} \|\alpha_r p_r\|^2\right).\end{aligned}$$

The boundedness of $\rho_I|(c_k)_j - (s_k)_j|$ then follows from Lemmas 2.2 and 3.11.

(ii) For all iterations l, $I < l \leq k$ we have $\rho_I(c_l)_j < -|(\lambda_l)_j|$. We add (3.51) for $r = I$ to $k-1$, to obtain

$$\rho_I|(c_k)_j - (s_k)_j| \leq \rho_I|(c_I)_j - (s_I)_j| + \rho_I O\left(\sum_{r=I}^{k-1} \|\alpha_r p_r\|^2\right),$$

and now the desired result follows from Lemmas 3.10 and 3.11. □

3.6. Boundedness of α_k. Given the result of Lemma 3.11, all that is left to establish the global convergence of the algorithm is to show that the steplength is bounded away from zero. As a consequence of the weak assumptions imposed on the multiplier estimate μ_k, it is not possible to show that such a bound exists. However, it can be proved that the bound does exist if there is no subsequence along which $\|p_k\| \to 0$. This is enough to prove convergence.

We first derive a bound on the norm of the second derivative along the linesearch.

LEMMA 3.13. *For $0 \leq \theta \leq \alpha_k$, there exists a positive constant N such that*

$$\phi_k''(\theta) \leq N.$$

Proof. We again drop the subscript k. From (3.2),

$$\nabla^2 L_A = \begin{pmatrix} \nabla^2 F - \sum_j \big(\lambda_j - \rho(c_j - s_j)\big)\nabla^2 c_j + \rho A^T A & -A^T & -\rho A^T \\ -A & 0 & I \\ -\rho A & I & \rho I \end{pmatrix}.$$

From the definition of ϕ, given in (2.2), we get

$$\begin{aligned} \phi''(\theta) = {} & p^T W p + \textstyle\sum_j \rho\big(c_j(\theta) - s_j(\theta)\big) p^T \nabla^2 c_j(\theta) p \\ & + \rho\big(A(\theta)p - q\big)^T\big(A(\theta)p - q\big) - 2\xi^T\big(A(\theta)p - q\big), \end{aligned} \tag{3.52}$$

where the argument θ denotes quantities evaluated at $x + \theta p$, except for $s(\theta) \equiv s + \theta q$ and

$$W \equiv \nabla^2 F(\theta) - \textstyle\sum_j (\lambda_j + \theta\xi_j)\nabla^2 c_j(\theta).$$

We now derive bounds on the terms on the right-hand side of (3.52). For the first term we can write

$$p^T W p \le N_1 \|p^2\| \le M_1, \tag{3.53}$$

for some constant M_1, using Assumption A3, the boundedness of $\|\lambda\|$ and $\|\xi\|$ (condition MC1 and Lemma 2.2), and the boundedness of $\|p\|$ (Lemma 3.5).

Expanding c_j in a Taylor series about x gives

$$c_j(\theta) = c_j(x) + \theta a_j(x)^T p + \tfrac{1}{2}\theta^2 p^T \nabla^2 c_j(\theta_j) p,$$

where $0 < \theta_j < \theta$. Using (2.10) and multiplying both sides by ρ gives

$$\rho\big(c_j(\theta) - (s_j(\theta)\big) = \rho\,(1-\theta)\big(c_j(x) - s_j\big) + \rho\,\tfrac{1}{2}\theta^2 p^T \nabla^2 c_j(\theta_j) p.$$

Lemma 3.12 implies that $\rho\big|c_j(x) - s_j\big|$ is bounded, Lemma 3.11 implies that $\rho\|\theta p\|^2$ is bounded for $\theta \le \alpha$, and Assumption A3 implies that $\|\nabla^2 c_j(\theta_j)\|$ is also bounded. Consequently,

$$\rho\,\big|\big(c_j(\theta) - s_j(\theta)\big)\big| \le N,$$

where N is a constant. This result and Lemma 3.5 imply the second term in (3.52) is also bounded, that is,

$$\sum_j \big|\rho\big(c_j(\theta) - s_j(\theta)\big) p^T \nabla^2 c_j(\theta) p\big| \le N_2 \|p\|^2 \le M_2, \tag{3.54}$$

where N_2 and M_2 are constants.

Consider now $\rho\|A(\theta)p - q\|^2$, the third term on the right-hand side of (3.52). Using Taylor series, we have

$$a_j(x + \theta p)^T p = a_j^T p + \theta p^T \nabla^2 c_j(\bar{\theta}_j) p, \tag{3.55}$$

where $0 < \bar{\theta}_j < \theta$. From (2.10) and Lemmas 3.11 and 3.12, we obtain

$$\rho\|A(\theta)p - q\|^2 \le M_3, \tag{3.56}$$

where M_3 is a constant.

From (3.55), (2.10), Assumption A3, and the boundedness of $\|\xi\|$ (Lemma 2.2), the final term on the right-hand side of (3.52) is also bounded,

$$\begin{aligned} -2\xi^T(A(\theta)p - q) &= -2\xi^T(Ap - q) + \sum_j \xi_j \theta p^T \nabla^2 c_j(\bar{\theta}_j)p \\ &\leq 2\xi^T(c - s) + N_4\|p\|^2 \leq M_4, \end{aligned} \tag{3.57}$$

where N_4 and M_4 are constants.

The desired bound follows from (3.52), (3.53), (3.54), (3.56), and (3.57). □

LEMMA 3.14. *For any $\epsilon > 0$, if $\|p_k\| \geq \epsilon$ there exists a value $\bar{\alpha}(\epsilon)$ such that $\alpha_k \geq \bar{\alpha}(\epsilon) > 0$, where α_k is the steplength computed by the algorithm.*

Proof. We drop the subscript k corresponding to the iteration number. We start by proving that $\hat{\alpha}$ (as defined in (2.14) and (2.15)) is bounded away from zero if $\|p\| > \epsilon$. If condition (2.14) is satisfied at a given iteration, then $\hat{\alpha} = 1$, trivially bounded away from zero. We assume therefore that $\hat{\alpha}$ is chosen to satisfy (2.15).

In the proof of Lemma 3.1 it was shown that the linesearch procedure was well defined, and in particular, that there exists a value $\hat{\alpha} \in (0, 1]$ satisfying (2.15) and such that condition (2.15b) is not satisfied for any value of $\alpha \in [0, \hat{\alpha})$; see (3.6), (3.8), and (3.7).

From the Taylor series expansion of ϕ' at $\hat{\alpha}$ we have

$$\phi'(\hat{\alpha}) = \phi'(0) + \hat{\alpha}\phi''(\theta),$$

where $0 < \theta < \hat{\alpha}$. Therefore, using (3.6) and noting that $\eta < 1$ and $\phi'(0) < 0$, we obtain

$$\hat{\alpha} = \frac{\phi'(\hat{\alpha}) - \phi'(0)}{\phi''(\theta)} = (1 - \eta)\frac{|\phi'(0)|}{\phi''(\theta)}. \tag{3.58}$$

(Since $\hat{\alpha} > 0$, θ must be such that $\phi''(\theta) > 0$.)

If $\|p\| \geq \epsilon$, then from (3.4) we have that $|\phi'(0)| \geq \frac{1}{2}\beta_{svH}\epsilon^2$, and from Lemma 3.13 we also have $\phi''(\theta) \leq N$, implying

$$\hat{\alpha} \geq \frac{\beta_{svH}}{2N}\epsilon^2.$$

If condition (2.16) is satisfied for $\hat{\alpha}$, then the previous bound holds for α. Otherwise, for some constraint j we must have $h_j(\hat{\alpha}) \equiv c_j(x + \hat{\alpha}p) + \beta_c < 0$ (using the notation introduced in Lemma 3.1). If $h_j(0) \geq \frac{1}{2}\beta_c > 0$, from the continuity of h there exists a value $\tilde{\alpha} < \hat{\alpha}$ such that $h_j(\tilde{\alpha}) = 0$ and $h_j(\alpha) \geq 0$ for all $\alpha \in [0, \tilde{\alpha}]$. From the mean-value theorem

$$\tilde{\alpha} = \frac{h_j(\tilde{\alpha}) - h_j(0)}{h_j'(\theta)} = \frac{h_j(0)}{|h_j'(\theta)|},$$

for some $\theta \in [0, \tilde{\alpha}]$. But as $|h_j'(\theta)| = |a_j(x + \theta p)^T p| \leq K$ for some $K > 0$ (from Assumption A3 and the boundedness of $\|p\|$, Lemma 3.5), we have

$$\tilde{\alpha} \geq \frac{\beta_c}{2K}. \tag{3.59}$$

If $h_j(0) \leq \frac{1}{2}\beta_c$, we must have from (2.4b),

$$h_j'(0) = a_j^T p \geq -c_j = \beta_c - h_j(0) \geq \tfrac{1}{2}\beta_c.$$

From $h_j(0) \geq 0$ and $h_j(\hat{\alpha}) < 0$ there must exist a value $\dot{\alpha} < \hat{\alpha}$ such that $h_j'(\dot{\alpha}) < 0$, implying the existence of $\tilde{\alpha} < \dot{\alpha}$ such that $h_j'(\tilde{\alpha}) = 0$ and $h_j'(\alpha) \geq 0$ for all $\alpha \in [0, \tilde{\alpha}]$ (also, $h_j(\alpha) \geq 0$ for all $\alpha \in [0, \tilde{\alpha}]$). From the mean-value theorem,

$$\tilde{\alpha} = \frac{h_j'(\tilde{\alpha}) - h_j'(0)}{h_j''(\theta)} = \frac{h_j'(0)}{|h_j''(\theta)|}$$

for some $\theta \in [0, \tilde{\alpha}]$. But $h_j'(0) \geq \frac{1}{2}\beta_c$, and $|h_j''(\theta)| = |p^T \nabla^2 c_j(x + \theta p)p| \leq \bar{K}$ for some $\bar{K} > 0$, from Assumption A3 and the boundedness of $\|p\|$, Lemma 3.5, implying again

$$\tilde{\alpha} \geq \frac{\beta_c}{2\bar{K}}. \tag{3.60}$$

The procedure to construct α will ensure that $\alpha \geq \frac{1}{2}\tilde{\alpha}$, and so the result presented in the lemma will hold. □

We can now prove the global convergence theorem for the algorithm.

THEOREM 3.15. *The sequence* $\{x_k\}$ *generated by the algorithm converges to a unique* KKT *point for* NP.

Proof. It follows from Lemma 3.9 that to prove $\|x_k - x^*\| \to 0$, it is sufficient to show

$$\lim_{k \to \infty} \|p_k\| \to 0. \tag{3.61}$$

If (3.61) is true then there exists K such that $\|x_k - x^*\| < \epsilon^*/2$ and $\|p_k\| < \epsilon^*$ for all $k > K$, where $2\epsilon^*$ is the minimum distance between two KKT points. It follows that x^* is unique for $k > K$ (the sequence converges to the unique KKT point nearest to x_K), otherwise it implies that for some $k > K$ that either $\|x_k - x^*\| > \epsilon^*/2$ or $\|p_k\| > \epsilon^*$. Consequently, to prove the theorem it is sufficient to show (3.61) is true.

If $\|p_k\| = 0$ for any k, the algorithm terminates and the theorem is true. Hence we assume that $\|p_k\| \neq 0$ for any k. If $p_k \not\to 0$, there must exist a subsequence $\{p_l\}$, and a positive constant ϵ, such that $\|p_l\| > \epsilon$ for all l. In this case, from Lemma 3.14 there will exist a uniform lower bound on α_l, $\alpha_l \geq \bar{\alpha} > 0$, but then

$$\rho_l \|\alpha_l p_l\| \geq \bar{\alpha}\epsilon\rho_l \to \infty,$$

contradicting the fact that $\rho_k \|\alpha_k p_k\|$ is bounded (Lemma 3.11).

In the bounded case, we know that there exists a value $\tilde{\rho}$ and an iteration index $\tilde{K}$ such that $\rho = \tilde{\rho}$ for all $k \geq \tilde{K}$. Again, the proof is by contradiction. Consider only indices l such that $l > \tilde{K}$. Every such iteration after $\tilde{K}$ must yield a strict decrease in the merit function because the termination condition for the linesearch (2.15a), together with the boundedness of the steplength (from Lemma 3.14 and $\|p_l\| > \epsilon$) and (3.4) imply

$$\phi_l(\alpha_l) - \phi_l(0) \leq \sigma\alpha_l\phi_l'(0) \leq -\tfrac{1}{2}\sigma\bar{\alpha}\beta_{svH}\|p_l\|^2 < 0.$$

The adjustment of the slack variables s in (2.7) can only lead to a further reduction in the merit function, as L_A is quadratic in s and the minimizer with respect to s_j

is given by $c_j - \lambda_j/\rho$. From the fact that the penalty parameter is not modified, for iterations from the subsequence we have

$$\phi(x_{l+1}) - \phi(x_l) \leq -\tfrac{1}{2}\sigma\bar{\alpha}\beta_{svH}\epsilon^2.$$

Therefore, since the merit function with $\rho = \tilde{\rho}$ decreases by at least a fixed quantity at every step in the subsequence, it must be unbounded below, contradicting (3.19). It follows that (3.61) must hold. □

Having established the global convergence of the algorithm, the next step is to show that the multiplier estimate $\lambda_k \to \lambda^*$. In order to prove this result, we need to strengthen our conditions on the multiplier estimate μ_k (if μ_k does not converge then λ_k will not converge either). Following is the additional condition.

MC3. $\|\mu_k - \lambda^*\| = O(\|x_k - x^*\|)$, where λ^* denotes any multiplier vector associated with a KKT point closest to x_k.

This condition requires that β_μ in condition MC1 be chosen so that

$$\beta_\mu \geq \|\lambda^*\|. \tag{3.62}$$

Estimates satisfying MC1, MC2, and MC3 may be obtained by computing a multiplier for the "active" constraints (say, least-squares estimates of least length), and expanding to the full multiplier space by augmenting this vector with zeros corresponding to the inactive constraints. If such an estimate does not satisfy MC1, then a suitable estimate may be determined by appropriate scaling. The multipliers at the stationary point of the QP also satisfy the three conditions. Note that if x^* is not unique then from Lemma 3.6, $\|x_k - x^*\| > \epsilon$ for some $\epsilon > 0$, and MC3 holds for any vector μ_k that is bounded.

We first show that under the stronger conditions on μ_k the steplength α_k is uniformly bounded away from zero.

LEMMA 3.16. *Under* MC3 *and all earlier assumptions and conditions,* $\alpha_k \geq \bar{\alpha} > 0$.

Proof. We again drop the subscript k. We first tighten the bound on $\phi''(\theta)$ given in Lemma 3.13. From (3.53) and (3.54), we have that the first two terms on the right-hand side of (3.52) are bounded by a multiple of $\|p\|^2$. For the remaining terms, from (3.55) and (2.10) we obtain

$$(\rho(A(\theta)p-q)-2\xi)^T(A(\theta)p-q) = \sum_j (\theta p^T \nabla c_j(\bar{\theta}_j)p - c_j + s_j - 2\xi_j)(\theta p^T \nabla c_j(\bar{\theta}_j)p - c_j + s_j).$$

Expanding this expression, and using Lemmas 3.11 and 3.12 to bound the terms $\rho(c_j - s_j)\theta p^T \nabla c_j(\bar{\theta}_j)p$ and $\rho\theta^2(p^T \nabla c_j(\bar{\theta}_j)p)^2$, we obtain

$$\rho\,\|A(\theta)p - q\|^2 - 2\xi^T\big(A(\theta)p - q\big) \leq \rho\,\|c - s\|^2 + 2\xi^T(c - s) + M\|p\|^2, \tag{3.63}$$

for some constant M.

Observe that from (3.3) and MC2,

$$\begin{aligned} \rho\,(c-s)^T(c-s) + 2\xi^T(c-s) &= -\phi'(0) + p^T g + \mu^T(c-s) \\ &= -\phi'(0) + p^T\big(g - A^T\mu\big) - \mu^T s. \end{aligned} \tag{3.64}$$

Using Taylor expansions and Lemma 3.9 it follows that

$$p^T\big(g - A^T\mu\big) = p^T\big(g^* - A^{*T}\mu\big) + O(\|p\|^2) = (\lambda^* - \mu)^T A^* p + O(\|p\|^2).$$

From this result and MC3 there exists a constant $\tilde{M}$ such that

$$p^T(g - A^T\mu) \leq \tilde{M}\|p\|^2. \tag{3.65}$$

From $\mu_k \to \lambda^*$, strict complementarity at a KKT point (Assumption A6), and the fact that the correct active set is identified for $\|p\|$ small enough (Lemma 3.8), we eventually have $\mu \geq 0$ and $\mu^T s \geq 0$. Consequently, it follows from (3.52), (3.53), (3.54), (3.63), (3.64), and (3.65) that

$$\phi_k''(\theta) \leq -\phi_k'(0) + N\|p_k\|^2$$

for some constant $N > 0$. This result and (3.4) can be used with (3.58) to imply that there exists a value $\hat{\alpha}$ satisfying (2.15) such that

$$\hat{\alpha} \geq (1-\eta)\frac{\beta_{svH}\|p^2\|}{(\beta_{svH}+2N)\|p^2\|} = (1-\eta)\frac{\beta_{svH}}{(\beta_{svH}+2N)} > 0.$$

The desired result then follows from an argument identical to that given in the final part of Lemma 3.14. □

This lemma also implies that the effort needed to compute the value for the steplength is uniformly bounded in the algorithm. We now establish the convergence of the multiplier estimate.

THEOREM 3.17. *Under* MC3 *and all other assumptions and conditions,*

$$\lim_{k\to\infty} \lambda_k = \lambda^*.$$

Proof. From (2.29),

$$\lambda_{k+1} = \sum_{j=0}^{k} \gamma_{jk}\mu_j, \tag{3.66}$$

where

$$\gamma_{kk} = \alpha_k', \qquad \gamma_{lk} = \alpha_l' \prod_{r=l+1}^{k} (1-\alpha_r'), \quad l < k, \tag{3.67}$$

with $\alpha_0' = 1$ and $\alpha_l' = \alpha_l$, $l \geq 1$. (This convention is used because of the special initial condition that $\lambda_0 = \mu_0$.) From Lemma 3.16 and (3.67), we observe that

$$0 < \bar{\alpha} \leq \alpha_l' \leq 1 \quad \text{for all } l, \tag{3.68a}$$

$$\sum_{l=0}^{k} \gamma_{lk} = 1, \tag{3.68b}$$

$$\gamma_{lk} \leq (1-\bar{\alpha})^{k-l}, \quad l < k. \tag{3.68c}$$

From condition MC3 we have

$$\mu_k = \lambda^* + M_k\delta_k t_k, \tag{3.69}$$

with $|M_k| \leq M$, $\delta_k = \|x_k - x^*\|$ and $\|t_k\| = 1$. From Theorem 3.15, for any $\epsilon > 0$ we can choose a value K_1 so that, for $k \geq K_1$,

$$|M_k\delta_k| \leq \tfrac{1}{2}\epsilon. \tag{3.70}$$

Given any $\epsilon > 0$, we can also define an iteration index K_2 with the following property:

$$(1 - \bar{\alpha})^k \leq \frac{\epsilon}{2(k+1)(1+2\beta_\mu)} \tag{3.71}$$

for $k \geq K_2 + 1$. Let $K = \max(K_1, K_2)$. Then, from (3.66) and (3.69), we have for $k \geq 2K$,

$$\lambda_{k+1} = \sum_{l=0}^{K} \gamma_{lk}\mu_l + \sum_{l=K+1}^{k} \gamma_{lk}(\lambda^* + M_l\delta_l t_l).$$

Hence it follows from (3.68b) that

$$\lambda_{k+1} - \lambda^* = \sum_{l=0}^{K} \gamma_{lk}(\mu_l - \lambda^*) + \sum_{l=K+1}^{k} \gamma_{lk} M_l \delta_l t_l.$$

From the bounds on $\|\mu_l\|$ (condition MC1), $\|t_l\|$, and (3.62), we obtain

$$\|\lambda_{k+1} - \lambda^*\| \leq 2\beta_\mu \sum_{l=0}^{K} \gamma_{lk} + \sum_{l=K+1}^{k} \gamma_{lk} |M_l \delta_l|. \tag{3.72}$$

Since we assume $k \geq 2K$, it follows from (3.68a) and (3.68c) that

$$\sum_{l=0}^{K} \gamma_{lk} \leq \sum_{l=0}^{K} (1-\bar{\alpha})^{k-l} \leq \sum_{l=0}^{K} (1-\bar{\alpha})^{2K-l} \leq (K+1)(1-\bar{\alpha})^K.$$

Using (3.71), we thus obtain the following bound for the first term on the right-hand side of (3.72):

$$2\beta_\mu \sum_{l=0}^{K} \gamma_{lk} \leq \tfrac{1}{2}\epsilon. \tag{3.73}$$

To bound the second term in (3.72), we use (3.68b) and (3.70):

$$\sum_{l=K+1}^{k} \gamma_{lk} |M_l \delta_l| \leq \tfrac{1}{2}\epsilon \sum_{l=K+1}^{k} \gamma_{lk} \leq \tfrac{1}{2}\epsilon. \tag{3.74}$$

Combining (3.72)–(3.74), we obtain the following result: given any $\epsilon > 0$, we can find K such that

$$\|\lambda_k - \lambda^*\| \leq \epsilon \quad \text{for} \quad k \geq 2K + 1,$$

which implies the desired result. □

4. Rate of convergence. In this section we shall show under additional assumptions on the multiplier estimate that the algorithm converges at a superlinear rate, independently of the asymptotic behavior of the penalty parameter.

Since $p_k \to 0$, we may assume without loss of generality that p_k has been obtained as the minimizer for the QP subproblem, and that the correct active set has been identified.

We again start by presenting an outline of the steps taken.

(i) Bounds on the rate of growth of the penalty parameter introduced in Lemmas 3.10, 3.11, and 3.12 are tightened.

- In Lemma 4.1 we prove that at all iterations at which ρ_k is increased (if we have an infinite sequence of such iterations)

$$\rho_k \|c_k - s_k\| \to 0 \quad \text{and} \quad \rho_k \|p_k\|^2 \to 0.$$

- In Lemmas 4.2 and 4.3 these results are extended to all iterations.

(ii) In Lemma 4.4 it is shown that $\mu_k^T s_k = 0$ for sufficiently large k.

(iii) Lemma 4.5 proves the superlinear convergence of the sequence $\{x_k + p_k - x^*\}$, under certain assumptions on H_k.

(iv) For k sufficiently large, $\alpha_k = 1$.

- Lemma 4.6 gives the relationship between the descent in one iteration $\phi_k(1) - \phi_k(0)$ and the initial derivative in the linesearch $\phi_k'(0)$.
- Theorem 4.7 shows that $\alpha_k = 1$ for all sufficiently large k, implying superlinear convergence.

(v) Finally, Theorem 4.8 shows that under an additional condition on the multipliers, the penalty parameter remains bounded.

The first two lemmas introduce refinements on the results presented in Lemmas 3.10, 3.11, and 3.12, and their proofs are based on the corresponding proofs for these lemmas.

LEMMA 4.1. *If $k_l \to \infty$, where k_l denotes an iteration at which the penalty parameter is increased, then*

$$\lim_{l\to\infty} \rho_{k_l} \|c_{k_l} - s_{k_l}\| = 0 \quad \textit{and} \quad \lim_{l\to\infty} \rho_{k_l} \|p_{k_l}\|^2 = 0.$$

Proof. We drop the subscript k_l in what follows.

Since p is the minimizer of QP, condition (2.5a) holds for a nonnegative vector π. From (2.4b) and (2.5a) we have $g^T p + \frac{1}{2} p^T H p = -\pi^T c$ and using this result in the definition of $\hat{\rho}$, (2.13),

$$\hat{\rho}\|c - s\|^2 = -\tfrac{1}{2} p^T H p + (2\lambda - \mu - \pi)^T (c - s) - \pi^T s \le \|2\lambda - \mu - \pi\| \|c - s\|.$$

From (2.12) we have $\rho \le 2\hat{\rho}$, and using Theorem 3.17, MC3, and Lemma 3.7 we obtain

$$\lim_{l\to\infty} \rho_{k_l} \|c_{k_l} - s_{k_l}\| \le 2 \lim_{l\to\infty} \|2\lambda_{k_l} - \mu_{k_l} - \pi_{k_l}\| = 0. \tag{4.1}$$

From (3.36) and (4.1) we have $\lim_{l\to\infty} \rho_{k_l} \|p_{k_l}\|^2 = 0$, completing the proof. □

LEMMA 4.2. *For general iterations k, $\lim_{k\to\infty} \rho_k \|p_k\|^2 = 0$.*

Proof. Define $I \equiv k_l$ and $K \equiv k_{l+1}$.

If ρ is bounded, the result follows from Theorem 3.15. If ρ is increased in an infinite number of iterations, from (3.38) and Lemma 3.14 we only need to show that $\phi_I - \phi_K \to 0$.

From the boundedness of $\|\lambda_k\|$ (Lemma 2.2), Lemma 4.1 and the fact that $\rho_I < \rho_K$, we have

$$\rho_I |\lambda_I{}^T (c_I - s_I)| \le 2\rho_I \|\lambda_I\| \, \|c_I - s_I\| \to 0,$$
$$\rho_I |\lambda_K{}^T (c_K - s_K)| \le 2\rho_K \|\lambda_K\| \, \|c_K - s_K\| \to 0.$$

We also have from Lemma 4.1,

$$\rho_I{}^2 \|c_I - s_I\|^2 \to 0, \qquad \rho_I{}^2 \|c_K - s_K\|^2 \to 0.$$

These results and the definition of ϕ, (2.2), imply

$$\rho_I(\phi_I - \phi_K) - \rho_I(F_I - F_K) \to 0. \tag{4.2}$$

We now analyze the asymptotic behavior of the term $\rho_I(F_I - F_K)$. We have

$$F_I - F_K = (c_I - c_K)^T\pi_I + O\Big(\max(\|p_I\|^2, \|p_K\|^2)\Big).$$

Using the same arguments as in the proof of Lemma 3.11, inequality (3.43) also holds in this case, and from (3.15),

$$\rho_I {\pi_I}^T c_I < \rho_I \|c_I - s_I\| \|2\lambda_I - \mu_I\| \leq 3\beta_\mu \rho_I \|c_I - s_I\|. \tag{4.3}$$

A second bound for this term can be obtained from $\pi_I \geq 0$ and $s_I \geq 0$, implying

$$\rho_I {\pi_I}^T c_I \geq \rho_I {\pi_I}^T (c_I - s_I) \geq -\rho_I \|\pi_I\| \|c_I - s_I\|. \tag{4.4}$$

Since $\|\pi_I\|$ is bounded, it follows from applying Lemma 4.1 to (4.3) and (4.4) that

$$\rho_I {\pi_I}^T c_I \to 0. \tag{4.5}$$

From (2.9), the boundedness of $\|\pi_I\|$ and Lemma 4.1,

$$-\rho_I {c_K}^T \pi_I \leq \rho_I {c_K}^{-T} \pi_I \leq \rho_I \|\pi_I\| \|c_K - s_K\| \to 0. \tag{4.6}$$

We can again use Lemma 4.1 to obtain

$$\rho_I O\Big(\max(\|p_I\|^2, \|p_K\|^2)\Big) \to 0. \tag{4.7}$$

From (3.42), (4.5), (4.6), and (4.7) we have that the sequence $\{\rho_I(F_I - F_K)\}$ is bounded above by a sequence that converges to zero. It then follows from $\phi_I - \phi_K \geq 0$ and (4.2) that $\rho_I(\phi_I - \phi_K) \to 0$ and the desired result follows from (3.38) and Lemma 3.16. □

LEMMA 4.3. *For general iterations k, $\lim_{k\to\infty} \rho_k \|c_k - s_k\| = 0$.*

Proof. If ρ is bounded the result follows from $c^* \geq 0$, $\lambda^* \geq 0$, ${\lambda^*}^T c^* = 0$, Theorems 3.15 and 3.17 and (2.8).

We assume therefore that ρ is increased an infinite number of times. Consider two cases.

Case 1. If constraint j is such that $c_j^* > 0$, then $\lambda_j^* = 0$ and from (2.8),

$$\rho|c_j - s_j| = |\min(\rho c_j, \lambda_j)|,$$

but from Theorem 3.17 and Assumptions A3 and A6, eventually $\lambda_j < \rho c_j$, implying

$$\rho|c_j - s_j| = |\lambda_j| \to 0.$$

Case 2. For those j such that $c_j^* = 0$, implying $\lambda_j^* > 0$, consider iteration indices large enough that the correct active set is identified (Lemma 3.8), implying $a_j^T p + c_j = 0$. From the Taylor series expansion for c_j and the boundedness of the steplength,

$$c_j(x_k + \alpha_k p_k) = c_j(x_k) + \alpha_k (a_k)_j^T p_k + O(\|\alpha_k p_k\|^2) = (1 - \alpha_k) c_j(x_k) + O(\|p_k\|^2).$$

Recurring this relationship for k, $I < k < K$, we get

$$\rho_k(c_k)_j = \rho_I(c_k)_j = \rho_I \prod_{l=I}^{k-1}(1-\alpha_l)(c_I)_j + \rho_I O\left(\sum_{l=I}^{k-1}\|p_l\|^2\right),$$

but as $0 < \alpha_l \leq 1$ we must have

$$\rho_k|(c_k)_j| \leq \rho_I|(c_I)_j| + \rho_I O\left(\sum_{l=I}^{k-1}\|p_l\|^2\right). \tag{4.8}$$

From $c_j^* = 0$, Assumptions A3 and A6, and (2.8), eventually it must hold that $\rho_I|(c_I)_j - (s_I)_j| = \rho_I|c(_I)_j|$, and using Lemma 4.1, (4.8), and Lemma 4.2,

$$\rho_k|(c_k)_j| \to 0.$$

From this result, definition (2.8), Assumptions A3 and A6, and Theorem 3.17, for k large enough

$$\rho_k|(c_k)_j - (s_k)_j| = |\min(\rho_k(c_k)_j, (\lambda_k)_j)| = |\rho_k(c_k)_j| \to 0.$$

This completes the proof. □

LEMMA 4.4. *For k large enough $\mu_k^T s_k = 0$.*

Proof. If constraint j is such that $c_j^* > 0$, then for k large enough $(c_k)_j \geq \epsilon > 0$, and $(a_k)_j^T p_k + (c_k)_j \geq \frac{1}{2}\epsilon > 0$. It therefore follows from MC2 that $(\mu_k)_j = 0$.

If j is such that $c_j^* = 0$, then from Assumption A6, $\lambda_j^* > 0$. Also, from Lemma 4.3, $\rho_k((c_k)_j - (s_k)_j) = \min(\rho_k(c_k)_j, (\lambda_k)_j) \to 0$, and for large enough k Theorem 3.17 will imply $\rho_k(c_k)_j \leq (\lambda_k)_j$; these two results and definition (2.7) imply

$$(s_k)_j = \max\left(0, (c_k)_j - \frac{(\lambda_k)_j}{\rho_k}\right) = 0,$$

completing the result. □

To prove that the algorithm converges superlinearly it is necessary to assume that H_k converges to an approximation of $\nabla_{xx}^2 L(x^*, \lambda^*)$ in some sense, where $L(x, \lambda)$ denotes the Lagrangian function for problem NP.

Define W_k as

$$W_k \equiv \nabla_{xx}^2 L(x_k, \lambda_k) = \nabla_{xx}^2 F(x_k) - \sum_j (\lambda_k)_j \nabla_{xx}^2 c_j(x_k). \tag{4.9}$$

We impose the following additional condition on H_k.

HC3. Following Boggs, Tolle, and Wang [3], we assume

$$\|Z_k^T(H_k - W_k)p_k\| = o(\|p_k\|),$$

where Z_k is a basis for the null space of $\hat{A}_k$, the Jacobian of x_k of those constraints active at x^*, that is bounded in norm and has its smallest singular value bounded away from 0.

The proof proceeds by first showing that the sequence $\{x_k + p_k - x^*\}$ converges superlinearly, and then proving that a steplength of one is eventually attained.

The following lemma corresponds to Theorem 3.1 in [3].

LEMMA 4.5. *Under Assumptions* A1–A7, *and conditions* MC1–MC3, HC1–HC3,

$$\|x_k + p_k - x^*\| = o(\|x_k - x^*\|). \tag{4.10}$$

The results presented on bounds for the growth rate of the penalty parameter allow us to obtain an asymptotic expansion for the quantities involved in the line-search termination criterion. We want to prove that condition (2.14) is satisfied for k sufficiently large. It is shown in the following lemma that the satisfaction of (2.14) is directly related to the asymptotic properties of $T_k \equiv p_k^T(g_k - A_k^T\mu_k) + p_k^T W_k p_k$.

LEMMA 4.6. *The following relationship holds:*

$$\phi_k(1) - \phi_k(0) = \tfrac{1}{2}\phi_k'(0) + \tfrac{1}{2}T_k + o(\|p_k\|^2).$$

Proof. In the proof we drop the subscript k, and we denote quantities associated with $x_k + p_k$ by a tilde, that is, $\tilde{F} \equiv F(x_k + p_k)$ while $F \equiv F(x_k)$.

From the definition of the merit function (2.2) and (2.1) we have

$$\begin{aligned}\phi(1) - \phi(0) = \tilde{F} - F - \mu^T(\tilde{c} - s - q) + \lambda^T(c - s)\\ + \frac{\rho}{2}(\tilde{c} - s - q)^T(\tilde{c} - s - q) - \frac{\rho}{2}(c - s)^T(c - s).\end{aligned} \tag{4.11}$$

From the Taylor series expansion of c around x and (2.10) we have

$$\tilde{c}_j - s_j - q_j = \tilde{c}_j - c_j - a_j^T p = \tfrac{1}{2}p^T\nabla^2 c_j p + o(\|p\|^2),$$

and using this result with the Taylor expansions for c and F in (4.11) we obtain

$$\begin{aligned}\phi(1) - \phi(0) = g^T p + \tfrac{1}{2}p^T\nabla^2 F p - \tfrac{1}{2}\textstyle\sum_j \mu_j\, p^T\nabla^2 c_j p + \lambda^T(c - s)\\ + \frac{\rho}{8}\textstyle\sum_j (p^T\nabla^2 c_j p)^2 - \frac{\rho}{2}(c - s)^T(c - s) + o(\|p\|^2).\end{aligned} \tag{4.12}$$

From (2.6), condition MC3 and Theorem 3.17 we have

$$\mu = \lambda + \xi = \lambda + o(1). \tag{4.13}$$

Also, from Lemma 4.2 and Assumption A3 we have

$$\rho p^T\nabla^2 c_j p = o(1) \quad \text{and} \quad \rho(p^T\nabla^2 c_j p)^2 = o(\|p\|^2).$$

Replacing these results in (4.12) and reordering the terms we obtain

$$\begin{aligned}\phi(1) - \phi(0) = g^T p + \tfrac{1}{2}p^T\nabla^2 F p - \tfrac{1}{2}\textstyle\sum_j \lambda_j\, p^T\nabla^2 c_j p + \tfrac{1}{2}(2\lambda - \mu)^T(c - s)\\ + \tfrac{1}{2}\mu^T(c - s) - \frac{\rho}{2}(c - s)^T(c - s) + o(\|p\|^2).\end{aligned}$$

Using (4.9) and (3.3) to simplify this expression,

$$\phi(1) - \phi(0) = \tfrac{1}{2}\phi'(0) + \tfrac{1}{2}\Big(g^T p + p^T W p + \mu^T(c - s)\Big) + o(\|p\|^2). \tag{4.14}$$

Finally, from condition MC2 we have $\mu^T c = -\mu^T A p$, and from Lemma 4.4 we know that eventually $\mu^T s = 0$, implying in particular that $\mu^T s = o(\|p\|^2)$, and replacing these bounds in (4.14) we have

$$\phi(1) - \phi(0) = \tfrac{1}{2}\phi'(0) + \tfrac{1}{2}\Big(p^T W p + p^T(g - A^T\mu)\Big) + o(\|p\|^2),$$

completing the result. □

The main result of this section is given in the next theorem. It is shown that, if condition MC3 is replaced by a stronger condition, then after a finite number of iterations a steplength of one is taken for all iterations thereafter, implying that the algorithm achieves superlinear convergence. The new condition is

MC3'. $\|\mu_k - \lambda^*\| = o(\|x_k - x^*\|)$.

It is possible to prove superlinear convergence without the need to strengthen the conditions on the multipliers. It is shown in [29] that there exists a constant M such that if $\rho_k > M$, condition MC3 is sufficient.

THEOREM 4.7. *If* MC3' *and all other assumptions and conditions hold then eventually a unit step is always taken and the algorithm converges superlinearly.*

Proof. As in Powell and Yuan [28], observe that the continuity of second derivatives gives the following relationships:

$$F(x_k + p_k) = F(x_k) + \tfrac{1}{2}\Big(g(x_k) + g(x_k + p_k)\Big)^T p_k + o(\|p_k\|^2), \tag{4.15}$$
$$c(x_k + p_k) = c(x_k) + \tfrac{1}{2}\Big(A(x_k) + A(x_k + p_k)\Big)p_k + o(\|p_k\|^2).$$

From the Taylor series expansions we have

$$F(x_k + p_k) = F(x_k) + g(x_k)^T p_k + \tfrac{1}{2}p_k^T \nabla^2 F(x_k) p_k + o(\|p_k\|^2), \tag{4.16}$$
$$c_j(x_k + p_k) = c_j(x_k) + a_j(x_k)^T p_k + \tfrac{1}{2}p_k^T \nabla^2 c_j(x_k) p_k + o(\|p_k\|^2),$$

and since (4.10) and Lemma 3.9 imply $g(x_k + p_k) = g^* + o(\|p_k\|)$, $a_j(x_k + p_k) = a_j^* + o(\|p_k\|)$, we get from (4.15) and (4.16) that (we drop the subscript k)

$$p^T \nabla^2 F p = (g^* - g)^T p + o(\|p\|^2), \tag{4.17a}$$
$$p^T \nabla^2 c_j p = (a_j^* - a_j)^T p + o(\|p\|^2). \tag{4.17b}$$

Condition MC3, Theorem 3.17, and (4.13) give $\sum_j \lambda_j\, p^T \nabla^2 c_j p = \sum_j \mu_j\, p^T \nabla^2 c_j p + o(\|p\|^2)$, and if we apply this bound to the result of adding (4.17a) to (4.17b) multiplied by λ_j, we have

$$p^T W p = p^T(g^* - A^{*^T}\mu) - p^T(g - A^T\mu) + o(\|p\|^2). \tag{4.18}$$

Condition MC3', (1.1), and Lemma 3.9 imply

$$p^T(g^* - A^{*^T}\mu) = p^T A^{*^T}(\lambda^* - \mu) = o(\|p\|^2),$$

and from (4.18),

$$T = p^T W p + p^T(g - A^T\mu) = p^T(g^* - A^{*^T}\mu) + o(\|p\|^2) = o(\|p\|^2). \tag{4.19}$$

From Lemma 4.6 and (4.19) we get

$$\phi(1) - \phi(0) = \tfrac{1}{2}\phi'(0) + o(\|p\|^2).$$

Since $\phi'(0) < 0$, the above relationship and Theorem 3.15 imply that condition (2.14) is eventually satisfied for k sufficiently large.

Regarding condition (2.16), we can use Taylor series expansions for c_j to write

$$c_j(x_k + p_k) = c_j(x_k) + a_j(x_k + \theta_j p_k)^T p_k \tag{4.20}$$

for some $\theta_j \in [0, 1]$, and

$$a_j(x_k + \theta_j p_k)^T p_k = a_j(x_k)^T p_k + p_k^T \nabla^2 c_j(x_k + \bar{\theta}_j p_k) p_k, \tag{4.21}$$

for $\bar{\theta}_j \in [0, \theta_j]$.

Using Theorem 3.15 and the boundedness of $\|\nabla^2 c_j(x_k + \bar{\theta}_j p_k)\|$ (from Assumption A3 and Lemma 3.4) in (4.21), for k large enough

$$a_j(x_k + \theta_j p_k)^T p_k \geq a_j(x_k)^T p_k - \tfrac{1}{2}\beta_c,$$

and from (2.4b),

$$a_j(x_k + \theta_j p_k)^T p_k \geq a_j(x_k)^T p_k - \tfrac{1}{2}\beta_c \geq -c_j(x_k) - \tfrac{1}{2}\beta_c.$$

Replacing this bound in (4.20), we obtain for all k large enough $c(x_k + p_k) \geq -\frac{1}{2}\beta_c e$, and condition (2.16) will also be satisfied, giving $x_{k+1} = x_k + p_k$. The required result then follows from Lemma 4.5. □

4.1. Boundedness of the penalty parameter. The last result in this section shows that, if condition MC3' is replaced by a slightly stronger condition, the penalty parameter needs to be modified in at most a finite number of iterations (and consequently it remains bounded). The criterion presented will be satisfied, for example, by the least-squares multipliers computed at $x_k + p_k$.

THEOREM 4.8. *If the multiplier estimates μ_k in the algorithm satisfy*

$$\|\mu_k - \lambda^*\| = O(\|x_k + p_k - x^*\|), \tag{4.22}$$

and all other assumptions and conditions hold then there exists a constant M such that $\rho_k \leq M$ for all k.

Proof. We may assume k large enough so that $\alpha_k = 1$. From (2.5), (2.4b), and $\pi_k^T s_k \geq 0$, we have

$$g_k^T p_k + p_k^T H_k p_k = p_k^T A_k \pi_k = -c_k^T \pi_k \leq -(c_k - s_k)^T \pi_k, \tag{4.23}$$

where π_k denotes the QP multipliers at iteration k. From (3.3), (4.23), and the fact that a unit steplength is accepted, it follows that

$$\phi_k'(0) \leq -p_k^T H_k p_k + \|2\mu_{k-1} - \mu_k - \pi_k\| \|c_k - s_k\| - \rho_k \|c_k - s_k\|^2. \tag{4.24}$$

From (4.22), HC2, and Lemmas 3.9, 3.8, and 3.7 we must have

$$\|2\mu_{k-1} - \mu_k - \pi_k\| \leq M_1 \|p_k\| \leq M_2 \sqrt{p_k^T H_k p_k}$$

for some positive constants M_1, M_2. It then follows using $a^2 + b^2 \geq 2ab$ that

$$\|2\mu_{k-1} - \mu_k - \pi_k\| \|c_k - s_k\| \leq M_2 \sqrt{p_k^T H_k p_k} \|c_k - s_k\| \leq \tfrac{1}{2} p_k^T H_k p_k + \tfrac{1}{2} M_2^2 \|c_k - s_k\|^2,$$

implying from (4.24) that

$$\phi_k'(0) \leq -\tfrac{1}{2} p_k^T H_k p_k + (\tfrac{1}{2} M_2^2 - \rho_k) \|c_k - s_k\|^2.$$

From this inequality it follows that if $\rho_k \geq \frac{1}{2} M_2^2$, condition (2.11) will be satisfied, and the penalty parameter will not be increased. Given that we are using the rule (2.12) for updating ρ_k, it must hold that $\rho_k \leq M_2^2$. □

5. Other merit functions. Several merit functions have been proposed and analyzed in the literature (a review can be found in Powell [27]). The question arises if the convergence results using early termination in the solution of the QP subproblem depend on our specific merit function, or if they are fairly independent of this choice. We shall show in this section that the choice of merit function is not critical. What we present is how to adapt our SQP algorithm to the use of other merit functions rather than examine other methods explicitly to see if the particular QP subproblem posed and the manner the search is performed can be adapted to the use of an incomplete solution. For example, we still perform a search in the x and λ spaces. Slack variables do not appear in the merit functions we shall consider, consequently the search in the space of the slack variables is no longer required.

We have selected as examples the study of two particular merit functions. The first one corresponds to a class of merit functions that includes among others the ℓ_1 merit function analyzed in Han [21], Byrd and Nocedal [5], and Burke and Han [4]. This general merit function takes the form:

$$\phi(x,\lambda) = F(x) + \lambda^T c^-(x) + \rho\|c^-(x)\|_p, \tag{5.1}$$

where an ℓ_p-norm ($1 \le p \le \infty$) is used, and $c_j^-(x) \equiv \max(0, -c_j(x))$. Again, we will omit the subscript if we refer to the ℓ_2-norm, and we will explicitly include it whenever we refer to a general ℓ_p-norm.

The second merit function we consider is

$$\phi(x,\lambda) = F(x) + \lambda^T c^-(x) + \tfrac{1}{2}\rho\|c^-(x)\|^2, \tag{5.2}$$

where we use the ℓ_2-norm. This merit function has been studied among others by Powell and Yuan [28] (applied to the equality-constrained problems only) and Schittkowski [32]. Unlike either of these algorithms, where the multiplier estimate λ was treated as a function of the iterate $\lambda(x)$, we do not explicitly define the form of the multiplier estimates although the ones used in both methods satisfy the criteria MC1, MC2, and MC3. Indeed the one used in [28] also satisfies MC3'.

We still assume A1–A7 hold for the problem. However, when the merit function (5.1) is used, the multiplier estimate μ_k is only required to satisfy MC1. This condition is trivial to satisfy. For example, we may choose $\lambda_0 = 0$ and $\mu_k = 0$ making the search in the multiplier space void. Such a choice reduces (5.1) to the well-known ℓ_1 merit function and our algorithm becomes very similar to that analyzed in [21]. When (5.2) is used, we assume conditions MC1 and MC2 hold. We have also assumed in the proofs that $\lambda_0 \ge 0$ and $\mu_k \ge 0$. We omit the proofs that the iterates lie on a compact set. For the first merit function (5.1) this proof is relatively straightforward, since it will be shown that the penalty parameter is bounded. The proof for the second merit function (5.2) is very similar to that for the Augmented Lagrangian merit function.

The criteria (2.15) for the choice of steplength α_k assume the merit function has continuous first derivatives. This property does not necessarily hold for the merit functions under consideration. Therefore we use the following criteria for determining a value α_k.

Define

$$\Delta_k \equiv g_k^T p_k + (\xi_k - \lambda_k)^T c^-(x_k) - \rho_k\|c^-(x_k)\|_p. \tag{5.3}$$

We start by selecting a value $\hat{\alpha}_k$ satisfying

$$\phi_k(\hat{\alpha}_k) \equiv \phi(x_k + \hat{\alpha}_k p_k, \lambda_k + \hat{\alpha}_k \xi_k) \le \phi_k(0) + \eta\hat{\alpha}_k\Delta_k, \tag{5.4}$$

and either

$$\hat{\alpha}_k \geq \gamma_l > 0 \tag{5.5}$$

or

$$\hat{\alpha}_k > \gamma_u \bar{\alpha}_k \qquad \text{and} \qquad \phi_k(\bar{\alpha}_k) > \phi_k(0) + \sigma \bar{\alpha}_k \Delta_k, \tag{5.6}$$

where $0 < \gamma_l < \gamma_u < 1$, $0 < \eta \leq \sigma < 1$ and $\bar{\alpha}_k > 0$. For a discussion of these criteria and the existence of $\hat{\alpha}_k$ see Calamai and Moré [6].

In addition to these conditions, we also also want to limit the size of the infeasibilities. If $\hat{\alpha}_k$ satisfies condition (2.16), then we let $\alpha_k = \hat{\alpha}_k$. Otherwise, we compute α_k by performing a backtracking linesearch from $\hat{\alpha}_k$ until conditions (5.4) and (2.16) are both satisfied.

Our preference for the criteria given in §2 is based on our belief that in practice they lead to a better choice of α_k. In the definition of our algorithm we could have used other steplength criteria without impacting the convergence properties.

The following basic relationships will be used to establish the convergence results,

$$c_j^-(x + \alpha p) \leq |c_j(x + \alpha p) - c_j(x) - \alpha a_j^T p| - \min(0, c_j(x) + \alpha a_j^T p) \tag{5.7a}$$

$$- \min(0, c_j(x) + \alpha a_j^T p) \leq (1 - \alpha) c_j^-(x), \tag{5.7b}$$

$$-\omega^T A p \leq -\|c^-(x)\|_p, \tag{5.7c}$$

$$-\Omega A p \leq -c^-(x). \tag{5.7d}$$

In these inequalities $A \equiv \nabla c(x)$. Also, Ω is a diagonal matrix such that $-\Omega A p$ is an element of the subdifferential of $c^-(x + \alpha p)$ at $\alpha = 0$. The diagonal entries of Ω take values in $[0, 1]$, are zero whenever $c_j(x) > 0$ and take the value one whenever $c_j(x) < 0$. Finally, $\omega^T A p$ represents an element of $\partial\varphi(0)$, the subdifferential of $\varphi(\alpha) \equiv \|c^-(x + \alpha p)\|_p$ at 0. The elements of ω are given by

$$\omega_j = (\Omega)_{jj} \left(\frac{c_j^-}{\|c^-\|_p} \right)^{p-1},$$

and have the property that $\omega^T c(x) = -\|c^-(x)\|_p$.

Consider now the case when ϕ has been defined from (5.1). From our assumption that $\lambda_k \geq 0$ and (2.4b),

$$\lambda_k^T \Omega_k (A_k p_k + c_k) \geq 0$$

for all k. It follows from this inequality and the relationships given in (5.7) that

$$\phi_k'(0) = g_k^T p_k + \xi_k^T c^-(x_k) - \lambda_k^T \Omega_k A_k p_k - \rho_k \omega_k^T A_k p_k \leq \Delta_k.$$

We select ρ_k such that

$$\Delta_k \leq -\tfrac{1}{2} p_k^T H_k p_k. \tag{5.8}$$

This rule is analogous to the ones used in Byrd and Nocedal [5], and Burke and Han [4].

The first step is to establish that such a value of ρ exists. From (3.14) and (5.3) we have

$$\Delta_k \leq -(\tfrac{1}{2} + \beta_1) p_k^T H_k p_k + \beta_2 \|c_k^-\| - (\xi_k - \lambda_k)^T c_k^- - \rho \|c_k^-\|_p. \tag{5.9}$$

If we now use (2.6), property MC1, and Lemma 2.2 to bound the multiplier term

$$(\xi_k - \lambda_k)^T c_k^- \leq \|\mu_k - 2\lambda_k\| \|c_k^-\| \leq 3\sqrt{m}\beta_\mu \|c_k^-\|_p,$$

where we have used $\|u\|_2 \leq \sqrt{m}\|u\|_p$, we obtain in (5.9)

$$\Delta_k \leq -(\tfrac{1}{2} + \beta_1) p_k^T H_k p_k + (\sqrt{m}\beta_2 + 3\sqrt{m}\beta_\mu - \rho)\|c_k^-\|_p.$$

Defining $\rho_u \equiv \sqrt{m}(\beta_2 + 3\beta_\mu)$, for any value $\rho \geq \rho_u$ condition (5.8) is satisfied for any k. This result also shows that the value of ρ will remain bounded in the algorithm.

THEOREM 5.1. *The algorithm modified to use the merit function* (5.1) *converges globally.*

Proof. Given the bound in Lemma 3.9, it suffices to show that $\|p_k\| \to 0$.

As ρ cannot grow without bound, any strategy for increasing ρ by a finite quantity whenever it is required to increase ρ implies that there exists an iteration value K such that $\rho_k = \rho_K$ for all $k \geq K$. We consider only iterations of this form. For $k \geq K$, from (5.4), (5.8) and condition MC2,

$$\phi(\alpha_k) - \phi(\alpha_{k-1}) \leq \alpha_k \eta \Delta_k \leq -\eta \beta_{svH} \alpha_k \|p_k\|^2.$$

From the boundedness of ϕ (Assumption A3), it follows that

$$\alpha_k \|p_k\|^2 \to 0. \tag{5.10}$$

If $\|p_k\| \to 0$, convergence follows from Lemma 3.9. Otherwise, if for a subsequence $\|p_k\| > \epsilon$, from (5.10) we must have $\alpha_k \to 0$ along the subsequence, and from the termination conditions for the linesearch (5.4), (5.5), and (5.6), $\bar{\alpha}_k \to 0$, as the step required to satisfy condition (2.16) is uniformly bounded away from zero (see (3.59) and (3.60)). Finally, from (5.6) we must also have $\hat{\alpha}_k \to 0$.

In the following relationships we drop the subscript k corresponding to the iteration number, and we denote by a tilde the value of functions evaluated at $x + \bar{\alpha}p$ (i.e., $\tilde{c} \equiv c(x + \bar{\alpha}p)$).

From the definition of the merit function (5.1),

$$\begin{aligned}\phi(\bar{\alpha}) - \phi(0) = {} & \bar{\alpha} g^T p + \lambda^T(\tilde{c}^- - c^-) + \bar{\alpha}\,\xi^T \tilde{c}^- - \bar{\alpha}\rho\|c^-\|_p \\ & + \left(\tilde{F} - F - \bar{\alpha} g^T p\right) + \rho(\|\tilde{c}^-\|_p - (1-\bar{\alpha})\|c^-\|_p).\end{aligned}$$

For the last term, from (5.7a) and (5.7b), it follows that

$$\|\tilde{c}^-\|_p - (1-\bar{\alpha})\|c^-\|_p \leq \|\tilde{c} - c - \bar{\alpha} A p\|_p,$$

and

$$\begin{aligned}\phi(\bar{\alpha}) - \phi(0) \leq {} & \bar{\alpha} g^T p + \lambda^T(\tilde{c}^- - c^-) + \bar{\alpha}\,\xi^T \tilde{c}^- - \bar{\alpha}\rho\|c^-\|_p \\ & + \left(\tilde{F} - F - \bar{\alpha} g^T p\right) + \rho\|\tilde{c} - c - \bar{\alpha} A p\|_p.\end{aligned}$$

If we use again (5.7a) and (5.7b) on the terms associated with the multiplier estimates (given that by assumption $\lambda + \bar{\alpha}\xi \geq 0$), and the Taylor series expansions for F and c, we obtain

$$\begin{aligned}\phi(\bar{\alpha}) - \phi(0) \leq {} & \bar{\alpha} g^T p + \textstyle\sum_j (\lambda_j + \bar{\alpha}\xi_j)|\tilde{c}_j - c_j - \bar{\alpha}\, a_j^T p| + (1-\bar{\alpha})\lambda^T c^- \\ & - \lambda^T c^- + \bar{\alpha}(1-\bar{\alpha})\xi^T c^- - \bar{\alpha}\rho\|c^-\|_p + O(\|\bar{\alpha}p\|^2).\end{aligned}$$

After simplifying this expression we have

$$\phi(\bar{\alpha}) - \phi(0) \leq \bar{\alpha}(g^T p + (\xi - \lambda)^T c^- - \rho\|c^-\|_p) + \sqrt{m}\bar{\alpha}^2\|\xi\|\|c^-\|_p + O(\|\bar{\alpha}p\|^2).$$

Replacing this bound in (5.6) implies

$$0 < (1-\sigma)\bar{\alpha}\Delta + \sqrt{m}\bar{\alpha}^2\|\xi\|\|c^-\|_p + O(\|\bar{\alpha}p\|^2).$$

Since from (5.8) and condition HC2, $\Delta \leq -\beta_{svH}\|p\|^2$, and we have assumed that $\|p\| > \epsilon$, it follows by taking limits along the subsequence that

$$0 \leq -(1-\sigma)\beta_{lvH}\epsilon^2.$$

However, this is not possible, implying $\|p_k\| \to 0$ for the whole sequence. □

Consider now the second merit function (5.2). The subgradient along the search direction at (x_k, λ_k) is given by

$$\phi_k'(0) = g_k^T p_k + \xi_k^T c^-(x_k) - \lambda_k^T \Omega_k A_k p_k - \rho_k c^-(x_k)^T A_k p_k \leq \Delta_k,$$

where

$$\Delta_k \equiv g_k^T p_k + (\xi_k - \lambda_k)^T c^-(x_k) - \rho_k\|c^-(x_k)\|^2.$$

Note that $\lambda_k \geq 0$ implies

$$(\Omega_k\lambda_k + \rho_k c_k^-)^T(A_k p_k + c_k) \geq 0.$$

In this case it is not immediately evident that ρ_k remains bounded. The convergence proof we give is similar to the one introduced in §3. The definition of ρ given in that section will be preserved, except $c - s$ is replaced by c^-.

THEOREM 5.2. *The algorithm modified to use the merit function* (5.2) *converges globally.*

Proof. Again, from Lemma 3.9 it is enough to show that $\|p_k\| \to 0$.

First assume that ρ is bounded. The argument used is similar to the one in Theorem 5.1. From (5.4), (5.8), condition MC2 and the boundedness of ϕ, (5.10) must hold also for this case.

If $\|p_k\| \to 0$, convergence follows from Lemma 3.9. Otherwise, if for a subsequence $\|p_k\| > \epsilon$, from (5.10) we must have $\alpha_k \to 0$, and from condition (5.6) and the boundedness of the step to satisfy (2.16), $\hat{\alpha}_k \to 0$.

From (5.2), (5.7a) and (5.7b), we also have (we again drop the index k in the following relationships, and use a tilde to indicate values at $x + \bar{\alpha}p$)

$$\begin{aligned}\phi(\bar{\alpha}) - \phi(0) \leq\ & \bar{\alpha}g^T p + \lambda^T(\tilde{c}^- - c^-) + \bar{\alpha}\,\xi^T\tilde{c}^- - \rho(\bar{\alpha} - \tfrac{1}{2}\bar{\alpha}^2)\|c^-\|^2 \\ & + \rho\|\tilde{c} - c - \bar{\alpha}Ap\|\Big(\tfrac{1}{2}\|\tilde{c} - c - \bar{\alpha}Ap\| + \|(c + \bar{\alpha}Ap)^-\|\Big) \\ & + \big(\tilde{F} - F - \bar{\alpha}g^T p\big),\end{aligned}$$

and again using (5.7a) and (5.7b) on the terms associated with the multiplier estimates, we obtain

$$\begin{aligned}\phi(\bar{\alpha}) - \phi(0) \leq\ & \bar{\alpha}\Big(g^T p + (\xi - \lambda)^T c^- - \rho\|c^-\|^2\Big) \\ & + \bar{\alpha}^2\|c^-\|\Big(\|\xi\| + \tfrac{1}{2}\rho\|c^-\|\Big) + O(\|\bar{\alpha}p\|^2).\end{aligned}$$

Replacing this bound in (5.6) implies

$$0 < (1-\sigma)\bar{\alpha}\Delta + \bar{\alpha}^2\|c^-\|\left(\|\xi\| + \tfrac{1}{2}\rho\|c^-\|\right) + O(\|\bar{\alpha}p\|^2).$$

Since from (5.8) and condition HC2, $\Delta \leq -\beta_{svH}\|p\|^2$, and we have assumed that $\|p\| > \epsilon$ and ρ is bounded, it follows by taking limits along the subsequence that

$$0 \leq -(1-\sigma)\beta_{lvH}\epsilon^2.$$

However, this is not possible, which implies $\|p_k\| \to 0$ for the whole sequence.

Assume now that ρ_k grows without bound. In this case we have that for all iterations where the value of the penalty parameter is increased

$$\rho_{k_l}\|c^-_{k_l}\| \leq K_1 \quad \text{and} \quad \rho_{k_l}\|p_{k_l}\|^2 \leq K_2.$$

The proof of this result is basically that of Lemma 3.10. From these bounds it is possible to show that we must also have

$$\rho_k\|p_k\|^2 \leq K$$

for all k (the proof is similar to the one for Lemma 3.11), implying $p_k \to 0$ and the convergence of the algorithm. □

6. Numerical results. In this section we present numerical results obtained from an implementation of our algorithm. As a first step we have modified the code NPSOL. We have called the modified routine INPSOL. Apart from the definition of the search direction all other aspects of INPSOL are identical to those of NPSOL. A detailed description of NPSOL is given in Gill et al. [15]. It should be noted that NPSOL does not incorporate linear constraints into the merit function. An initial point is obtained that is feasible with respect to the linear constraints and thereafter feasibility is retained (by incorporating the linear constraints in the QP subproblem). On many practical problems the feasible region with respect to the linear constraints is compact. On such problems this approach ensures Assumption A2 is satisfied, and Assumption A1 is implied by Assumption A3.

The purpose of the testing reported is to demonstrate that the efficiency and robustness of the modified algorithm are comparable to those of NPSOL. Naturally, we can only test the hypothesis on the domain of problems NPSOL is designed to solve, namely, problems having a small number of variables and constraints, although on these problems the opportunities for improvement are limited, as we discuss later. What this implementation really tests is whether the introduction of flexibility in the determination of the search direction has a significant cost. The parameter β_c was set to infinity to avoid differences with NPSOL arising due entirely to the linesearch.

6.1. The search direction. The algorithm described in §2 allows for considerable flexibility of design. We describe here the specific choices made in our implementation. The search direction p_k is computed according to the following steps. (The subscript k is dropped from now on.)

1. An initial feasible point for each QP subproblem, p_0, is obtained following the same procedure as NPSOL. No special effort was made to satisfy conditions (2.18) since on the problems tested no failure was detected that could be attributed to the size of $\|p_0\|$.

2. The active-set method used in NPSOL was terminated at $\tilde{p}$, the *first* stationary point. The multipliers π at $\tilde{p}$ are then computed. Define $\hat{\pi}$ as $\hat{\pi}_j \equiv \pi_j \|a_j\|$.

3. Let ϵ_M denote machine precision. If

$$\forall j, \quad \hat{\pi}_j \geq -\sqrt{\epsilon_M}, \tag{6.1}$$

then $\tilde{p}$ is taken as the search direction.

4. If (6.1) does not hold a step that moves off a subset of the active constraints is computed. To identify the set of active constraints to be deleted, define $\pi_{\min} \equiv \min_j \hat{\pi}_j$, and introduce a vector e_I as

$$(e_I)_j \equiv \begin{cases} \|a_j\| & \text{if } \hat{\pi}_j \leq 10^{-3}\pi_{\min}, \\ 0 & \text{otherwise.} \end{cases} \tag{6.2}$$

5. There is also a limit of 50 on the maximum number of constraints to be deleted. If (6.2) is satisfied by more than 50 active constraints, only the ones having the smallest multipliers are deleted. For most problems this limit has no effect, since the total number of constraints is less than 50. This limit was introduced to limit the cost of refactorization for the Jacobian matrix.

6. The direction d that moves off the selected constraints is obtained as the least-length solution of the system $Au = e_I$; that is, we define

$$d = Y(AY)^{-1}e_I,$$

where Y denotes a basis for the range-space of A^T.

7. We obtain the search direction p from (2.21), as

$$p \equiv \begin{cases} \tilde{p} + \tilde{\gamma} d & \text{if } \|\tilde{p}\| < \beta_{slp}\|\tilde{p} + \tilde{\gamma} d\|, \\ \tilde{p} & \text{otherwise,} \end{cases}$$

where $\tilde{\gamma}$ was defined as in (2.26) with $\gamma_M = 10^{10}$ and $\beta_{slp} = 100$ (with this value the step $\tilde{p} + \tilde{\gamma} d$ is accepted in nearly all cases).

8. Finally, the multiplier estimate used to define the linesearch is taken to be π if $p = \tilde{p}$. Otherwise, it is taken to be the least-squares estimate μ_L obtained from

$$AA^T\mu_L = Ag.$$

6.2. Test problems. The two algorithms, NPSOL and INPSOL, have been compared by solving a collection of 114 problems from the literature. The problems have been obtained from the following sources.

(i) Problem 1 is the example problem distributed with NPSOL; its description can be found in [15]. Problems 3 and 4 are slight reformulations of the same problem, where the bounds $-1 \leq x_3 \leq 1$ have been replaced by the constraint $x_3^2 \leq 1$. Problem 3 uses the starting point

$$(\tfrac{1}{3}, \tfrac{2}{3}, \tfrac{11}{10}, \tfrac{2}{3}, \tfrac{1}{3}, \tfrac{1}{3}, \tfrac{2}{3}, -\tfrac{1}{3}, -\tfrac{1}{3}).$$

(ii) Descriptions for problems 6 and 12–15 can be found in [25]. The version of problem 6 considered is the one corresponding to a value $T = 10$. Problems 12 and 13 start from point (d) for Wright No. 4 as indicated in the reference, while problems 14 and 15 start from points (a) and (b) for Wright No. 9, respectively.

(iii) A description of the SQUARE ROOT problems (17–20) and of EXP6 (9) can be found in Fraley [14].

(iv) Problems 21–30 were obtained from Boggs and Tolle [2].

(v) All problems having names starting with HS are from Hock and Schittkowski [22].

(vi) Problems 85–95 can be found in Dembo [8].

All the above problems have been used in the past to test NPSOL. It should be noted that the problems in this group are small; the average number of variables is 10, and the average number of constraints is 6. Nevertheless, many of these problems are considered hard to solve. Moreover, for some of these problems the assumptions made to establish the convergence results fail to hold; for example, in some cases the Jacobian of the active NP constraints at x^* is singular, or no feasible points exist for some QP subproblems. In problem 42 no feasible point exists for NP.

The algorithms have also been tested on another group of problems.

(vii) The structural optimization problems 99–114 are from Ringertz [30]. The letters I and E in the problem name indicate if the formulation used included explicitly the displacement variables (E) or eliminated them in advance. Also, the following number (10, 25, 36, or 63) denotes the number of bars in the truss considered. Finally, whenever a number is included at the end of the name (006, 040, or 060), the initial point taken has been modified to be $x_j = 6$, 40, or 60, respectively.

These problems have been introduced due to the atypical behavior of quasi-Newton SQP algorithms on them. For this group, the ratio of QP to nonlinear iterations is large when compared to the size of the problem; on the first test set (problems 1–98) the average ratio for NPSOL is 2 QP iterations per nonlinear iteration, while on problems 99–114 the average ratio is 30.

The normal behavior of NPSOL on the first set of test problems is to require a relatively large number of QP iterations in the first few nonlinear iterations. Typically, the number of QP iterations declines exponentially until near a KKT point, when only one iteration is required. The STRUC problems depart from this "standard" behavior, in the sense that the number of QP iterations declines much more gradually. (Although only one QP iteration is required in the end, most nonlinear iterations require more.) This offers the possibility of observing the reductions that can be achieved by using the early-termination criterion, with limited distortion from the asymptotic behavior of NPSOL.

Finally, the problems in this second group are larger than the ones presented above; the average number of variables is now 55, and the average number of constraints is 100. For all the reasons mentioned, this set of problems provides a better environment in which to test the ability of the proposed early-termination criterion to reduce the number of QP iterations.

6.3. Computing environment. Version 4.02 of NPSOL was used in these comparisons. For this test set, all parameters used in the code have been fixed at their default values (see [15]). No attempt was made to improve the results by selecting a different set of parameters. It would be difficult to compare the relative effort to adjust input parameters for the two algorithms. The runs were performed as batch jobs on a DEC VAXstation II with 5 Mb main memory. The operating system was VAX/VMS version 4.5, and the compiler used was VAX FORTRAN version 4.6 with default options.

6.4. Results. The results obtained from running both algorithms on the test set are presented in Tables 4 and 2.

The parameters chosen to characterize the relative performance of both algorithms have been: the number of outer (nonlinear) iterations for each problem; the

TABLE 1

Average behavior: NPSOL *vs.* INPSOL.

	Problems		
	All	1–98	99–114
Nonlinear iterations	.988	.979	1.044
Function evaluations	.994	.999	.963
QP iterations	1.190	1.112	1.884
CPU time	1.043	1.022	1.200

number of calls to the routine computing the values of the objective function, the constraint functions and their derivatives (function evaluations); the total number of inner (QP) iterations for the problem (this includes the number of iterations necessary to compute a feasible point); and the running (CPU) time needed to solve the problem. The results corresponding to both algorithms are given as a single entry in the tables, with the figures separated by a slash (/) symbol, in the form NPSOL result/INPSOL result.

Given that most of the problems are not convex, the algorithms may converge to different KKT points. Three such events occurred. Another possible outcome is failure—that is, the algorithm terminates without finding a solution, because the iteration limit has been exceeded, because no significant progress can be made at the current point with respect to the merit function, or because the objective or constraint functions need to be evaluated at a point for which they are not defined in the code. Such failures are indicated by a long dash (—).

For the set of 114 problems, NPSOL was able to find a KKT point in 107 cases, while INPSOL was able to solve 105 problems. We should emphasize that only the default value of the input parameters were used. Undoubtedly adjustment of the input parameters on the problems that failed would have led to more successes. The figures illustrate the reliability of INPSOL.

Table 1 presents a summary of the results for the four quantities monitored in Table 2. The average values have been computed as the geometric means for the ratios of the values for NPSOL and for INPSOL; that is, averages larger than one indicate that the corresponding value for NPSOL is larger than the value for INPSOL. Also, the averages exclude those problems where one of the algorithms failed. Separate entries have been provided for problems 1–98 (the smaller problems), and for problems 99–114 (the structural optimization problems).

We now comment briefly on the implications of these results.

(i) The early-termination rule seems to behave very well regarding the numbers of nonlinear iterations and function evaluations; even if we are now using a search direction of "worse quality" than in NPSOL, the numbers are very close for both algorithms.

(ii) The number of QP iterations is reduced by 20% for the complete set. When judging this figure we must take into account that the problems are small, implying that the number of QP iterations required per nonlinear iteration is also small. (In fact, the average value for the test set is 5.6 QP iterations per nonlinear iteration.) The opportunity for improvement is correspondingly limited. Moreover, both codes use the active set at the solution of the previous QP subproblem as a prediction for the correct active set in the current subproblem, resulting in a small number of QP iterations close to a KKT point. As a result, significant savings achieved by incomplete solution of QP subproblems in the early iterations are masked by a large number of subproblems requiring only a few QP iterations. As an example, for problem 98 the largest number of QP iterations needed in any nonlinear iteration is reduced from 57 for NPSOL to 15 for INPSOL. This effect is much less clear when we look at total numbers of QP iterations (244 for NPSOL vs. 170 for INPSOL). Recall that it is necessary in any implementation to limit the number of iterations taken to solve the subproblem. This large reduction in the maximum number of iterations is encouraging. Moreover, it indicates that INPSOL and NPSOL took quite different paths to obtain a solution on many of the problems. In the light of this fact the similarity of performance is quite remarkable. Finally, the early-termination rule still requires a feasible point, and the feasibility phase is the same as in NPSOL. When this phase accounts for most of the total number of iterations, as with the STRUC problems, the possibility of improvement is further diminished.

Nonetheless, it should be noted that for problems 99–114 the improvement obtained is significantly greater than 20%, as the mean ratio is now 1.88; in fact, when we look only at the larger problems, the relative performance of INPSOL improves markedly. This offers the promise that for even larger problems the results obtained may be substantially better than the values shown above.

(iii) The CPU time required by INPSOL is lower than the time for NPSOL, but by a factor that is much smaller than for the number of QP iterations. This is due not only to the fact that function evaluations can be expensive when compared to the effort to solve each QP subproblem, but also to some details in the implementation that have been chosen to affect the number of QP iterations, even at the expense of running time. For example, the multiplier estimate used for the linesearch (the least-squares multiplier) is expensive to compute when many constraints are deleted in the last step, as the factorization for the Jacobian of the active constraints must be updated. There are still options to be explored that might reduce the CPU time for the modified algorithm.

7. Acknowledgments. We are grateful to the referees for their effort at refereeing a long and difficult paper. Their care and attention to detail resulted in a substantial improvement over the first version of this paper. The prodding of one referee in particular led to our weakening our assumptions and including considerable new material in the paper.

TABLE 2
Numerical results.

No.	Problem name	Nonlinear iterations	Function evaluations	QP iterations	CPU time (s)
1	NPSOL SAMPLE PROBLEM	12/13	16/18	45/34	3.69/3.61
2	SINGULAR	15/15	16/16	4/4	1.03/1.05
3	HEXAGON	15/16	21/23	32/29	4.41/4.41
4	HEXAGON (ALT. START)	11/11	16/14	35/26	3.56/3.26
5	LC7	7/9	9/11	13/16	.76/.95
6	ALAN MANNE'S PROBLEM	17/17	18/18	40/37	21.13/21.92
7	ROSEN-SUZUKI	8/8	11/11	9/9	.81/.81
8	QP PROBLEM	8/10	9/11	23/15	1.10/1.04
9	EXP6	33/53	35/57	38/57	1.96/3.08
10	STEINKE2	—*/5	—/6	—/14	—/.87
11	NORWAY	4/6†	5/7	34/13	1.23/.65
12	MHW4	10/10	18/15	14/12	1.31/1.25
13	MHW9	30/19†	56/28	42/24	3.71/2.31
14	MHW9 INEQUALITY 1	28/23	38/28	59/40	3.41/2.73
15	MHW9 INEQUALITY 2	41/14†	58/27	80/24	4.83/1.77
16	WOPLANT	25/29	29/33	44/35	6.85/7.17
17	SQUARE ROOT 1	—*/—*	—/—	—/—	—/—
18	SQUARE ROOT 2	23/23	36/36	0/0	5.01/5.32
19	SQUARE ROOT 3	6/6	9/9	7/7	.95/.94
20	SQUARE ROOT 4	—*/—*	—/—	—/—	—/—
21	BT1	11/11	19/19	11/11	.81/.83
22	BT2	9/9	14/14	9/9	.71/.70
23	BT3	2/2	5/5	2/2	.19/.19
24	BT4	12/12	18/18	13/13	.92/.92
25	BT5–HS63	6/6	9/9	8/8	.58/.58
26	BT6–HS77	15/15	21/21	16/16	1.52/1.54
27	BT7	31/31	56/56	32/32	3.36/3.43
28	BT8	17/17	19/19	17/17	1.25/1.44
29	BT9–HS39	13/13	16/16	14/14	.95/1.19
30	BT10	8/8	11/11	0/0	.48/.52
31	BT11–HS79	9/9	12/12	10/10	1.05/1.06
32	BT12	27/27	57/57	28/28	3.04/3.04
33	BT13	32/32	44/44	34/34	2.61/2.62
34	POWELL TRIANGLES	23/15	37/16	36/23	3.27/2.28
35	POWELL BADLY SCALED	12/12	15/15	13/13	.85/.85
36	POWELL WRIGGLE	34/32	69/55	60/40	2.77/2.39
37	POWELL-MARATOS	6/6	7/7	6/6	.44/.44
38	HS72	7/7	8/8	8/8	.69/.67
39	HS73 (CATTLE FEED)	4/4	5/5	4/4	.38/.36
40	HS107	11/11	18/18	27/18	2.77/2.56
41	MUKAI-POLAK	10/10	16/16	13/13	1.08/1.11
42	INFEASIBLE SUBPROBLEM	—*/—*	—/—	—/—	—/—
43	HS26	47/47	64/64	48/48	3.39/3.41
44	HS32	2/4	3/5	3/5	.25/.38
45	HS46	55/55	58/58	56/56	5.26/4.98
46	HS51	2/2	5/5	2/2	.18/.14
47	HS52	2/2	5/5	2/2	.19/.16
48	HS53	2/2	5/5	2/2	.19/.16
49	PENALTY1 A	16/16	18/19	77/41	20.01/16.49
50	PENALTY1 B	6/7	14/19	67/32	14.77/11.77
51	PENALTY1 C	29/15	85/40	152/65	24.35/11.65
52	HS13	22/19	23/20	13/10	1.29/1.22
53	HS64	29/43	39/62	47/60	2.34/3.33
54	HS65	8/9	10/11	16/16	.70/.78
55	HS70	36/—*	39/—	39/- -	3.33/—
56	HS71	5/7	6/9	9/9	.53/.67
57	HS74	10/26	15/48	14/28	1.17/2.68
58	HS75	6/8	10/11	7/9	.72/.90
59	HS78	10/10	14/14	11/11	1.15/1.15
60	HS80	8/8	10/10	8/8	.92/.92
61	HS81	14/14	20/20	15/15	1.57/1.60
62	HS84	—*/4	—/5	—/9	—/.51
63	HS85	17/14	18/15	33/20	4.00/3.12
64	HS86 (COLVILLE 1)	6/7	8/8	11/11	.62/.64
65	HS87 (COLVILLE 6)	11/8	18/9	18/14	1.63/1.23
66	HS93	12/12	15/15	14/14	1.36/1.38
67	HS95	1/1	2/2	1/1	.15/.15
68	HS96	1/1	2/2	1/1	.17/.15
69	HS97	3/3	6/6	3/3	.40/.41
70	HS98	3/3	6/6	8/8	.43/.44
71	HS99	23/—*	44/—	74/—	3.99/—
72	HS100	14/14	29/29	18/18	2.07/2.02
73	HS104	18/18	20/20	23/23	3.36/3.37
74	HS105	43/—*	61/—	97/—	27.14/—
75	HS108 (HEXAGON)	24/32	45/49	57/87	6.78/9.36
76	HS109	11/10	13/11	25/29	3.23/3.26
77	HS110	6/6	9/9	24/15	.78/.69
78	HS111	41/49	64/75	44/52	8.08/9.05

* Failed to solve the problem.
† Converged to a different minimizer.

TABLE 2 (cont.)
Numerical results.

No.	Problem name	Nonlinear iterations	Function evaluations	QP iterations	CPU time (s)
79	HS112 (CHEMICAL EQ.)	19/—*	39/—	54/—	2.78/—
80	HS113	14/16	19/23	38/36	3.12/3.41
81	HS114	18/16	19/24	36/33	3.81/3.60
82	HS117 (COLVILLE 2)	17/18	21/27	96/39	6.75/5.34
83	HS118 (LC PROBLEM)	4/4	6/6	20/20	1.35/1.40
84	HS119 (COLVILLE 7)	12/17	16/19	41/47	4.25/5.60
85	DEMBO 1B	281/—*	437/—	296/—	75.46/—
86	DEMBO 2–HS83	4/4	6/6	4/4	.54/.54
87	DEMBO 3	9/8	11/9	37/20	2.01/1.78
88	DEMBO 4A	19/19	23/23	24/24	3.53/3.31
89	DEMBO 4C	13/13	15/15	20/23	3.10/3.20
90	DEMBO 5–HS106	17/18	21/24	30/31	2.90/3.04
91	DEMBO 6–HS116	36/43	96/69	144/248	21.84/29.65
92	DEMBO 7	19/12	24/15	126/68	15.54/9.82
93	DEMBO 8A	33/42	85/118	105/99	7.52/9.17
94	DEMBO 8B	29/29	69/71	88/73	6.51/6.45
95	DEMBO 8C	25/27	60/68	89/65	6.19/6.06
96	OPF	18/17	19/18	53/51	468.12/456.10
97	GBD EQUILIBRIUM MOD.	5/6	6/7	37/26	6.22/6.10
98	WEAPON ASSIGNMENT	96/73	98/76	244/170	120.78/114.93
99	STRUCI10KON	18/17	34/30	65/42	13.67/11.73
100	STRUCE10KON	26/29	49/67	87/84	17.68/20.75
101	STRUCI10VAN	23/19	41/34	54/51	16.30/13.85
102	STRUCE10VAN	—*/24	—/48	—/91	—/19.44
103	STRUCI25006	42/37	68/62	147/85	92.44/80.99
104	STRUCE25006	20/28	32/36	178/95	357.83/260.79
105	STRUCI25DAT	11/12	19/21	24/22	24.75/27.11
106	STRUCE25DAT	52/21	106/37	687/65	647.13/191.44
107	STRUCI36DAT	23/20	38/34	59/46	120.79/108.02
108	STRUCE36DAT	29/30	53/62	87/90	971.16/1021.9
109	STRUCI63040	117/112	211/202	6116/3091	8182.1/7159.0
110	STRUCE63040	375/—*	794/—	3545/—	77286.6/—
111	STRUCI63060	—*/98	—/244	—/3899	—/8281.0
112	STRUCE63060	63/115	150/316	6675/3407	25090.2/33228.4
113	STRUCI63DAT	246/136	354/412	9043/2060	12591.6/11424.5
114	STRUCE63DAT	52/72	86/145	8049/2858	41793.8/22740.7

* Failed to solve the problem.
† Converged to a different minimizer.

REFERENCES

[1] M. C. BIGGS (1972), *Constrained minimization using recursive equality quadratic programming*, in Numerical Methods for Nonlinear Optimization, F.A. Lootsma, ed., Academic Press, London, New York.

[2] P. T. BOGGS AND J. W. TOLLE (1984), *A family of descent functions for constrained optimization*, SIAM J. Numer. Anal., 21, pp. 1146–1161.

[3] P. T. BOGGS, J. W. TOLLE, AND P. WANG (1982), *On the local convergence of quasi-Newton methods for constrained optimization*, SIAM J. Control Optim., 20, pp. 161–171.

[4] J. V. BURKE AND S.-P. HAN (1989), *A robust sequential quadratic programming algorithm*, Math. Programming, 43, pp. 277–303.

[5] R. H. BYRD AND J. NOCEDAL (1988), *An analysis of reduced Hessian methods for constrained optimization*, Report CU-CS-398-88, Department of Computer Science, University of Colorado, Boulder.

[6] P. H. CALAMAI AND J. J. MORÉ (1987), *Projected gradient methods for linearly constrained problems*, Math. Programming, 39, pp. 93–116.

[7] M. R. CELIS, J. E. DENNIS, JR., AND R. A. TAPIA (1985), *A trust region strategy for nonlinear equality constrained optimization*, in Numerical Optimization, P. T. Boggs, R. H. Byrd, and R. B. Schnabel, eds., Society for Industrial and Applied Mathematics, Philadelphia.

[8] R.S. DEMBO (1976), A set of geometric programming test problems and their solutions, *Math. Programming*, 10, pp. 192–213.

[9] R.S. DEMBO AND U. TULOWITZKI (1985), *Sequential truncated quadratic programming methods*, in Numerical Optimization, P.T. Boggs, R.H. Byrd, and R.B. Schnabel, eds., Society for Industrial and Applied Mathematics, Philadelphia.

[10] S.K. ELDERSVELD (1991), *Large-scale sequential quadratic programming algorithms*, Ph. D. thesis, Stanford University, Stanford, CA.

[11] R. FLETCHER (1970), *A class of methods for nonlinear programming with termination and convergence properties*, in Integer and Nonlinear Programming, J. Abadie, ed., North Holland, Amsterdam.

[12] ——— (1985), *An ℓ_1 penalty method for nonlinear constraints*, in Numerical Optimization, P.T. Boggs, R.H. Byrd, and R.B. Schnabel, eds., Society for Industrial and Applied Mathematics, Philadelphia.

[13] ——— (1987), *Practical Methods of Optimization*, John Wiley and Sons, Chichester, New York, Brisbane, Toronto, Singapore.

[14] C. FRALEY (1988), *Software performance on nonlinear least-squares problems*, SOL Report 88-17, Department of Operations Research, Stanford University, Stanford, CA.

[15] P.E. GILL, W. MURRAY, M.A. SAUNDERS, AND M.H. WRIGHT (1986), *User's guide for* NPSOL *(Version 4.0): a* FORTRAN *package for nonlinear programming*, Report SOL 86-2, Department of Operations Research, Stanford University, Stanford, CA.

[16] ———(1986), *Some theoretical properties of an augmented Lagrangian merit function*, Report SOL 86-6R, Department of Operations Research, Stanford University, Stanford, CA.

[17] ———(1988), *Inertia-controlling methods for quadratic programming*, SIAM Rev., 33, pp. 1–33.

[18] P.E. GILL, W. MURRAY, AND M.H. WRIGHT (1981), *Practical Optimization*, Academic Press, London, New York.

[19] J. GOODMAN (1985), *Newton's method for constrained optimization*, Math. Programming, 33, pp. 162–171.

[20] C.B. GURWITZ AND M.L. OVERTON (1989), *Sequential quadratic programming methods based on approximating a projected Hessian matrix*, SIAM J. Sci. Statist. Comput., 10, pp. 631–653.

[21] S.-P. HAN (1976), *Superlinearly convergent variable metric algorithms for general nonlinear programming problems*, Math. Programming, 11, pp. 263–282.

[22] W. HOCK AND K. SCHITTKOWSKI, (1981), *Test examples for nonlinear programming*, Lecture Notes in Economics and Mathematical Systems, Vol. 187, Springer-Verlag, Berlin, Heidelberg, New York.

[23] J.J. MORÉ AND D.C. SORENSEN (1984), *Newton's method*, in Studies in Numerical Analysis, G.H. Golub, ed., Mathematical Association of America, pp. 29–82.

[24] W. MURRAY (1969), *An algorithm for constrained minimization*, in Optimization, R. Fletcher, ed., Academic Press, London, New York.

[25] B.A. MURTAGH AND M.A. SAUNDERS (1982), *A projected Lagrangian algorithm and its implementation for sparse nonlinear constraints*, Math. Programming Stud., 16, pp. 84–117.

[26] M.J.D. POWELL (1978), *A fast algorithm for nonlinearly constrained calculations*, in Nonlinear Programming 3, O.L. Mangasarian, R.R. Meyer, and S.M. Robinson, eds., Academic Press, New York.

[27] ——— (1987), *Methods for nonlinear constraints in optimization calculations*, in Proceedings of the 1986 IMA/SIAM Conference, Clarendon Press, Oxford.

[28] M.J.D. POWELL AND Y. YUAN (1986), *A recursive quadratic programming algorithm that uses differentiable exact penalty functions*, Math. Programming, 35, pp. 265–278.

[29] F.J. PRIETO (1989), *Sequential quadratic programming algorithms for optimization*, Report SOL 89-7, Department of Operations Research, Stanford University, Stanford, CA.

[30] U.T. RINGERTZ (1988), *A mathematical programming approach to structural optimization*, Report No. 88-24, Dept. of Aeronautical Structures and Materials, The Royal Institute of Technology, Stockholm.

[31] S.M. ROBINSON (1974), *Perturbed Kuhn–Tucker points and rates of convergence for a class of nonlinear programming algorithms*, Math. Programming, 7, pp. 1–16.

[32] K. SCHITTKOWSKI (1981), *The nonlinear programming method of Wilson, Han and Powell with an augmented Lagrangian line search function*, Numer. Math., 38, pp. 83–114.

[33] R.B. WILSON (1963), *A Simplicial Algorithm for Concave Programming*, Ph.D. thesis, Harvard University, Cambridge, MA.

[34] M.H. WRIGHT (1976), *Numerical methods for nonlinearly constrained optimization*, Ph.D. thesis, Stanford University, Stanford, CA.

SIAM J. OPTIMIZATION
Vol. 5, No. 3, pp. 641–658, August 1995

008

LOCAL CONVERGENCE OF SQP METHODS IN SEMI-INFINITE PROGRAMMING*

G. GRAMLICH†, R. HETTICH‡, AND E.W. SACHS‡

Abstract. In this paper we begin with pointing out how a semi-infinite programming problem can be reduced locally to a problem of finite dimensional programming. Such a reduction has the advantage that efficient numerical methods like sequential quadratic programming (SQP) methods can be applied. However, the reduced problem involves constraint functions that are defined only implicitly. Values of these functions and their derivatives must be computed iteratively with controllable errors. We interpret them as perturbations of the correct constraints and apply an SQP method with a Broyden-Fletcher–Goldfarb–Shanno (BFGS) update. Extending the convergence analysis by Fontecilla, Steihaug, and Tapia for these methods to include perturbations of the constraints and their derivatives, we are able to show q-superlinear convergence and at the same time to indicate at which rate the error in the calculation of the constraints must be reduced as the iteration progresses.

Key words. semi-infinite programming, SQP methods, superlinear convergence

AMS subject classifications. 65K05, 49D39, 49D15

1. Introduction.

In this paper, we consider semi-infinite programming problems of the following type:

$$\text{(SIP)} \qquad \text{Maximize } F(z) \text{ subject to } z \in Z \subset \mathbb{R}^n,$$

where the feasible set Z is assumed to be nonempty and defined by

$$Z := \{z \mid g(z,t) \leq 0 \text{ for all } t \in B\} \subset \mathbb{R}^n \tag{1.1}$$

with $B \subset \mathbb{R}^m$, a compact set given by

$$B := \{t \mid h^j(t) \leq 0, j \in J\} \subset \mathbb{R}^m$$

for a finite set of indices J. All the functions

$$\begin{aligned} F &: \mathbb{R}^n \to \mathbb{R}, \\ g &: \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}, \\ h^j &: \mathbb{R}^m \to \mathbb{R} \end{aligned}$$

are assumed to be everywhere twice continuously differentiable with Lipschitz-continuous derivatives. More generally, instead of only one constraint $g(z,t) \leq 0$ we could have finitely many in (1.1) and, in addition, a finite number of equality constraints $e(z) = 0$. Since the following algorithm and convergence theory can be extended to this case in an obvious way, the formulation of (SIP) has been chosen for simplicity. The same is true for so-called generalized semi-infinite programming problems, where B also depends on z (i.e., $h^j = h^j(z,t)$ are considered). In this case more complicated expressions for the derivatives of the Lagrangians, etc. occur (cf. [16], [11] for

* Received by the editors December 9, 1992; accepted for publication (in revised form) May 26, 1994.

† DLR, Institut für Robotik und Systemdynamik Oberpfaffenhofen, 82230 Wessling, Germany.

‡ Universität Trier, Fachbereich IV – Mathematik, 54286 Trier, Germany (hettich@uni-trier.de, sachs@uni-trier.de).

instance). However, in principle there is no serious difficulty in dealing with these problems in the same way as with (SIP).

It is well known (cf. [10]) that under some regularity assumptions, the problem can be reduced locally to a finite optimization problem with constraints $v^\ell(z) \leq 0$, where the functions $v^\ell(z) := g(z, t^\ell(z))$ are C^2 and are defined implicitly via the local maxima $t^\ell(z)$ of $g(z,t)$, $t \in B$. This reduction is presented in §2 and forms the basis of so-called reduction methods in §3. A special one is discussed in §4: The reduced problems are solved by the sequential quadratic approximation (SQP) technique using augmented Lagrangians and quasi-Newton BFGS approximations of the Hessian. Theorem 4.2 gives the local q-superlinear convergence of this method as an immediate consequence of theorems in [5].

In [14] the following question was posed using the usual Lagrangian instead of the augmented Lagrangian: If the exact maxima $t^\ell(z)$ were replaced by approximations $\tilde{t}^\ell(z)$, what are sufficient conditions on $\|t^\ell(z) - \tilde{t}^\ell(z)\|$ for superlinear convergence? Under strong assumptions on the problem it was shown in [14] that $\|\tilde{t}^\ell(z) - t^\ell(z)\| = O(\|z - z_*\|^2)$ is sufficient (and $O(\|z - z_*\|)$ is not).

Under common definiteness assumptions on z_*, it was shown in [6] that the superlinear convergence is retained under the assumption of

$$\|\tilde{t}^\ell(z+s) - t^\ell(z+s)\| + \|\tilde{t}^\ell(z) - t^\ell(z)\| \leq K \max\{\|z - z_*\|, \|z + s - z_*\|\}\|z + s - z_*\| .$$

In this paper, using results from [5], we give (under the same assumptions on z_* as in [6]) a simpler proof of superlinear convergence under the following, more transparent assumptions on the approximation of $t^\ell(z)$ by $\tilde{t}^\ell(z)$:

$$\|\tilde{t}^\ell(z) - t^\ell(z)\| = o(\|z - z_*\|)$$

and (6.10) (equivalent to (6.9) in Assumption 6.4)

$$\|\tilde{t}^\ell(z+s) - \tilde{t}^\ell(z) - t^\ell_z(z_*)s\| \leq \kappa_6 \max\{\|z + s - z_*\|, \|z - z_*\|\}\|s\|$$

for z "close" to z_* and "small" s (cf. Assumption 6.4). The latter bound is intuitively clear by observing that in the second derivatives of v^ℓ (cf. (2.4)), the derivative $t_z(z_*)$ occurs. Therefore, a second order theory is likely to require good approximations such as (6.10) for the derivative. We note that $\|\tilde{t}^\ell(z) - t^\ell(z)\| = O(\|z - z_*\|^2)$ follows from our assumptions.

This paper is concerned mainly with an analysis of the convergence properties. We note that the algorithms considered in this paper have been implemented and tested on various problems arising in the literature, see [6]. Furthermore, these methods have been applied successfully to the path planning problem in robotics, see [8] and [7], a complex and highly nonlinear problem of practical interest.

2. Local representation of the feasible set. Define the parametric programming problem as

$$\text{(PAR}(z)\text{)} \qquad \text{Maximize} \quad g(z,t) \quad \text{subject to} \quad t \in B.$$

We denote by $v(z)$ the marginal value function of (PAR(z)), i.e.,

$$v(z) := \max\{g(z,t) \quad | \quad t \in B\}.$$

Obviously we have

$$z \in Z \quad \text{if and only if} \quad v(z) \leq 0.$$

Therefore, (SIP) can be stated equivalently as an optimization problem with only one constraint:

$$\text{(SIP)} \qquad \text{Maximize } F(z) \text{ subject to } v(z) \leq 0.$$

However, as $v(z)$ is generally a complicated, nondifferentiable function, there is no substantial progress in this reformulation.

The goal of the following is to represent $v(z)$ locally near almost every $\bar{z} \in \mathbb{R}^n$ by

$$v(z) = \max \{v^\ell(z) \quad | \quad \ell \in L\}$$

with smooth (C^1- or C^2-) functions $v^\ell, |L| < \infty$, defined on some neighborhood of $\bar{z}$. This can be achieved under certain regularity assumptions that are of a generic nature.

Denote for given $\bar{z} \in \mathbb{R}^n$ with

$$\bar{t}^\ell, \quad \ell \in \bar{L}, \quad \bar{L} \text{ not necessarily countable},$$

all local solutions of (PAR($\bar{z}$)). Obviously

$$\bar{z} \in Z \quad \text{if and only if} \quad g(\bar{z}, \bar{t}^\ell) \leq 0, \quad \ell \in \bar{L}.$$

Then we define $v^\ell(z) = g(z, \bar{t}^\ell)$. This will be extended below by letting $\bar{t}^\ell$ depend also on z. For every $\ell \in \bar{L}$, consider the Karush–Kuhn–Tucker system

$$\text{(KKT}_\ell(\bar{z})) \qquad \begin{aligned} g_t(z,t) - \sum_{j \in J^\ell(\bar{z})} \alpha^{\ell,j} \quad & h_t^j(t) = 0, \\ & h^j(t) = 0, \qquad j \in J^\ell(\bar{z}), \end{aligned}$$

with

$$J^\ell(\bar{z}) = \{j \in J \quad | \quad h^j(\bar{t}^\ell) = 0\},$$

where $\bar{t}^\ell$ is a local solution of (PAR($\bar{z}$)).

Suppose that the linear independence constraint qualification (LICQ) holds in $\bar{t}^\ell$, i.e.,

$$h_t^j(\bar{t}^\ell), \quad j \in J^\ell(\bar{z}), \qquad \text{are linearly independent.} \tag{2.1}$$

Then a solution of (KKT$_\ell(\bar{z})$) is given by $\bar{z}, \bar{t}^\ell, \bar{\alpha}^{\ell,j}, j \in J^\ell(\bar{z})$, where $\bar{\alpha}^{\ell,j} \geq 0$ are the unique Lagrange multipliers. Next, we give assumptions which ensure that (KKT$_\ell(\bar{z})$) defines implicitly functions $t^\ell, \alpha^{\ell,j}$ on a neighborhood $U_{\bar{z}}$ of $\bar{z}$ such that $t^\ell(z)$, $\alpha^{\ell,j}(z)$, $z \in U_{\bar{z}}$, are solutions and Lagrange multipliers of (PAR(z)).

DEFINITION 2.1. *The local solution $\bar{t}^\ell$ of* (PAR($\bar{z}$)) *is called* nondegenerate, *if the following conditions hold:*

(i) *the linear independence constraint qualification* (LICQ), *cf.* (2.1),

(ii) *the strong second order sufficiency condition* (SSOSC), *which requires that with the Lagrangian*

$$\mathcal{L}^\ell(z,t,\alpha) := g(z,t) - \sum_{j \in J^\ell(\bar{z})} \alpha^{\ell,j} h^j(t),$$

we have

$$\mathcal{L}_t^\ell(\bar{z}, \bar{t}^\ell, \bar{\alpha}^\ell) = 0, \quad \bar{\alpha}^\ell = (\bar{\alpha}^{\ell,j})_{j \in J^\ell(\bar{z})},$$

$$\eta^T \mathcal{L}_{tt}^{\ell}(\bar{z}, \bar{t}^{\ell}, \bar{\alpha}^{\ell})\eta < 0 \quad \textit{for} \quad \eta \in T_{\ell} \setminus \{0\},$$

where

$$T_{\ell} := \{\eta \quad | \quad \eta^T h_t^j(\bar{t}^{\ell}) = 0, \quad j \in J^{\ell}(\bar{z})\},$$

(iii) *the strict complementary slackness* (SCS), *i.e.,*

$$\bar{\alpha}^{\ell,j} > 0, \quad j \in J^{\ell}(\bar{z}).$$

The following theorem is standard in semi-infinite programming and is easily proved by means of the implicit function theorem (see [11], [12]).

THEOREM 2.2. *Given* $\bar{z} \in \mathbb{R}^n$. *Let* $\bar{t}^{\ell}$ *be a nondegenerate local solution of* (PAR($\bar{z}$)). *Then there exists a neighborhood* $U_{\bar{z}}^{\ell}$ *of* $\bar{z}$ *and continuously differentiable functions*

$$t^{\ell} : U_{\bar{z}}^{\ell} \to B, \quad \alpha^{\ell,j}(\bar{z}) : U_{\bar{z}}^{\ell} \to \mathbb{R}^+, \quad j \in J^{\ell}(\bar{z}),$$

with

$$t^{\ell}(\bar{z}) = \bar{t}^{\ell}, \quad \alpha^{\ell,j}(\bar{z}) = \bar{\alpha}^{\ell,j}, \quad j \in J^{\ell}(\bar{z}),$$

such that

(i) *for every* $z \in U_{\bar{z}}^{\ell}$ *the point* $t^{\ell}(z)$ *is a nondegenerate local solution of* (PAR(z)) *with* (*unique*) *Lagrange multipliers* $\alpha^{\ell,j}(z)$;

(ii) *the derivative-matrices* $t_z^{\ell}(\bar{z}) \in \mathbb{R}^{m \times n}$, $\alpha_z^{\ell}(\bar{z}) \in \mathbb{R}^{|J^{\ell}(\bar{z})| \times n}$ *are the unique solutions of the system*

$$\begin{pmatrix} \mathcal{L}_{tt}^{\ell}(\bar{z}, \bar{t}^{\ell}, \bar{\alpha}^{\ell}) & (H^{\ell}(\bar{t}^{\ell}))^T \\ H^{\ell}(\bar{t}^{\ell}) & 0 \end{pmatrix} \begin{pmatrix} t_z^{\ell}(\bar{z}) \\ \alpha_z^{\ell}(\bar{z}) \end{pmatrix} = - \begin{pmatrix} g_{tz}(\bar{z}, \bar{t}^{\ell}) \\ 0 \end{pmatrix}$$

with

$$\alpha_z^{\ell}(\bar{z}) = \begin{pmatrix} \vdots \\ \alpha_z^{\ell,j}(\bar{z}) \\ \vdots \end{pmatrix}_{j \in J^{\ell}(\bar{z})}$$

and

$$H^{\ell}(\bar{t}^{\ell}) = \begin{pmatrix} \vdots \\ h_t^j(\bar{t}^{\ell}) \\ \vdots \end{pmatrix}_{j \in J^{\ell}(\bar{z})};$$

(iii) *the* (*local*) *marginal value function*

$$v^{\ell}(z) := g(z, t^{\ell}(z)) \tag{2.2}$$

is twice continuously differentiable on $U_{\bar{z}}^{\ell}$ *with derivatives*

$$v_z^{\ell}(z) = g_z(z, t^{\ell}(z)), \tag{2.3}$$

$$v^\ell_{zz}(z) = g_{zz}(z, t^\ell(z)) - (t^\ell_z(z))^T \mathcal{L}^\ell_{tt}(z, t^\ell(z), \alpha^\ell(z)) t^\ell_z(z). \tag{2.4}$$

The next theorem is easily proved by means of Theorem 2.2 and continuity arguments (cf. [12]). We first define the basic assumption.

Assumption 2.3. For given $\bar{z} \in \mathbb{R}^n$ let all local solutions $\bar{t}^\ell, \ell \in \bar{L}$ of (PAR($\bar{z}$)) be nondegenerate and let $\bar{L}$ be a finite set of indices

$$\bar{L} := \{1, 2, \ldots, n_c(\bar{z})\}. \tag{2.5}$$

Remark 2.4. Assumption 2.3 for $\bar{z} = z_*, z_*$ a (local) solution of (SIP), is one of the assumptions of Theorem 4.2. This theorem states the (local) superlinear convergence of a special algorithm from the class introduced in §3. If one is only interested in this theorem, it would be sufficient to assume only that all global solutions t^ℓ_* of (PAR(z_*)) (for which $g(z_*, t^\ell_*) = 0$) are nondegenerate, which implies that there are only finitely many of them, because B is compact and nondegenerate solutions are isolated. For numerical reasons it is necessary to take not only the global but also local solution t^ℓ of (PAR(z^i)) into account. Therefore, it is more appropriate to require Assumption 2.3.

THEOREM 2.5. *Suppose that Assumption* 2.3 *is valid for a given* $\bar{z} \in \mathbb{R}^n$. *Then, with* t^ℓ, v^ℓ *as in Theorem* 2.2 , *we have that*

(i) *there exists a neighborhood* $U(\bar{z})$ *of* $\bar{z}$ *such that for all* $z \in U(\bar{z})$ *the set of local solutions of* (PAR(z)) *is given by* $t^\ell(z), \quad \ell = 1, \ldots, n_c(\bar{z})$,

(ii) *a point* $z \in U(\bar{z})$ *is feasible for* (SIP) (*i.e.,* $z \in Z$) *if and only if*

$$v^\ell(z) \le 0, \quad \ell = 1, \ldots, n_c(\bar{z}).$$

As an immediate consequence of Theorem 2.5, in $U(\bar{z})$ we can replace (SIP) by the following "reduced" problem with finitely many constraints:

(SIP$_{\text{red}}(\bar{z})$) Maximize $F(z)$ subject to $z \in Z_{\text{red}}(\bar{z})$, where

$$Z_{\text{red}}(\bar{z}) := \{z \mid v^\ell(z) \le 0, \quad \ell = 1, \ldots, n_c(\bar{z})\}, \tag{2.6}$$

(see (2.2), (2.5) for the definitions of v^ℓ and $n_c(\bar{z})$).

If the $U^\ell_{\bar{z}}$ in Theorem 2.2 are chosen as large as possible, the set

$$U_{\max}(\bar{z}) := \bigcap_{\ell=1,\ldots,n_c(\bar{z})} U^\ell_{\bar{z}} \tag{2.7}$$

can be considered as the maximal set on which (SIP$_{\text{red}}(\bar{z})$) is defined. For $z \notin U_{\max}(\bar{z})$, at least one of the paths $t^\ell(z)$ is lost (or no longer describes a nondegenerate local solution).

It is helpful to keep the following facts in mind ($U(\bar{z})$ as in Theorem 2.5):

- $U(\bar{z}) \cap Z = U(\bar{z}) \cap Z_{\text{red}}(\bar{z})$, i.e., in $U(\bar{z})$ we have the equivalence of (SIP) and (SIP$_{\text{red}}(\bar{z})$);
- $U_{\max}(\bar{z}) \cap Z \subset U_{\max}(\bar{z}) \cap Z_{red}(\bar{z})$, i.e., consideration of $Z_{\text{red}}(\bar{z})$ in $U_{\max}(\bar{z}) \setminus U(\bar{z})$ may lead to points z infeasible for (SIP);
- for all $z \in U(\bar{z})$ the reduced problem (SIP$_{\text{red}}(z)$) is identical to (SIP$_{\text{red}}(\bar{z})$);
- for $z \in U_{\max}(\bar{z}) \setminus U(\bar{z})$, problem (SIP$_{\text{red}}(z)$) has at least one constraint more than (SIP$_{\text{red}}(\bar{z})$),

- for $z \notin U_{\max}(\bar{z})$, at least one constraint of $(\mathrm{SIP}_{\mathrm{red}}(\bar{z}))$ is no longer present in $(\mathrm{SIP}_{\mathrm{red}}(z))$.

Thus, intuitively we may think of $\mathbb{R}^n$ as a complicated quilt with different reduced problems $(\mathrm{SIP}_{\mathrm{red}}(\cdot))$ on the respective patches. Note, however, that there may be points that are not contained in $U = \bigcup\{U_{\max}(\bar{z}) | \bar{z} \in \mathbb{R}^n\}$, because U contains only points in $\mathbb{R}^n$ for which all local solutions of $(\mathrm{PAR}(z))$ are nondegenerate. However, it follows from results in [20], that points not in U may be considered as exceptional in some specified sense.

3. Reduction methods. The considerations in §2 give rise to the following method.

Conceptual Reduction Method

Given $\bar{z} = z^i \in \mathbb{R}^n$. Perform steps (i)–(iii).

(i) Determine the local solutions $\bar{t}^\ell, \ell = 1, \ldots, n_c(z^i)$, of the problem

$$(\mathrm{PAR}(z^i)): \text{ Maximize } g(z^i, t) \quad \text{subject to} \quad t \in B.$$

(ii) Starting with $z^{i,0} = z^i$ carry out k_i steps of a nonlinear programming (NLP) algorithm on the reduced problem

$$(\mathrm{SIP}_{\mathrm{red}}(z^i)): \text{ Maximize } F(z) \text{ s.t. } Yv^\ell(z) = g(z, t^\ell(z)) \leq 0, \ell = 1, \ldots, n_c(z^i).$$

Denote the iterates by $z^{i,1}, \ldots, z^{i,k_i}$.

(iii) Set $z^{i+1} = z^{i,k_i}$.

The following remarks are important in deriving implementable, efficient algorithms from this concept.

(a) In general, an expensive global search (i) on B is required to determine all local solutions of $(\mathrm{PAR}(z^i))$ and to be able to start a new problem $(\mathrm{SIP}_{\mathrm{red}}(z^i))$ which remains fixed during step (ii). The ideal strategy in choosing the number k_i of NLP steps in (ii) would take z^{i,k_i} as the last iterate in $U_{\max}(z^i)$ (cf. (2.7)) such that for no local solution $\tilde{t}$ of $(\mathrm{PAR}(z^{i,k_i}))$, $\tilde{t} \neq t^\ell(z^{i,k_i})$, $\ell = 1, \ldots, n_c(z^i)$, $g(z^{i,k_i}, \tilde{t}) > 0$, i.e., no "new" maxima $\tilde{t}$ of $g(z^{i,k_i}, t)$ lead to infeasibilities. However, the test that there is no $\tilde{t}$ with this property, requires just a step of type (i). Therefore, in [8], k_i is taken as the largest number less or equal to a fixed $k_{\max}$ for which $t^\ell(z^{i,k_i})$ can be identified. If in the following, step (i) $z^{i+1} = z^{i,k_i}$ proves to have infeasibilities in local solutions different from the $t^\ell(z^{i,k_i})$, $\ell = 1, \ldots, n_c(z^i)$, backtracking to a former z^{i,k_i-j} is implemented to avoid serious infeasibilities. This is done in a way that a global merit function can be defined that is reduced in every step and forms the basis of global convergence results (cf. [8] and remark (d) below).

(b) Note that in (ii) the paths $t^\ell(z)$, $\ell = 1, \ldots, n_c(z^i)$, must be followed along $z^{i,j}$, $j = 0, 1, \ldots, k_i$. Contrary to (i), this requires only local continuation starting from the $\bar{t}^\ell$, which can be done much more efficiently than a global search. We discuss the question below as to what accuracy $t^\ell(z)$ should be traced in connection with a specific NLP algorithm to preserve the rate of convergence to the solution of (SIP).

(c) In principle, any NLP method could be used in step (ii). However, with regard to an efficient control of k_i (cf.(a)), it is convenient to use algorithms with a superlinear rate of convergence. It is interesting to note that in the literature only SQP

methods have been considered up to now. In [12], [18], [15], [3], [17], Wilson's method is proposed with explicit use of second order derivatives. The SQP (augmented) Lagrangian secant method which will also be considered in the remaining sections of this paper is used in [14], [13], [6], and [8].

(d) To obtain global convergence for the type of method considered above, the following must be assumed.

(1) In each point of iteration $(z^i; z^{i,j})$ Assumption 2.3 holds.

(2) After a finite number of steps, the reduced problems $(\mathrm{SIP}_{\mathrm{red}}(z^i))$ no longer change (i.e., no $t^\ell(z)$ must be deleted or added).

Then global convergence results from finite programming can be applied in an obvious way. These usually depend on the construction of a merit function that is reduced from step to step. In [18] assumption (2) is missing; therefore it could happen that the reduced problem and, accordingly, the merit function needs to be changed infinitely often, destroying global convergence. The algorithm with the backtracking strategy considered in [8] allows to define a global merit function without the assumption of (2).

To assume (1) seems inevitable. Under strong global conditions on the problem, (1) can be warranted for every starting point. These conditions allow a "global finite description of Z" by finitely many constraints $\tilde{v}^\ell(z) \leq 0$, where the $\tilde{v}^\ell$ are appropriate extensions of v^ℓ in the reduced problems (cf. [9]).

4. The reduction method with the SQP augmented Lagrangian BFGS method as NLP solver. Now we will give a more explicit version of a reduction algorithm by specifying step (ii) in the conceptional method of §2.

We will use the augmented Lagrangian of the reduced problem $(\mathrm{SIP}_{\mathrm{red}}(\bar{z}))$ with $\bar{z} = z^i$:

$$L^{\bar{z}}(z, \lambda, c) := F(z) + \sum_{\ell=1}^{n_c(\bar{z})} \lambda_\ell v^\ell(z) + \frac{c}{2} \sum_{\ell=1}^{n_c(\bar{z})} (v^\ell(z))^2.$$

Then step (ii) with the SQP augmented Lagrangian BFGS method is carried out as follows:

Given $z^{i,0} = z^i, B_{i,0} = B_i \in \mathbb{R}^{n \times n}$ negative definite.

Then, for $j = 1, \ldots, k_i$ do:

Given $z^{i,j-1}, B_{i,j-1}$, perform substeps (ii_1)–(ii_3):

(ii_1) Compute a solution s^j and multipliers $\lambda^{i,j}$ of the quadratic programming problem:

Maximize $F_z(z^{i,j-1})^T s + \frac{1}{2} s^T B_{i,j-1} s$

subject to $v^\ell(z^{i,j-1}) + v_z^\ell(z^{i,j-1})^T s \leq 0, \quad \ell = 1, \ldots, n_c(z^i)$.

(ii_2) Compute a steplength α_j (see below).

(ii_3) Update with the BFGS update formula

$$z^{i,j} = z^{i,j-1} + \alpha_j s^j, \quad B_{i,j} = B + \frac{y^j (y^j)^T}{(y^j)^T s} - \frac{B_{i,j-1} s^j (B_{i,j-1} s^j)^T}{(s^j)^T B_{i,j-1} s^j},$$

where

$$y^j = L_z^{z^i}(z^{i,j}, \lambda^{i,j}, c) - L_z^{z^i}(z^{i,j-1}, \lambda^{i,j}, c).$$

Set $$B_{i+1} = B_{i,k_i}, \quad z^{i+1} = z^{i,k_i}.$$

Since this paper deals mainly with local convergence results, we refer to common strategies for a determination of a steplength α_j (see, for instance, [2]). It is important that a strategy is used avoiding the Maratos effect, i.e., that in case of convergence $\alpha_j = 1$ is obtained for $i \geq i_0$ with some $i_0 \in \mathbb{N}$.

For the remainder of this section, assume that $z_* \in Z$ is a local solution of our problem (SIP). Let

$$A_* = \{\ell \in \{1, \ldots, n_c(z_*)\} \quad | \quad g(z_*, \bar{t}^\ell) = 0\}$$

denote the indices of those points $\bar{t}^\ell$ for which our constraint $g(z_*, t) \leq 0$ is active.

We require the following second order sufficient optimality condition to hold in z_* [19], [10].

Assumption 4.1. Let Assumption 2.3 hold at z_*. Let the gradients

$$v_z^\ell(z_*) = g_z(z_*, \bar{t}^\ell), \qquad \ell \in A_*,$$

be linearly independent, and assume that there are $\bar{\lambda}_{*,\ell} > 0$, $\ell \in A_*$, such that (for v_{zz}^ℓ cf. (2.4))

$$F_z(z_*) - \sum_{\ell \in A} \bar{\lambda}_{*,\ell} v_z^\ell(z_*) = 0$$

and

$$\xi^T \left(F_{zz}(z_*) - \sum_{\ell \in A} \bar{\lambda}_{*,\ell} v_{zz}^\ell(z_*) \right) \xi < 0$$

for all $\xi \neq 0$ with

$$\xi^T v_z^\ell(z_*) = 0, \qquad \ell \in A_*.$$

Note, that Assumption 4.1 is also sufficient for z_* to be an isolated local solution of the following finite, equality-constrained problem:

$(\mathrm{SIP}_{\mathrm{red}}^{=}(z_*))$ Maximize $F(z)$ subject to $v^\ell(z) = 0, \quad \ell \in A_*.$

With the same arguments as in common SQP theory we make the following conclusion from this observation.

For z^i in a small neighborhood of z_*, the reduction method with step (ii) as above will generate the same iterates as the SQP augmented Lagrangian BFGS method (cf. [5]) when applied to $(\mathrm{SIP}_{\mathrm{red}}^{=}(z_*))$. The general step of the latter is again given by substeps (ii_1)–(ii_3) (iterates properly renumbered) with equality constraints

$$v^\ell(z^{i,j-1}) + v_z^\ell(z^{i,j-1})^T s = 0, \quad \ell \in A_*,$$

instead of inequalities in (ii_1).

Therefore, the following theorem is an immediate consequence of results in [5].

THEOREM 4.2. *Suppose that Assumption* 4.1 *holds and c has been chosen such that $L^{z_*}(z_*, \bar{\lambda}_*, c)$ is negative definite. Then the reduction method with the* SQP *algorithm presented at the beginning of this section as* NLP *solver produces sequences*

$$\left\{ \begin{pmatrix} z^{i,j} \\ \lambda^{i,j} \end{pmatrix} \right\} \text{ and } \{z^{i,j}\} \text{ which converge q-superlinearly to } \begin{pmatrix} z_* \\ \bar{\lambda}_* \end{pmatrix} \text{ and } z_*, \text{ resp.}$$

5. SQP under perturbations. We consider the following reduced semi-infinite programming problem $(\mathrm{SIP}^{=}_{\mathrm{red}}(z_*))$, where z_* is a local solution of (SIP) which satisfies Assumption 4.1:

$$\text{Maximize } F(z) \text{ s. t. } v^l(z) = g(z, t^l(z)) = 0 \text{ for all } l \in A_*, \tag{5.1}$$

where

$$F : \mathbb{R}^n \to \mathbb{R}, \quad t^l : D \subset \mathbb{R}^n \to B \subset \mathbb{R}^m, \quad g : \mathbb{R}^{n+m} \to \mathbb{R},$$

and D is a proper neighborhood of z_* (chosen according to Lemma 5.2 below).

For convenience we assume w.l.o.g. $A_* = \{1, \dots, r\}$ and define

$$v : \mathbb{R}^n \to \mathbb{R}^r \text{ by } v = (v^1, \dots, v^r)^T.$$

We assume further that F and v are twice continuously differentiable functions with Lipschitz-continuous derivatives and introduce the following notation: Let $L : \mathbb{R}^{n+r+1} \to \mathbb{R}$ denote the augmented Lagrangian

$$L(z, \lambda, c) := L^{z_*}(z, \lambda, c) = F(z) + \lambda^T v(z) + \frac{c}{2} \|v(z)\|^2,$$

where $\|\cdot\|$ is always the Euclidean norm.

The *Diagonalized Multiplier Method* by Tapia is formulated as follows.

ALGORITHM 5.1. For given $z \in \mathbb{R}^n, \lambda \in \mathbb{R}^r, B \in \mathbb{R}^{n \times n}$

- set $\lambda_+ = \mathcal{U}(z, \lambda, B)$,
- solve $Bs = -L_z(z, \lambda_+, c)$,
- set $z_+ = z + s$,
- set $B_+ = \mathcal{B}(z, \lambda, B)$, where

$$\mathcal{U} : \mathbb{R}^n \times \mathbb{R}^r \times \mathbb{R}^{n \times n} \to \mathbb{R}^r \text{ and } \mathcal{B} : \mathbb{R}^n \times \mathbb{R}^r \times \mathbb{R}^{n \times n} \to \mathbb{R}^{n \times n}$$

are properly defined mappings.

Recall that by Assumption 4.1 we have the following lemma.

LEMMA 5.2. *Let z_* be a local solution of* (5.1) *and suppose that Assumption* 4.1 *holds. Then*

(i) *the functions f and v are twice continuously differentiable in a neighborhood D of z_*,*

(ii) *$v_z(z_*)$ has full rank,*

(iii) *$\zeta^T L_{zz}(z_*, \lambda_*, c)\zeta < 0$ for all $\zeta \in \mathbb{R}^n \setminus \{0\}$ with $v_z(z_*)^T \zeta = 0$ and $c \geq 0$.*

Under these assumptions, the following update formulas (i.e., mappings $\mathcal{U}, \mathcal{B}$) are well defined. For $\mathcal{U}$ we choose the Newton multiplier update (with B nonsingular):

$$\begin{aligned}\mathcal{U}(z,\lambda,B) &:= \lambda + (v_z(z)^T B^{-1} v_z(z))^{-1}(v(z) - v_z(z)^T B^{-1} L_z(z,\lambda,c)) \\ &= (v_z(z)^T B^{-1} v_z(z))^{-1}(v(z) - v_z(z)^T B^{-1}(F_z(z) + c v_z(z) v(z))) \\ &= \mathcal{U}(z,B).\end{aligned}$$

In this case $\mathcal{U}$ depends only on z and B. The BFGS formula then can also be redefined as a mapping depending only on z and B:

$$\begin{aligned}\mathcal{B}(z,B) &:= B + \frac{yy^T}{y^T s} - \frac{(Bs)(Bs)^T}{s^T B s}, \\ s = s(z,B) &:= -B^{-1} L_z(z, \mathcal{U}(z,B), c), \\ y = y(z,B) &:= L_z(z+s, \mathcal{U}(z,B), c) - L_z(z, \mathcal{U}(z,B), c).\end{aligned}$$

In [5] a complete local convergence analysis of this method has been given. In the case of semi-infinite programming the computation of $v(z)$ for given z requires an iterative procedure that in itself can become quite costly. Therefore it is desirable to carry out these inner iterations only to an accuracy necessary to maintain the overall convergence properties. Our goal is to control these errors so that we still maintain a superlinear rate of convergence.

For each z we denote by $\tilde{t}^l(z)$ the perturbed value of $t^l(z)$. In the algorithm, $t^l(z)$ enters through the equality constraint $v^l(z) = g(z, t^l(z)) = 0$ such that we set

$$\tilde{v}^l(z) := g(z, \tilde{t}^l(z)). \tag{5.2}$$

Note that it is not reasonable to assume that $\tilde{t}^l(z)$ is differentiable. Therefore the derviative of v^l also needs to be approximated. Observe that by (2.3)

$$v_z^l(z) = g_z(z, t^l(z)) \tag{5.3}$$

holds. Hence we approximate it by $d\tilde{v}^l(\cdot) : \mathbb{R}^n \to \mathbb{R}^n$

$$d\tilde{v}^l(z) := g_z(z, \tilde{t}^l(z)). \tag{5.4}$$

In the sequel we assume for simplicity that we have only one element in A_*, i.e., there is only one t^ℓ and one v^ℓ and we omit the index ℓ. The validity of the following results for the general case is obvious.

The smoothness assumptions on g yield for $z \in D$ and some $\kappa_1 > 0$

$$\|\tilde{v}(z) - v(z)\| \le \kappa_1 \|\tilde{t}(z) - t(z)\|^2, \tag{5.5}$$

$$\|d\tilde{v}(z) - v_z(z)\| \le \kappa_1 \|\tilde{t}(z) - t(z)\|. \tag{5.6}$$

If one replaces the corresponding quantitities in Algorithm 5.1 by its approximations, then we obtain the following algorithm.

ALGORITHM 5.3. For given $z \in \mathbb{R}^n, \lambda \in \mathbb{R}^r, B \in \mathbb{R}^{n\times n}$ regular,

- set $\lambda_+ = (d\tilde{v}(z)^T B^{-1} d\tilde{v}(z))^{-1}(\tilde{v}(z) - d\tilde{v}(z)^T B^{-1}(F_z(z) + c\, d\tilde{v}(z)\tilde{v}(z)))$,

- solve $B\tilde{s} = -F_z(z) - d\tilde{v}(z)\lambda_+ - c\, d\tilde{v}(z)\tilde{v}(z)$,
- set $z_+ = z + \tilde{s}$,
- set $\tilde{y} = F_z(z_+) - F_z(z) + (d\tilde{v}(z_+) - d\tilde{v}(z))\lambda_+ + c(d\tilde{v}(z_+)\tilde{v}(z_+) - d\tilde{v}(z)\tilde{v}(z))$,
- set $B_+ = B + \dfrac{\tilde{y}\tilde{y}^T}{\tilde{y}^T\tilde{s}} - \dfrac{(B\tilde{s})(B\tilde{s})^T}{\tilde{s}^T B\tilde{s}}$.

In order to use the framework developed in [5], we define perturbed update mappings (note: $r = 1$)

$$\tilde{\mathcal{U}} : \mathbb{R}^n \times \mathbb{R}^{n\times n} \to \mathbb{R}^r \quad \text{and} \quad \tilde{\mathcal{B}} : \mathbb{R}^n \times \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$$

for the Lagrange multiplier and the approximation of the Hessian. Since in [5] conditions on the update maps are given that do not include any smoothness conditions, we can directly apply their theorems to deduce convergence properties. Set

$$\tilde{\mathcal{U}}(z, B) = (d\tilde{v}(z)^T B^{-1} d\tilde{v}(z))^{-1}(\tilde{v}(z) - d\tilde{v}(z)^T B^{-1}(F_z(z) + c\, d\tilde{v}(z)\tilde{v}(z)),$$

$$\tilde{\mathcal{B}}(z, B) := B + \frac{\tilde{y}\tilde{y}}{\tilde{y}\tilde{s}} - \frac{(B\tilde{s})(B\tilde{s})^T}{\tilde{s}^T B\tilde{s}},$$

where $\tilde{s}, \tilde{y}$ are defined by

$$\begin{aligned}\tilde{y} = \tilde{y}(z, B) :=& F_z(z + \tilde{s}) + d\tilde{v}(z + \tilde{s})\tilde{\mathcal{U}}(z, B) + cd\tilde{v}(z + \tilde{s})\tilde{v}(z + \tilde{s}) \\ &- F_z(z) - d\tilde{v}(z)\tilde{\mathcal{U}}(z, B) - cd\tilde{v}(z)\tilde{v}(z), \\ \tilde{s} = \tilde{s}(z, B) :=& -B^{-1}(F_z(z) + d\tilde{v}(z)\tilde{\mathcal{U}}(z, B) + c\, d\tilde{v}(z)\tilde{v}(z)).\end{aligned}$$

Notice that in addition to the generality in [5], where $\mathcal{U}$ and $\mathcal{B}$ are given by update formulas, we also need to approximate the derivative of the augmented Lagrangian in the second step of Algorithm 5.1. We denote the approximation of $L_z(z, \lambda, c)$ by

$$d\tilde{L}(\cdot, \cdot, \cdot) : \mathbb{R}^n \times \mathbb{R}^r \times \mathbb{R} \to \mathbb{R}^n$$

and

$$d\tilde{L}(z, \lambda, c) := F_z(z) + d\tilde{v}(z)\lambda + cd\tilde{v}(z)\tilde{v}(z).$$

In the following definition we list the assumptions on the update formulas.

DEFINITION 5.4. (i) $\tilde{\mathcal{B}}$ *is of* bounded deterioration *if there exist constants* α_1 *and* α_2 *such that for each* (z, B) *in a neighborhood* N *of* $(z_*, L_{zz}(z_*, \lambda_*, c))$ *and for*

$$\begin{aligned}\lambda_+ &= \mathcal{U}(z, B), \\ z_+ &= z - B^{-1}L_z(z, \lambda_+, c), \\ B_+ &= \mathcal{B}(z, B),\end{aligned} \tag{5.7}$$

we have

$$\|B_+ - L_{zz}(z_*, \lambda_*, c)\| \le (1 + \alpha_1\tilde{\sigma})\|B - L_{zz}(z_*, \lambda_*, c)\| + \alpha_2\tilde{\sigma}$$

with

$$\tilde{\sigma} = \max\{\|z - z_*\|, \|z_+ - z_*\|, \|\lambda_+ - \lambda_*\|\}.$$

(ii) $\tilde{\mathcal{U}}$ *is* z-dominated *if there exists a constant* $\phi(c) < 1$ *such that for all* $(z, B) \in N$ *we have*

$$\|L_{zz}(z_*, \lambda_*, c)^{-1}v_z(z_*)(\lambda_+ - \lambda_*)\| \le \phi(c)\|z - z_*\|.$$

(iii) $d\tilde{L}$ *is* consistent *at* z_* *if there is* $\epsilon(c) > 0$ *such that for all* z *with* $\|z - z_*\| \le \epsilon(c)$ *and all* λ *we have*

$$\begin{aligned} &\|d\tilde{L}(z,\lambda,c) - d\tilde{L}(z_*,\lambda,c) - L_{zz}(z_*,\lambda_*,c)(z-z_*)\| \\ &= o(\|z-z_*\|) + O(\|z-z_*\|\|\lambda-\lambda_*\|). \end{aligned}$$

Note that (ii) and Lemma 5.2 imply that for some constant κ_2 and all $(z,B) \in N$,

$$\|\lambda_+ - \lambda_*\| \le \kappa_2 \|z - z_*\|. \tag{5.8}$$

The following theorem is an extended version of Theorem 3.1 of [5] which takes into account inexact data in the right-hand side of the equation to be solved at each iteration. We point out that the regularity of $L_{zz}(z_*,\lambda_*,0)$, which is part of Assumption A4 in [5], is not needed in this proof. The following theorem shows that under appropriate conditions linear convergence with arbitrarily small rate factor $\rho \in (0,1)$ is obtained.

THEOREM 5.5. *For given* $z^j \in \mathbb{R}^n, \lambda^j \in \mathbb{R}^r, B_j \in \mathbb{R}^{n\times n}$

- *set* $\lambda^{j+1} = \mathcal{U}(z^j, B_j)$,
- *solve* $B_j s^j = -d\tilde{L}(z^j, \lambda^{j+1}, c)$,
- *set* $z^{j+1} = z^j + s^j$,
- *set* $B_{j+1} = \mathcal{B}(z^j, B_j)$.

Let Assumption 4.1 *hold and let* $L_{zz}(z_*,\lambda_*,c)$ *be negative definite. Assume that* $\mathcal{B}$ *is of bounded deterioration,* $\mathcal{U}$ *is* z*-dominated, and* $d\tilde{L}$ *is consistent at* z_*. *For each* $\rho \in (0,1)$ *there exists* $\epsilon(\rho) > 0$ *such that for all* z_0, λ_0, B_0 *with*

$$\|z_0 - z_*\| \le \epsilon(\rho), \quad \|\lambda_0 - \lambda_*\| \le \epsilon(\rho), \quad \|B_0 - L_{zz}(z_*,\lambda_*,c)\| \le \epsilon(\rho)$$

the sequence (z^j, λ^j) *generated by this algorithm is well defined and converges to* (z_*, λ_*) *with a linear rate of convergence for* z^j *according to*

$$\|z^{j+1} - z_*\| \le \rho \|z^j - z_*\|.$$

Moreover, B_j *and* $(B_j)^{-1}$ *are bounded.*

The proof is similar to the one given in [5] if one chooses ϵ_x properly. In order to obtain in [5] line (3.12) from (3.7) we can use the consistency of $d\tilde{L}$ instead of Lemma 2.3 in [5].

6. Convergence proof. In this section we use Theorem 5.5 and results in [5] to establish the linear convergence and then use a result of [1] to deduce superlinear convergence. Recall that in (5.5) and (5.6) we estimated the error in the constraint and its derivative by the error made in the evaluation of the maxima $t^l(z)$ of $g(z,t)$ on B. We make the following assumption on the error. In order to establish a fast rate of convergence we assume that the error for $\tilde{t}(z)$ is reduced at a rate that depends on the distance from z to the solution z_*.

Assumption 6.1. For all $z \in D$ we assume that

$$\|\tilde{t}(z) - t(z)\| = o(\|z - z_*\|).$$

This implies in particular that

$$\tilde{t}(z_*) = t(z_*), \tag{6.1}$$

and with (5.2) and (5.4)

$$\tilde{v}(z_*) = v(z_*) \quad \text{and} \quad \tilde{v}(z_*) = v_z(z_*). \tag{6.2}$$

LEMMA 6.2. *Let Assumptions* 4.1 *and* 6.1 *hold and let* $L_{zz}(z_*, \lambda_*, c)$ *be negative definite. Then the update for the Lagrange multipliers* $\tilde{\mathcal{U}}$ *is z-dominated.*

Proof. Using (6.1), (6.2), and the Kuhn–Tucker conditions, for any regular matrix $B \in \mathbb{R}^{n\times n}$, we have

$$\begin{aligned} \tilde{\mathcal{U}}(z_*, B) &= (v_z(z_*)^T B^{-1} v_z(z_*))^{-1}(v(z_*) - v_z(z_*)^T B^{-1} F_z(z_*)) \\ &= \lambda_* = \mathcal{U}(z_*, B). \end{aligned} \tag{6.3}$$

Hence

$$\lambda_+ - \lambda_* = \tilde{\mathcal{U}}(z, B) - \mathcal{U}(z_*, B). \tag{6.4}$$

Using the notation

$$\begin{aligned} R(z) &:= v_z(z)^T B^{-1} v_z(z), \quad \tilde{R}(z) := d\tilde{v}(z)^T B^{-1} d\tilde{v}(z), \\ r(z) &:= v(z) - v_z(z)^T B^{-1}(F_z(z) + c\, v_z(z)v(z))), \\ \tilde{r}(z) &:= \tilde{v}(z) - d\tilde{v}(z)^T B^{-1}(F_z(z) + c\, d\tilde{v}(z)\tilde{v}(z)), \end{aligned}$$

we can show with (5.5) and (5.6) that for some constant κ_3 and all $z \in D$

$$\max\{\|\tilde{R}(z) - R(z)\|, \|\tilde{r}(z) - r(z)\|\} \le \kappa_3 \|\tilde{t}(z) - t(z)\| \tag{6.5}$$

holds. We can write

$$\mathcal{U}(z, B) = R(z)^{-1} r(z), \quad \tilde{\mathcal{U}}(z, B) = \tilde{R}(z)^{-1}\tilde{r}(z).$$

With (6.5) we can estimate with some constant κ_4 for any $z \in D$ and B in a ϵ-neighborhood of $L_{zz}(z_*, \lambda_*, c)$

$$\begin{aligned} &\|\tilde{\mathcal{U}}(z, B) - \mathcal{U}(z, B)\| \\ &= \|\tilde{R}(z)^{-1}\tilde{r}(z) - R(z)^{-1} r(z)\| \\ &= \|\tilde{R}(z)^{-1}(\tilde{r}(z) - r(z)) - R(z)^{-1}(\tilde{R}(z) - R(z))\tilde{R}(z)^{-1} r(z)\| \\ &\le \kappa_4 \|\tilde{t}(z) - t(z)\|. \end{aligned} \tag{6.6}$$

From [5], Proposition 4.2, it follows that the Newton multiplier update $\mathcal{U}$ is z-dominated in a neighborhood of $(z_*, L_{zz}(z_*, \lambda_*, c))$. Hence for some $\phi(c) < 1$, we have

$$\|L_{zz}(z_*, \lambda_*, c)^{-1} v_z(z_*)(\mathcal{U}(z, B) - \mathcal{U}(z_*, B))\| \le \phi(c)\|z - z_*\|. \tag{6.7}$$

We can deduce from (6.6) and Assumption 6.1 that for a sufficiently small neighborhood around z_*

$$\|L_{zz}(z_*, \lambda_*, c)^{-1} v_z(z_*)(\tilde{\mathcal{U}}(z, B) - \mathcal{U}(z, B))\| \le \frac{1 - \phi(c)}{2}\|z - z_*\|. \tag{6.8}$$

Both (6.7) and (6.8) applied to (6.4) yield

$$\begin{aligned} &\|L_{zz}(z_*, \lambda_*, c)^{-1} v_z(z_*)(\lambda_+ - \lambda_*)\| \\ = \;& \|L_{zz}(z_*, \lambda_*, c)^{-1} v_z(z_*)\,(\tilde{\mathcal{U}}(z, B) - \mathcal{U}(z, B) + \mathcal{U}(z, B) - \mathcal{U}(z_*, B))\| \\ \le \;& \tilde{\phi}(c)\|z - z_*\|, \end{aligned}$$

where

$$\tilde{\phi}(c) = \phi(c) + (1 - \phi(c))/2 = (1 + \phi(c))/2 < 1,$$

which proves that the perturbed update $\tilde{\mathcal{U}}$ is z-dominated. □

LEMMA 6.3. *Let Assumptions* 4.1 *and* 6.1 *hold. Then the approximation* $d\tilde{L}$ *for the gradient of the augmented Lagrangian is consistent.*

Proof. We estimate the quantity

$$\begin{aligned}
&D(z,\lambda,c)\\
&:= \|d\tilde{L}(z,\lambda,c) - d\tilde{L}(z_*,\lambda,c) - L_{zz}(z_*,\lambda_*,c)(z-z_*)\|\\
&= \|F_z(z) + d\tilde{v}(z)\lambda + cd\tilde{v}(z)\tilde{v}(z) - F_z(z_*) - d\tilde{v}(z_*)\lambda - cd\tilde{v}(z_*)\tilde{v}(z_*)\\
&\quad - F_{zz}(z_*)(z-z_*) - (v_z(z_*)\lambda_*)_z(z-z_*) - c(v_z(z_*)v(z_*))_z(z-z_*)\|\\
&\le \|F_z(z) - F_z(z_*) - F_{zz}(z_*)(z-z_*)\| + c\|d\tilde{v}(z)\tilde{v}(z) - v_z(z)v(z)\|\\
&\quad + c\|v_z(z)v(z) - v_z(z_*)v(z_*) - (v_z(z_*)v(z_*))_z(z-z_*)\| + \|(d\tilde{v}(z) - v_z(z))\lambda_*\|\\
&\quad + \|(d\tilde{v}(z) - v_z(z))(\lambda-\lambda_*)\| + \|(v_z(z) - v_z(z_*))(\lambda-\lambda_*)\|\\
&\quad + \|(v_z(z) - v_z(z_*))\lambda_* - (v_z(z_*)\lambda_*)_z(z-z_*)\|.
\end{aligned}$$

The terms involving second derivatives can be estimated with Assumption 4.1 by $O(\|z-z_*\|^2)$. All the others are by Assumption 6.1 of order $o(\|z-z_*\|)$, except for the second last term which is of order $O(\|z-z_*\|\|\lambda-\lambda_*\|)$. This altogether yields

$$D(z,\lambda,c) = o(\|z-z_*\|) + O(\|z-z_*\|\|\lambda-\lambda_*\|),$$

which shows that $d\tilde{L}$ is consistent. □

It remains to show the bounded deterioration property for the mapping $\tilde{\mathcal{B}}$. Since this concerns a statement about the precision with which the Hessian of the augmented Lagrangian is approximated, we impose an additional condition on the approximation $\tilde{t}$ of t which ensures that the difference quotient $(\tilde{t}(z+\tilde{s})-\tilde{t}(z))/\|\tilde{s}\|$ approaches $t_z(z_*)\tilde{s}$.

Assumption 6.4. For all $z \in D$, $z \neq z_*$ we assume that for all s sufficiently small and some constant κ_5

$$\|\tilde{t}(z+s) - t(z+s) - (\tilde{t}(z) - t(z))\| \le \kappa_5\sigma(z,s)\|s\|, \tag{6.9}$$

where we use the notation

$$\sigma(z,s) = \max\{\|z+s-z_*\|, \|z-z_*\|\}.$$

We note that (6.9) is equivalent to

$$\|\tilde{t}(z+s) - \tilde{t}(z) - t_z(z_*)s\| \le \kappa_6\sigma(z,s)\|s\|, \tag{6.10}$$

with a constant κ_6. This assumption implies a similar statement for a composition of functions.

LEMMA 6.5. *Let Assumptions* 2.3, 6.1, *and* 6.4 *hold. If* $\psi : \mathbb{R}^m \to \mathbb{R}^r$ *is continuously differentiable with Lipschitz-continuous derivative in a neighborhood of* $(z_*, t(z_*))$, *then Assumption* 6.4 *implies for*

$$p(z) = \psi(z,t(z)), \quad \tilde{p}(z) = \psi(z,\tilde{t}(z)) \tag{6.11}$$

that there exists $\kappa_7 > 0$ *with*

$$\|\tilde{p}(z+s) - \tilde{p}(z) - p'(z_*)s\| \le \kappa_7\sigma(z,s)\|s\|$$

for z close to z_* and s sufficiently small.

Proof. Set $a := \tilde{t}(z+s) - \tilde{t}(z) - t_z(z_*)s$. Then

$$\begin{aligned}
&\|\tilde{p}(z+s) - \tilde{p}(z) - p'(z_*)s\| \\
&= \|\psi(z+s, \tilde{t}(z) + t_z(z_*)s + a) - \psi(z, \tilde{t}(z)) - \psi_z(z_*, t(z_*))s - \psi_t(z_*, t(z_*))t_z(z_*)s\| \\
&\le \int_0^1 \|(\psi_z(z+\tau s, \tilde{t}(z) + \tau(t_z(z_*)s + a)) - \psi_z(z_*, t(z_*))s\| d\tau \\
&\quad + \int_0^1 \|(\psi_t(z+\tau s, \tilde{t}(z) + \tau(t_z(z_*)s + a)) - \psi_t(z_*, t(z_*)))t_z(z_*)s\| d\tau \\
&\quad + \int_0^1 \|\psi_t(z+\tau s, \tilde{t}(z) + \tau(t_z(z_*)s + a))a\| d\tau \\
&\le \gamma_1(\sigma(z,s) + \|\tilde{t}(z) - t(z_*)\| + \|t_z(z_*)s\| + \|a\|)(\|s\| + \|t_z(z_*)s\|) + \gamma_2\|a\|,
\end{aligned}$$

where γ_1 is a Lipschitz constant and γ_2 is a bound on ψ_t. Note that by (6.10) the inequality $\|a\| \le \kappa_6\sigma(z,s)\,\|s\|$ holds. Furthermore, observe $\|s\| \le 2\sigma(z,s)$ and that by Assumption 6.1

$$\|\tilde{t}(z) - t(z_*)\| \le \|\tilde{t}(z) - t(z)\| + \|t(z) - t(z_*)\| \le \sigma(z,s)$$

holds. From these inequalities altogether the statement of the lemma follows. □

The following inequality plays a key role for the linear and superlinear convergence proof.

LEMMA 6.6. *Let Assumptions* 4.1, 6.1, *and* 6.4 *hold. Then for all $z, \tilde{s} = \tilde{s}(z,B)$ as in Assumption* 6.4 *and $\tilde{y} = \tilde{y}(z,B)$*

$$\|\tilde{y} - L_{zz}(z_*, \lambda_*, c)\tilde{s}\| \le \kappa_8\sigma(z,\tilde{s})\|\tilde{s}\| \tag{6.12}$$

and for c sufficiently large

$$\tilde{y}^T\tilde{s} < 0. \tag{6.13}$$

Proof. In order to show (6.12) we estimate

$$\|\tilde{y} - L_{zz}(z_*, \lambda_*, c)\tilde{s}\| \le \eta^1 + \eta^2 + \eta^3 \tag{6.14}$$

with

$$\begin{aligned}
\eta^1 &= \|F_z(z+\tilde{s}) - F_z(z) - F_{zz}(z_*)\tilde{s}\|, \\
\eta^2 &= \|(d\tilde{v}(z+\tilde{s}) - d\tilde{v}(z))\tilde{\mathcal{U}}(z,B) - (v_z(z_*)\lambda_*)_z\tilde{s}\|, \\
\eta^3 &= c\|d\tilde{v}(z+\tilde{s})\tilde{v}(z+\tilde{s}) - d\tilde{v}(z)\tilde{v}(z) - (v_z(z_*)v(z_*))_z\tilde{s}\|.
\end{aligned}$$

Then by the mean value theorem

$$\eta^1 \le \kappa_9\sigma(z,\tilde{s})\|\tilde{s}\|. \tag{6.15}$$

With (6.3) we use $\lambda_* = \tilde{\mathcal{U}}(z_*, B)$ to derive

$$\begin{aligned}
\eta^2 \le &\|(d\tilde{v}(z+\tilde{s}) - d\tilde{v}(z) - v_{zz}(z_*)\tilde{s})\tilde{\mathcal{U}}(z,B)\| \\
&+ \|v_{zz}(z_*)\tilde{s}(\tilde{\mathcal{U}}(z_*,B) - \tilde{\mathcal{U}}(z,B))\|.
\end{aligned}$$

We use (5.8) to estimate the second term. For the first term we can apply Lemma 6.5 with $\psi(z,\cdot) = g_z(z,\cdot)$ and obtain

$$\eta^2 \le \kappa_{10}\sigma(z,\tilde{s})\|\tilde{s}\|. \tag{6.16}$$

We can again use Lemma 6.5 with $\psi(z,\cdot) = g_z(z,\cdot)g(z,\cdot)$ to estimate η^3 by

$$\eta^3 \leq \kappa_{11}\sigma(z,\tilde{s})\|\tilde{s}\|. \tag{6.17}$$

Hence (6.14)–(6.17) imply (6.12).

In order to show (6.13) observe that with (6.12) for $\kappa_8 > 0$

$$\begin{aligned}\tilde{y}^T\tilde{s} &= \tilde{s}^T L_{zz}(z_*,\lambda_*,c)\tilde{s} + (\tilde{y} - L_{zz}(z_*,\lambda_*,c)\tilde{s})^T\tilde{s} \\ &\leq \tilde{s}^T L_{zz}(z_*,\lambda_*,c)\tilde{s} + \kappa_8\sigma(z,\tilde{s})\|\tilde{s}\|^2,\end{aligned}$$

which is negative if c is large enough, z is sufficiently close to z_*, and $\|\tilde{s}\|$ is sufficiently small. □

LEMMA 6.7. *Let Assumptions* 4.1, 6.1, *and* 6.4 *hold. Then the update* $\mathcal{B}$ *for the approximations of the Hessian of the augmented Lagrangian fulfills the bounded deterioration property.*

Proof. Recall that for $\tilde{y}^j$ and $\tilde{s}^j$ the inequality (6.12) holds. It is well known; see e.g., Theorem 4 in [4], that this implies the bounded deterioration property. □

At this point all the requirements for linear convergence have been shown and we can apply Theorem 5.5 to derive the next theorem.

THEOREM 6.8. *Let Assumptions* 4.1, 6.1, *and* 6.4 *hold. For each* $\rho \in (0,1)$ *there exist* $\underline{c}(\rho), \epsilon(\rho) > 0$ *such that for all* c, z_0, λ_0, B_0 *with* $c \geq \underline{c}(\rho)$

$$\|z_0 - z_*\| \leq \epsilon(\rho), \quad \|\lambda_0 - \lambda_*\| \leq \epsilon(\rho), \quad \|B_0 - L_{zz}(z_*,\lambda_*,c)\| \leq \epsilon(\rho)$$

the sequence (z^j,λ^j) *generated by Algorithm* 5.3 *is well defined and converges to* (z_*,λ_*) *with a linear rate of convergence for* z^j *according to*

$$\|z^{j+1} - z_*\| \leq \rho\|z^j - z_*\|,$$

and B_j *and* $(B_j)^{-1}$ *are bounded.*

We can also deduce the superlinear convergence from Lemma 6.6.

THEOREM 6.9. *Let Assumptions* 4.1, 6.1, *and* 6.4 *hold. Then there exists* $\underline{c}, \epsilon > 0$ *such that for all* c, z_0, λ_0, B_0 *positive definite with* $c \geq \underline{c}$,

$$\|z_0 - z_*\| \leq \epsilon, \quad \|\lambda_0 - \lambda_*\| \leq \epsilon, \quad \|B_0 - L_{zz}(z_*,\lambda_*,c)\| \leq \epsilon$$

the sequence (z^j,λ^j) *generated by Algorithm* 5.3 *is well defined and converges to* (z_*,λ_*). *The rate of convergence for* z^j *is q-superlinear, i.e.,*

$$\lim_{j\to\infty} \frac{\|z^{j+1} - z_*\|}{\|z^j - z_*\|} = 0.$$

Proof. From Theorem 6.8 we can deduce the convergence and the linear rate of (z^j) converging to z_*. In order to invoke Theorem 3.2 in [1] we must prove that

$$\sum_{j=1}^{\infty} \frac{\|\tilde{y}^j - L_{zz}(z_*,\lambda_*,c)\tilde{s}^j\|}{\|\tilde{s}^j\|} < \infty \tag{6.18}$$

and

$$(\tilde{y}^j)^T\tilde{s}^j < 0. \tag{6.19}$$

We will use the fact that linear convergence holds, which implies that for some $\rho \in (0,1)$

$$\|z^{j+1} - z_*\| \leq \rho \|z^j - z_*\| \leq \rho^j \|z_0 - z_*\|.$$

From Lemma 6.6 we obtain

$$\sum_{j=1}^{\infty} \frac{\|\tilde{y}^j - L_{zz}(z_*, \lambda_*, c)\tilde{s}^j\|}{\|\tilde{s}^j\|} \leq \sum_{j=1}^{\infty} \rho^j \|z_0 - z_*\| < \infty.$$

Due to Theorem 3.2 in [1], (6.18) and (6.19) imply the Dennis–Morè condition

$$\lim_{j\to\infty} \frac{\|(B_j - L_{zz}(z_*, \lambda_*, c))\tilde{s}^j\|}{\|\tilde{s}^j\|} = 0. \tag{6.20}$$

In order to prove the superlinear convergence we note that (6.20) implies the Boggs–Tolle–Wang condition

$$\lim_{j\to\infty} \frac{\|P_j(B_j - L_{zz}(z_*, \lambda_*, c))\tilde{s}^j\|}{\|\tilde{s}^j\|} = 0,$$

where the projection P_j is defined as

$$P_j := I - v_z(z^j)(v_z(z^j)^T v_z(z^j))^{-1} v_z(z^j)^T.$$

According to Theorem 5.3 in [5] the superlinear rate of convergence is shown if

$$\lim_{j\to\infty} \frac{\|v(z^j) + v_z(z^j)^T \tilde{s}^j\|}{\|\tilde{s}^j\|} = 0.$$

This is true because the linear convergence yields

$$(1-\rho)\|z^j - z_*\| \leq \|\tilde{s}^j\| \leq (1+\rho)\|z^j - z_*\|$$

and $d\tilde{v}(z^j)^T \tilde{s}^j = -\tilde{v}(z^j)$ implies

$$\begin{aligned} &\|v(z^j) + v_z(z^j)^T \tilde{s}^j\| \\ &\leq \|v(z^j) - \tilde{v}(z^j)\| + \|(v_z(z^j) - d\tilde{v}(z^j))^T \tilde{s}^j\| \\ &= o(\|z^j - z_*\|) + O(\|z^j - z_*\|\|\tilde{s}^j\|) = o(\|\tilde{s}^j\|). \end{aligned}$$ □

Acknowledgment. The authors are grateful for the careful refereeing that led to a substantially improved version of the paper.

REFERENCES

[1] R. H. Byrd and J. Nocedal, *A tool for the analysis of quasi-Newton methods with application to unconstrained minimization*, SIAM J. Numer. Anal., 26 (1989), pp. 727–739.

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

[3] I. D. COOPE AND G. A. WATSON, *A projected Lagrangian algorithm for semi-infinite programming*, Math. Programming, 32 (1987), pp. 337–356.
[4] J. R. ENGELS AND H. J. MARTINEZ, *Local and superlinear convergence for partially known quasi-Newton methods*, SIAM J. Optim., 1 (1991), pp. 42 –56.
[5] R. FONTECILLA, T. STEIHAUG, AND R. A. TAPIA, *A convergence theory for a class of quasi-Newton methods for constrained optimization*, SIAM J. Numer. Anal., 24 (1987), pp. 1133–1151.
[6] G. M. GRAMLICH, SQP-*Methoden für semiinfinite Optimierungsprobleme*, Ph.D. thesis, Universität Trier, 1990.
[7] E. HAAREN-RETAGNE, *Robot trajectory planning with semi-infinite programming*, Tech. report, Universität Trier, FB IV – Mathematik, 1991.
[8] ———, *Semi-infinite programming algorithm for robot trajectory planning*, Ph.D. thesis, Universität Trier, 1992.
[9] E. HAAREN-RETAGNE AND H. U. KALEX, *On global finite representations of semi-infinite programming problems*, Universität Trier, FB IV – Mathematik, 1991, working paper.
[10] R. HETTICH AND H. T. JONGEN, *Semi-infinite programming: conditions of optimality and applications*, in Optim. Techniques, Part 2, J. Stoer, ed., Springer-Verlag, 1978, pp. 1–11.
[11] R. HETTICH AND G. STILL, *Second order conditions for semi-infinite optimization*, Optimization, to appear.
[12] R. HETTICH AND W. VAN HONSTEDE, *On quadratically convergent methods for semi-infinite programming*, in Semi-Infinite Programming, R. Hettich, ed., Springer-Verlag, 1979, pp. 97–111.
[13] M. HUTH, *Superlinear konvergente Verfahren zur Lösung semiinifiniter Optimierungsaufgaben*, Ph.D. thesis, Pädagogische Hochschule Halle, 1987.
[14] K. OETTERSHAGEN, *Ein superlinear konvergenter Algorithmus zur Lösung semi-inifiniter Optimierungsprobleme*, Ph.D. thesis, Universität Bonn, 1982.
[15] E. POLAK AND A. L. TITS, *A recursive quadratic programming algorithm for semi-infinite optimization problems*, Appl. Math. Optim., 8 (1982), pp. 325–349.
[16] A. SHAPIRO, *Second-order derivatives of extremal-value functions and optimality conditions for semi-infinite programs*, Math. of Oper. Res., 10 (1985), pp. 207–219.
[17] Y. TANAKA, M. FUKUSHIMA, AND T. IBARAKI, *A globally convergent* SQP *method for semi-infinite nonlinear optimization*, J. Comput. Appl. Math., 23 (1988), pp. 141–153.
[18] G. A. WATSON, *Globally convergent methods for semi-infinite programming*, BIT, 21 (1981), pp. 362–373.
[19] W. WETTERLING, *Definitheitsbedingungen für relative Extrema bei Optimierungs- und Approximationsaufgaben*, Numer. Math., 15 (1970), pp. 122–136.
[20] G. ZWIER, *Structural analysis in semi-infinite programming*, Ph.D. thesis, University of Twente, Enschede, 1987.

SIAM J. OPTIMIZATION
Vol. 5, No. 3, pp. 659–669, August 1995

009

TAYLOR'S FORMULA FOR $C^{k,1}$ FUNCTIONS*

DINH THE LUC†

Abstract. In this paper, using Clarke's generalized Jacobian we establish Taylor's formula for functions whose kth order derivatives are locally Lipschitz. A calculus rule for generalized Hessian of implicit functions is presented. The results are then applied to derive high-order optimality conditions and second-order characterizations of quasiconvex functions.

Key words. Taylor's formula, generalized Jacobian, mean value theorem, inverse function theorem, implicit function theorem, quasiconvex function

AMS subject classifications. 26B25, 49J52

1. Introduction.

Let f be a $(k+1)$-time differentiable function from R^n to R. The classical Taylor theorem states that for every couple of points $a, b \in R^n$, there can be found a point c in the open interval (a, b) such that

$$f(b) - f(a) = \sum_{i=1}^{k} \frac{1}{i!} D^i f(a)(b-a, \ldots, b-a) + \frac{1}{(k+1)!} D^{k+1} f(c)(b-a, \ldots, b-a),$$

where $D^i f(a)$ is the ith order derivative of f at a. This is one of the fundamental formulas of classical analysis that is frequently used in applied mathematics. The formula with $k = 0$, known as mean value theorem, is of particular interest. In recent years several extensions of this formula (with $k = 0$) have been developed for larger classes of functions due to the introduction of new concepts of generalized derivatives in nonsmooth analysis. For instance, Wegge's mean value theorem [34] was established for convex functions using convex analysis subdifferential; Lebourg's theorem [17] was established for locally Lipschitz functions using Clarke's subgradients; and recent results of [11], [20], [21], [26], [32], [35] were given for lower semicontinuous functions by means of generalized subgradients. Although these latter theorems are formulated for general functions, most of their practical applications are restricted to the class of locally Lipschitz functions. A smaller class, which is also very important as shown in [12], is the class of so-called $C^{1,1}$ functions, i.e., differentiable functions with locally Lipschitz derivatives. Using Clarke's generalized Jacobian [5], the authors of [12] introduce the concept of generalized Hessian matrix, prove a second-order Taylor expansion for $C^{1,1}$ functions, then apply it to get second-order optimality conditions for nonlinear constrained problems. Following this tradition, we set our aim to extend Taylor's formula to $C^{k,1}$ functions, i.e., functions whose kth order derivatives are locally Lipschitz, and to apply it to derive high-order optimality conditions and characterizations of quasiconvex functions.

This paper is structured as follows. In the next section we define the $(k+1)$th order subdifferential of a $C^{k,1}$ function and give a chain rule needed in the sequel. In §3, two formulations of Taylor's theorem are established for $C^{k,1}$ functions. In one of the formulations the remainder satisfies the same convergence property as in the classical form for $(k+1)$-time differentiable functions. In §4, we prove an implicit function theorem and present a calculus rule for generalized Hessian of implicit functions of class $C^{1,1}$. Section 5 is devoted to an application of Taylor's formula in

* Received by the editors June 25, 1993; accepted for publication (in revised form) March 4, 1994.
† On leave from the Institute of Mathematics, Hanoi, Vietnam. Département de Mathématiques, Université d'Avignon, 33 rue Louis Pasteur, 84000 Avignon, France (luc@frmop22.cnusc.fr).

optimization problems with $C^{k,1}$ data. Another application is given in §6. Using Taylor's formula and an estimation formula of generalized Hessian, we extend Crouzeix's quasiconvexity criterion (see [6], [2]), one of the deepest results known on second-order characterizations of quasiconvex functions, to $C^{1,1}$ functions.

2. High order subdifferential. Throughout the paper we denote by $C^{k,1}, k \geq 1$, the class of k-time differentiable functions on R^n whose kth order derivatives are locally Lipschitz functions from R^n to R^{n^k}, and by $C^{0,1}$ the class of locally Lipschitz functions on R^n. For every $f \in C^{k,1}$ by Rademacher's theorem, its kth order derivative $D^k f$ is a function differentiable almost everywhere. The generalized Jacobian of $D^k f$ at $x \in R^n$ in Clarke's sense [5], denoted by $\tilde{J}D^k f(x)$, is defined as the convex hull of all $(n^k \times n)$-matrices obtained as the limit of a sequence of the form $JD^k f(x_i)$, where $\{x_i\}_{i=1}^{\infty}$ converges to x and the classical Jacobian matrix $JD^k f(x_i)$ of $D^k f$ at x_i exists.

Let us define the $(k+1)$th order subdifferential of f at x as the set

$$\partial^{k+1} f(x) := \tilde{J}D^k f(x).$$

Elements of this set are called $(k+1)$th order subgradients of f at x. They can be considered as multilinear functions on the space $R^n \times \ldots \times R^n$ ($k+1$ times). The space of $(n^k \times n)$-matrices is endowed with the norm (see [5])

$$\|M\| = \left\{ \sum_{i=1}^{n^k} |a_i|^2 : \ a_i \text{ is the } i\text{th row of } M \right\}^{\frac{1}{2}}.$$

In [12] the second order subdifferential is called generalized Hessian matrix. It is clear that $(k+1)$th order subdifferentials enjoy all the properties of generalized Jacobian. We refer the reader to [5] for properties and calculus rules of the latter (see also [12] for the case $k = 1$.) The following chain rule will be needed.

LEMMA 2.1. *Let x and u be two points of R^n. Let g be a function from R to R^n defined by $g(t) = x + tu$ for every $t \in R$, and f a $C^{k,1}$ function on R^n. Then*

$$\partial^{k+1}(f \circ g)(t) \subseteq \partial^{k+1} f(x + tu)(u, \ldots, u).$$

Proof. Denote by $\Omega_{f \circ g}$ the set of points where the function $f \circ g$ fails to be $(k+1)$-time differentiable. By the definition of the $(k+1)$th order subdifferential,

$$\partial^{k+1}(f \circ g)(t_0) = \operatorname{conv}\{\lim D^{k+1}(f \circ g)(t_i) : t_i \to t_0, t_i \notin \Omega_{f \circ g}\},$$

where $\operatorname{conv}\{\ldots\}$ stands for the convex hull of the set under the parentheses. For each t_i, by the definition of derivative,

$$D^{k+1}(f \circ g)(t_i) = \lim_{t \to 0} \frac{D^k f \circ g(t_i + t) - D^k f \circ g(t_i)}{t}.$$

By the classical chain rule, we have

$$D^k f \circ g(t_i + t) = D^k f(g(t_i + t))(u, \ldots, u),$$
$$D^k f \circ g(t_i) = D^k f(g(t_i))(u, \ldots, u).$$

Applying the mean value theorem [5] for the vector function $D^k f$, one can find a matrix $A_i(t) \in \operatorname{conv}\tilde{J}D^k f([g(t_i), g(t_i + t)])$ such that

$$D^k f(g(t_i + t)) - D^k f(g(t_i)) = A_i(t)(tu).$$

Consequently,

$$D^{k+1}f \circ g(t_i) = \lim_{t\to 0} A_i(t)(u,\dots,u).$$

By the upper continuity of the generalized Jacobian map we conclude that $D^{k+1}f \circ g(t_i) \in \partial^{k+1}f(g(t_i))(u,\dots,u)$, and hence

$$\lim_{t_i\to t_0} D^{k+1}f \circ g(t_i) \in \partial^{k+1}f(g(t_0))(u,\dots,u).$$

The proof is complete. □

We also make use of the following functions of two variables $x, u \in R^n$:

$$D_+^{k+1}f(x;u) := \sup\{A(u,\dots,u) : A \in \partial^{k+1}f(x)\},$$
$$D_-^{k+1}f(x;u) := \inf\{A(u,\dots,u) : A \in \partial^{k+1}f(x)\}.$$

Note that sup and inf in the above expressions can be replaced by max and min, respectively, because the set $\partial^{k+1}f(x)$ is nonempty compact.

LEMMA 2.2. *For every fixed $x \in R^n$, as functions in the variable u, $D_+^{k+1}f$ and $D_-^{k+1}f$ are continuous, positively homogeneous of degree $k+1$ if k is even and homogeneous of degree $k+1$ if k is odd. Furthermore, for every fixed $u \in R^n$, the function $D_+^{k+1}f$ is upper semicontinuous, while the function $D_-^{k+1}f$ is lower semicontinuous in the variable x.*

Proof. The first part of the lemma is evident because every $(k+1)$th order subgradient at a fixed point is a multilinear function. The second part follows from the fact that the $(k+1)$th order subdifferential map is upper continuous compact-valued. □

3. Taylor's formula. Let us consider the case $n = 1$ first.

LEMMA 3.1. *Let φ be a $C^{k,1}$ function from R to R. There exist $t_0 \in (0,1)$ and $\alpha \in \partial^{k+1}\varphi(t_0)$ such that*

$$\varphi(1) - \varphi(0) = \sum_{i=1}^{k} \frac{1}{i!} D^i\varphi(0) + \frac{1}{(k+1)!}\alpha.$$

Proof. Denote by α a real number that satisfies the equality in the lemma and consider the following function:

$$h(t) = \varphi(1) - \varphi(t) - \sum_{i=1}^{k} \frac{1}{i!} D^i\varphi(t)(1-t)^i - \frac{1}{(k+1)!}\alpha(1-t)^{k+1}.$$

This function is locally Lipschitz, therefore we can apply Lebourg's mean value theorem to get a point $t_0 \in (0,1)$ such that $0 \in \partial h(t_0)$. It is evident that

$$\partial h(t_0) = -\frac{1}{k!}\partial^{k+1}\varphi(t_0)(1-t_0)^k + \frac{1}{k!}\alpha(1-t_0)^k.$$

Hence $\alpha \in \partial^{k+1}\varphi(t_0)$, and the proof is complete. □

THEOREM 3.2. *Let f be a $C^{k,1}$ function from R^n to R and let a, b be two arbitrary points in R^n. Then there exist a point $c \in (a,b)$ and a $(k+1)$th order subgradient A_b of f at c such that*

$$f(b) - f(a) = \sum_{i=1}^{k} \frac{1}{i!} D^i f(a)(b-a,\dots,b-a) + \frac{1}{(k+1)!} A_b(b-a,\dots,b-a).$$

Moreover, there exist a neighborhood U of a and a positive K such that $\|A_b\| \leq K$, for all $b \in U$.

Proof. Define a function φ from R to R by $\varphi(t) = f(a + t(b - a))$. It is obvious that φ is of class $C^{k,1}$. Now apply Lemmas 2.1 and 3.1 to get the Taylor formula. The second assertion of the theorem is derived from [5, Prop. 2.6.2]. □

To get a better estimation for the remainder, we give another formulation of Taylor's theorem.

COROLLARY 3.3. *Let f be as in the previous theorem. Then for every $x \in R^n$, there exist a $(k+1)$th order subgradient A_x of f at a and a $(n^k \times n)$-matrix $r(x)$ such that*

$$f(x) = f(a) + \sum_{i=1}^{k} \frac{1}{i!} D^i f(a)(x-a, \ldots, x-a) + \frac{1}{(k+1)!} A_x(x-a, \ldots, x-a) + r(x)(x-a, \ldots, x-a),$$

where $\lim_{x \to a} \|r(x)\| = 0$.

Proof. By Lemma 3.1, for $x \in R^n$, there can be found $c \in (a, x)$ and a $(k+1)$th order subgradient B_x of f at c such that

$$(1)\quad f(x) = f(a) + \sum_{i=1}^{k} \frac{1}{i!} D^i f(a)(x-a, \ldots, x-a) + \frac{1}{(k+1)!} B_x(x-a, \ldots, x-a).$$

Let $A_x \in \partial^{k+1} f(a)$ be a matrix minimizing the distance from B_x to the elements of the convex compact set $\partial^{k+1} f(a)$. Set

$$r(x) = \frac{B_x - A_x}{(k+1)!}.$$

Then (1) gives us the formula of the corollary. Moreover, since the map $\partial^{k+1} f$ is upper continuous, the distance from B_x to $\partial^{k+1} f(a)$ converges to zero as x tends to a, hence the norm of $r(x)$ converges to zero as well. □

4. Implicit functions. The main purpose of this section is to calculate the generalized Hessian of an implicit function that will be needed in applications. Let us first formulate the inverse function theorem and implicit function theorem for $C^{k,1}$ functions.

LEMMA 4.1. *Let g be a $C^{k,1}$ function from R^n to R^n with the property that every matrix of the first order subdifferential of g at $x_0 \in R^n$ is inversible. Then there exists a $C^{k,1}$ inverse function g^{-1} of g on a sufficiently small neighborhood of $g(x_0)$ in R^n.*

Proof. The conclusion of the lemma has been proven in [5] for the case $k = 0$. Thus, there exists a neighborhood U of $g(x_0)$ and the Lipschitz inverse function g^{-1} on U. We must show that this inverse function is of class $C^{k,1}$ if g is (with $k \geq 1$.) By the classical inverse theorem,

$$Dg^{-1}(y) = [Dg(x)]^{-1},$$

where $x = g^{-1}(y), y \in U$. Since Dg is Lipschitz in some neighborhood V of x_0, its inverse $[Dg]^{-1}$ must be also. Let K_1 and K_2 be Lipschitz constants for g^{-1} and

$[Dg]^{-1}$, respectively. Then for $y_1, y_2 \in g^{-1}(V) \cap U$, one has

$$\begin{aligned}\|Dg^{-1}(y_1) - Dg^{-1}(y_2)\| &= \|[Dg(x_1)]^{-1} - [Dg(x_2)]^{-1}\| \\ &\leq K_2\|x_1 - x_2\| \\ &\leq K_2\|g^{-1}(y_1) - g^{-1}(y_2)\| \\ &\leq K_2K_1\|y_1 - y_2\|,\end{aligned}$$

where $x_i = g^{-1}(y_i), i = 1, 2$. This means that Dg^{-1} is Lipschitz around x_0.

For $k = 2$, using the classical calculus rule one has

$$D^2g^{-1}(y) = -Dg^{-1}(y)D^2g(x)[Dg(x)]^{-2}.$$

The Lipschitz property of $D^2g, [Dg]^{-1}$, and Dg^{-1} implies that of D^2g^{-1}. Continuing this process for other k, we conclude that g^{-1} is of class $C^{k,1}$ if g is also. □

Recall [5] that if f is a function from $R^n \times R^m$ to R^m, by $\pi_z\partial f(y, z)$ we denote the set of all $(m \times m)$-matrices M such that for some $(m \times n)$-matrix N, the $(m \times (m+n))$-matrix $[N, M]$ belongs to $\partial f(y, z)$.

LEMMA 4.2. *Let $f(y, z)$ be a $C^{k,1}$ function from $R^n \times R^m$ to R^m with the property that $f(y_0, z_0) = 0$ and every matrix of $\pi_z\partial f(y_0, z_0)$ is inversible. Then there exists a $C^{k,1}$ function g from a sufficiently small neighborhood U of y_0 in R^n to R^m such that $g(y_0) = z_0$ and $f(y, g(y)) = 0$ for all $y \in U$.*

Proof. Invoke to the preceding lemma and to the implicit function theorem (for the case $k = 0$) of [5]. □

Now let us calculate the first and the second order subdifferentials of the function g obtained in Lemma 4.2 for the case $k = 0$ and $k = 1$, respectively. For the sake of simple presentation, assume from now on that $m = 1$. We shall make use of the following partition for every $((n + 1) \times (n + 1))$-matrix H:

$$H = \begin{pmatrix} H_{yy} & H_{yz} \\ H_{zy} & H_{zz} \end{pmatrix},$$

where the dimensions of the submatrices $H_{yy}, H_{yz}, H_{zy}, H_{zz}$ are $n \times n$, $n \times 1$, $1 \times n$, 1×1, respectively.

PROPOSITION 4.3. *Under the hypothesis of Lemma 4.2 we have the following formulas for $k = 0$ and $k = 1$, respectively:*

$$\begin{aligned}\partial g(y_0) \subseteq\ & -[\partial_z f(y_0, z_0)]^{-1}\partial_y f(y_0, z_0); \\ \partial^2 g(y_0) \subseteq\ & -[D_z f(y_0, z_0)]^{-1}\{H_{yy} + [Dg(y_0)]^T H_{zy} \\ & + H_{yz}Dg(y_0) + H_{zz}[Dg(y_0)]^T Dg(y_0) : \\ & H \in \partial^2 f(y_0, g(y_0))\},\end{aligned}$$

where $(\ldots)^T$ denotes the transposition of the matrix under the parentheses.

Proof. Let us prove the second inclusion. The first one can be done by a similar argument. It follows from the classical implicit function theorem that

$$D_y f(y, g(y)) + D_z f(y, g(y))Dg(y) = 0,$$

for every y in a small neighborhood U of y_0. We consider first the case where g is twice differentiable at some point $\bar{y}$ near to y_0. One has

$$\begin{aligned}&[D_y f(y, g(y)) - D_y f(\bar{y}, g(\bar{y}))] \\ &+[D_z f(y, g(y))Dg(y) - D_z f(\bar{y}, g(\bar{y}))Dg(\bar{y})] = 0\end{aligned} \tag{2}$$

for every $y \in U$. To evaluate the differences in the left-hand side of (2), let us apply the mean value theorem [5] to the vector function $Df = (D_y f, D_z f)$ from $R^n \times R^1$ to itself. Thus, there exists a $((n+1) \times (n+1))$-matrix $H(y) \in \operatorname{conv} \partial Df([(\bar{y}, g(\bar{y})), (y, g(y))])$ such that

$$Df(y, g(y)) - Df(\bar{y}, g(\bar{y})) = H(y)(y - \bar{y}, g(y) - g(\bar{y})).$$

In terms of the partition of the matrix $H(y)$ the latter equality reads as

$$D_y f(y, g(y)) - D_y f(\bar{y}, g(\bar{y})) = H_{yy}(y)(y - \bar{y}) + H_{yz}(y)(g(y) - g(\bar{y})), \tag{3}$$

$$D_z f(y, g(y)) - D_z f(\bar{y}, g(\bar{y})) = H_{zy}(y)(y - \bar{y}) + H_{zz}(y)(g(y) - g(\bar{y})). \tag{4}$$

Again, applying the mean value theorem for the function g, we can find an $(1 \times n)$-matrix $B(y) \in \operatorname{conv} Dg([\bar{y}, y])$ such that

$$g(y) - g(\bar{y}) = B(y)(y - \bar{y}).$$

With this, the equalities (3) and (4) become

$$D_y f(y, g(y)) - D_y f(\bar{y}, g(\bar{y})) = [H_{yy}(y) + H_{yz}(y)\ B(y)](y - \bar{y}), \tag{5}$$

$$D_z f(y, g(y)) - D_z f(\bar{y}, g(\bar{y})) = [H_{zy}(y) + H_{zz}(y)\ B(y)](y - \bar{y}). \tag{6}$$

Furthermore, since g is twice differentiable at $\bar{y}$, using Taylor's formula one has

$$Dg(y) - Dg(\bar{y}) = D^2 g(\bar{y})(y - \bar{y}) + r(y)(y - \bar{y}), \tag{7}$$

where $\lim_{y \to \bar{y}} \|r(y)\| = 0$. By (6) and (7) the second difference in the left-hand side of (2) can be evaluated as

$$\begin{aligned} & D_z f(y, g(y)) Dg(y) - D_z f(\bar{y}, g(\bar{y})) Dg(\bar{y}) \\ &= [D_z f(y, g(y)) - D_z f(\bar{y}, g(\bar{y}))] Dg(y) + D_z f(\bar{y}, g(\bar{y}))[Dg(y) - Dg(\bar{y})] \\ &= \{[Dg(y)]^T (H_{zy}(y) + H_{zz}(y)\ B(y)) \\ &\quad + D_z f(\bar{y}, g(\bar{y}))(D^2 g(\bar{y}) + r(y))\}(y - \bar{y}). \end{aligned} \tag{8}$$

Combining (5) and (8) with (2), we finally obtain

$$\begin{aligned} & [H_{yy}(y) + H_{yz}(y)\ B(y) + [Dg(\bar{y})]^T (H_{zy}(y) + H_{zz}(y)\ B(y)) \\ & + D_z f(\bar{y}, g(\bar{y})) D^2 g(\bar{y})](y - \bar{y}) + D_z f(\bar{y}, g(\bar{y})) r(y)(y - \bar{y}) = 0 \end{aligned} \tag{9}$$

for all $y \in U$. By letting y converge to $\bar{y}$ and using the upper continuity of generalized Jacobian maps, we have

$$\lim_{y \to \bar{y}} Dg(y) = Dg(\bar{y}),$$

$$\lim_{y \to \bar{y}} B(y) \subseteq \partial g(\bar{y}) = Dg(\bar{y}),$$

$$\limsup_{y \to \bar{y}} H(y) \subseteq \partial^2 f(\bar{y}, g(\bar{y})),$$

where lim sup denotes the upper limit in the Kuratowski–Painleve sense [1]. Consequently, (9) yields

$$\begin{aligned} D^2 g(\bar{y}) \in -[D_z f(\bar{y}, \bar{z})]^{-1} \{ & H_{yy}(\bar{y}) + [Dg(\bar{y})]^T H_{zy}(\bar{y}) \\ & + H_{yz}(\bar{y}) Dg(\bar{y}) + H_{zz}(\bar{y})[Dg(\bar{y})]^T Dg(\bar{y}) : \\ & \text{where} \begin{pmatrix} H_{yy}(\bar{y}) & H_{yz}(\bar{y}) \\ H_{zy}(\bar{y}) & H_{zz}(\bar{y}) \end{pmatrix} \in \partial^2 f(\bar{y}, g(\bar{y})) \} \end{aligned} \tag{10}$$

and the formula of the proposition is established for $\bar{y}$.

Now, for the point y_0, by the definition

$$\partial^2 g(y_0) = \operatorname{conv}\{\lim D^2 g(\bar{y}) : \bar{y} \to y_0,\ g \text{ is twice differentiable at } \bar{y}\}.$$

Again, by the upper continuity of generalized Jacobian maps and using (10), we obtain

$$\begin{aligned}\partial^2 g(y_0) \subseteq -\operatorname{conv}\{&[D_z f(y_0, z_0)]^{-1}\{H_{yy} + [Dg(y_0)]^T H_{zy} \\ &+ H_{yz} Dg(y_0) + H_{zz}[Dg(y_0)]^T Dg(y_0) : \\ &H \in \partial^2 f(y_0, g(y_0))\}\}.\end{aligned}$$

In the latter formula conv is superfluous because the set under the parentheses is already convex. □

It should be noted that for the case $k = 0$ using a result of [16] one can express the subdifferential of g in terms of the subdifferential of the inverse function ϕ of $f(y_0, .)$ and the one of $f(., z_0)$. Namely, let us recall that the set-valued directional derivative of f at x in direction h, denoted by $\Delta f(x; h)$, is the set consisting of all limits

$$v = \lim \frac{f(x_k + \lambda_k h) - f(x_k)}{\lambda_k},$$

where $x_k \to x$ and $\lambda_k \downarrow 0$ and that $\partial f(x)h = \operatorname{conv}\Delta f(x; h)$. It was shown in [16] that

$$\Delta g(y_0; h) = \Delta\phi(z_0; -\Delta_y f((y_0, z_0); h))$$

for all $h \in R^n$ under the conditions: (i) $0 \notin \Delta f((y_0, z_0); (0, \tau))$ for all $\tau \in R \setminus \{0\}$; (ii) $\Delta f((y_0, z_0); (u, \tau)) = \Delta_y f((y_0, z_0); u) + \Delta_z f((y_0, z_0); \tau)$. The above result implies $\partial g(y_0) = -\partial\phi(z_0)\partial_y f(y_0, z_0)$. This in its turn will give the first inclusion of Lemma 4.1 if it additionally happens that $\partial\phi(z_0) \subseteq [\partial_z f(y_0, z_0)]^{-1}$.

5. Optimality conditions. A function f of class $C^{k,1}$ on R^n is given. We shall use Taylor's formula to derive optimality conditions via $(k+1)$th order subdifferential of f.

PROPOSITION 5.1. *Let $x_0 \in R^n$ be a local minimum of f with the property that $D^i f(x_0) = 0$ for $i = 1, \ldots, k$. Then $D_+^{k+1} f(x_0)(u) \geq 0$ for all $u \in R^n$. In particular, if k is even, then $0 \in \partial^{k+1} f(x_0)(u, \ldots, u)$ for all $u \in R^n$.*

Proof. Using Lemma 3.1, we have

$$\begin{aligned}f(x_0 + tu) - f(x_0) \in &\frac{1}{(k+1)!}\partial^{k+1} f(x_0)(tu, \ldots, tu) \\ &+ r(x_0 + tu)(tu, \ldots, tu).\end{aligned} \tag{11}$$

Observe that when $t \geq 0$ is close to 0 the difference in the left-hand side of the above inclusion is nonnegative. Moreover, since $r(x_0 + tu)$ converges to 0 as t tends to 0, we conclude that $D_+^{k+1} f(x_0)(u) \geq 0$. If k is even, by taking $t < 0$ in (11) we see that $D_-^{k+1} f(x_0)(u) \leq 0$. Hence $0 \in \partial^{k+1} f(x_0)(u, \ldots, u)$, for all $u \in R^n$. □

PROPOSITION 5.2. *Let $x_0 \in R^n$ be a point with the property that $D^i f(x_0) = 0$ for $i = 1, \ldots, k$, and $D_-^{k+1} f(x_0)(u) > 0$ for all $u \in R^n, u \neq 0$. Then x_0 is a local strict minimum of f.*

Proof. We have the formula (11) as in the preceding proof. If x_0 is not a local strict minimum, there exists a sequence $\{x_i\}_{i=1}^{\infty} \subseteq R^n$ distinct from and converging to x_0 such that $f(x_i) - f(x_0) \leq 0$. Let $A_i \in \partial^{k+1} f(x_0)$ be such that

$$f(x_i) - f(x_0) = \frac{1}{(k+1)!} A_i(x_i - x_0, \ldots, x_i - x_0) + r(x_i)(x_i - x_0, \ldots, x_i - x_0).$$

Without loss of generality we may assume that A_i converges to some $A \in \partial^{k+1} f(x_0)$ and $(x_i - x_0)/\|x_i - x_0\|$ converges to some vector $u \in R^n, u \neq 0$. Then $A(u, \ldots, u) \leq 0$, which contradicts the assumption of the proposition. □

The results of the previous propositions have been proven for the case $k = 1$ in [12], [14]. Similar optimality conditions expressed in terms of high order directional derivatives have been given in [33]. The difference between the conditions given in this section and those of [33] (Corollaries 2.1, and 2.2 for the case $C = R^n$) results from the choice of using Clarke's generalized derivatives in our study and the contingent derivatives in [33].

It should be noted that under the hypothesis of Proposition 5.2, the integer k must be odd, because the function $D_-^{k+1} f$ is homogeneous of degree $k+1$ in the case where k is even by Lemma 2.2.

6. Generalized Hessian of quasiconvex functions. We recall that a function f from R^n to R is said to be convex (respectively, quasiconvex) if for every $x, y \in R^n$ and for every $\lambda \in (0,1)$ one has $f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y)$ (respectively, $f(\lambda x + (1-\lambda)y) \leq \max\{f(x), f(y)\}$).

There exists an extensive number of papers and books on these functions (see [2], [3], [4], [8], [13], [15], [24]–[25], [27], [28], [30], [31]) Most characterizations of convex and quasiconvex functions are expressed in terms of first and second order derivatives, and recently, in terms of generalized subgradients (see [9], [18], [19], [21], [29]). By our knowledge, [12] was the first paper where generalized Hessian was used to give the second order criterion for convex functions. The authors of [12] pointed out that a $C^{1,1}$ function is convex if and only if its second order subgradients at any point are positive semidefinite (see also [29]). In this section, we use generalized Hessian to give quasiconvexity conditions for $C^{1,1}$ functions.

PROPOSITION 6.1. *Assume that f is a quasiconvex $C^{1,1}$ function. Then for every $x, u \in R^n$, $Df(x)(u) = 0$ implies $D_+^2 f(x)(u) \geq 0$.*

Proof. Suppose to the contrary that $D_+^2 f(x)(u) < 0$ for some $x, u \in R^n$ with $Df(x)(u) = 0$. Since $D_+^2 f$ is upper semicontinuous (in view of Lemma 2.2), there exist two positives t_0 and ϵ such that $D_+^2 f(x+tu)(u) < -\epsilon$ for every $t \in [-t_0, t_0]$. By Theorem 3.2, one can find $A_t \in \partial^2 f([x, x+tu])$ such that

$$f(x+tu) - f(x) = t^2 A_t(u, u).$$

This implies that $f(x+tu) - f(x) < -\epsilon t^2$, for all $t \in [-t_0, t_0]$. In particular, $f(x+t_0 u) < f(x)$ and $f(x - t_0 u) < f(x)$. This contradicts the quasiconvexity of f. □

It is worthwhile to notice that the conclusion of the previous proposition is not valid for $D_-^2 f(x)(u)$. For instance the function defined on R by

$$f(x) = \begin{cases} x^2 \ if \ x \leq 0, \\ -x^2 \ \text{otherwise}, \end{cases}$$

is a $C^{1,1}$ quasiconvex function. At $x = 0$, the first derivative is zero and the second subdifferential is $[-1, 1]$, with $D_-^2 f(0)(1) = -1$. Moreover, the condition given in the

proposition is necessary for quasiconvextity, but generally it is not sufficient, even in the case of C^2 functions (see [2]). Among existing second order sufficient conditions, Crouzeix's result [6] is probably the strongest. Let us now extend it to $C^{1,1}$ functions.

PROPOSITION 6.2. *Assume that a $C^{1,1}$ function f satisfies the following condition: for every $x, u \in R^n, u \neq 0$,*

$Df(x)(u) = 0$ implies $D^2_- f(x)(u) \geq 0$,

$Df(x) = 0$ implies $D^2_- f(x)(u) > 0$.

Then f is quasiconvex.

Proof. Let us follow the method of [6] by using Theorem 3.2 and Proposition 4.3 instead of the corresponding classical theorems. Suppose that f is not quasiconvex, i.e., there exist $x, u \in R^n, u \neq 0$, such that

$$\max_{t\in[0,1]} f(x+tu) > \max\{f(x), f(x+u)\}.$$

Let $c = x + t_0 u$ be the maximum point that is the closest to x. It is evident that $Df(c)(u) = 0$. There are two possible cases: (i) $Df(c) = 0$; and (ii) $Df(c) \neq 0$. In case (i), applying Theorem 3.2, we have

$$f(x+tu) - f(c) = (t - t_0)^2 A_t(u, u),$$

for some $A_t \in \partial^2 f([c, x+tu])$. Since $D^2_- f(c)(u) > 0$ in view of Lemma 2.2, we see that $A_t(u, u) > 0$ whenever t is close to t_0. Consequently, $f(x+tu) > f(c)$ a contradiction. Let us treat case (ii). Without loss of generality we may assume that $c = 0, f(c) = 0$, and $Df(c) = (0, \ldots, 0, 1)$. We shall write (y, z) to indicate a vector of the product space $R^{n-1} \times R$ and $c = (y_0, z_0)$. With this notation, $f(y_0, z_0) = 0$ and $D_z f(y_0, z_0) = 1$. By Lemma 4.2, there exists a small ball $U \subseteq R^{n-1}$ with center at the origin and a $C^{1,1}$ function g on U such that $f(y, g(y)) = 0$ for all $y \in U$.

We want to show that g is concave on U. Indeed, for every $v \in R^{n-1}$, by the implicit function theorem, the vector $(v, vDg(y))$ satisfies the relation

$$Df(y, g(y))(v, vDg(y)) = 0.$$

Therefore, by the hypothesis of the proposition, one has

$$H((v, vDg(y)), (v, vDg(y))) \geq 0$$

for all $H \in \partial^2 f(y, g(y))$. This and Proposition 4.3 show that $A(v, v) \leq 0$ for every $A \in \partial^2 g(y), y$ sufficiently close to y_0, and for every $v \in R^{n-1}$. Thus g is a concave function on some convex neighborhood $U_0 \subseteq U$ of y_0.

Remember that $Df(c)(u) = 0$. Let us write u in the new coordinates: $u = (v_0, w_0) \in R^{n-1} \times R$. Since $Df(c) = (0, \ldots, 0, 1)$, the component w_0 must be zero. Hence, $f(tv_0, 0) < 0$ for all $t \in [-t_0, 0)$, which implies that $g(tv_0) \neq 0$ for every $t \in [-t_0, 0)$ with $tv_0 \in U_0$. By the continuity of Df there exist a neighborhood $V \subseteq U_0$ and a positive δ such that $D_z f(y, z) > 0$ for all $(y, z) \in V \times [-\delta, \delta]$. Furthermore, by the implicit function theorem, $Dg(y_0) = [D_z f(y_0, z_0)]^{-1} D_y f(y_0, z_0) = 0$. This fact and the concavity of g imply that $g(tv_0) \leq 0$ for every $t \in [-t_0, 0]$ with $tv_0 \in U_0$. Thus, there can be found $t_1 \in [-t_0, 0)$ such that $t_1 v_0 \in U_0$ and $-\delta < g(t_1 v_0) < 0$. We have finally,

$$f(t_1 v_0, 0) < 0, f(t_1 v_0, g(t_1 v_0)) = 0, \tag{12}$$

and $D_z f(t_1 v_0, z) > 0$ for all $z \in [g(t_1 v_0), 0]$. The latter inequality shows that the function $f(t_1 v_0, .)$ is increasing. In particular, $f(t_1 v_0, g(t_1 v_0)) < f(t_1 v_0, 0)$. This contradicts (12) and completes the proof. □

Remark. Under the assumptions of Proposition 6.2 the function f is pseudoconvex in the sense that for all $x, y \in R^n, f(y) < f(x)$ implies $Df(x)(y - x) \leq 0$. To see this it suffices to apply the above proposition, Theorem 2.2 of [7], and to observe if $Df(x) = 0, f$ has a local minimum at x according to Proposition 5.2.

Acknowledgments. This paper was written during the author's stay at the Universitat Autonoma de Barcelona. The author thanks Professors I.Fradera and J.E.Martinez-Legaz for their kind invitation, and the referees for several useful remarks on the first version of the paper, especially for pointing out references [7], [14], [16], and [33].

REFERENCES

[1] AUBIN, H. FRANKOWSKA, *Set-Valued Analysis*, Birkhauser, Basel, 1990.
[2] M. AVRIEL, W. E. DIEWERT, S. SCHAIBLE, AND I. ZANG, *Generalized Concavity*, Plenum Press, New York, London, 1988.
[3] K. J. ARROW AND A. C. ENTHOVEN, *Quasiconcave programming*, Econometrica, 29(1961), pp. 779–800.
[4] A. CAMBINI, E. CASTAGNOLI, L. MARTEIN, P. MAZZOLENI, AND S. SCHAIBLE, EDS., *Generalized Convexity and Fractional Programming with Economic Applications*, Lecture Notes in Economics and Mathematical Systems 345, Springer-Verlag, Berlin, Heidelberg, 1990.
[5] F. H. CLARKE, *Optimization and Nonsmooth Analysis*, Wiley, New York, 1983.
[6] J. P. CROUZEIX, *On second order conditions for quasiconvexity*, Math. Programming, 18(1980), pp. 349–352.
[7] J. P. CROUZEIX AND J. A. FERLAND, *Criteria for quasi-convexity and pseudo-convexity: relationships and comparisons*, Math. Programming, 23(1982), pp. 193–205.
[8] W. E. DIEWERT, M. AVRIEL, AND I. ZANG, *Nine kinds of quasiconcavity and concavity*, J. Economic Theory, 25 (1981), pp. 397–420.
[9] R. ELLAIA AND H. HASSOUNI, *Characterizations of nonsmooth functions through their generalized gradients*, Optimization, 22(1991), pp. 401–416.
[10] J.-B. HIRIART-URRUTY, *Miscellanies on nonsmooth analysis and optimization*, Lecture Notes in Economics and Mathematical Systems, 255(1986), pp. 8–24.
[11] ———, *Mean value theorems in nonsmooth analysis*, Numer. Funct. Anal. Optim., 2(1980), pp. 1–30.
[12] J.-B. HIRIART-URRUTY, J.-J. STRODIOT, AND V. HIEN NGUYEN, *Generalized Hessian matrix and second order optimality conditions for problems with $C^{1,1}$ data*, Appl. Math. Optim., 11(1984), pp. 43–56.
[13] S. KARAMARDIAN, *Duality in mathematical programming*, J. Math. Anal. Appl., 20(1967), pp. 344–358.
[14] D. KLATTE AND K. TAMMER, *On second order sufficient optimality conditions for $C^{1,1}$ optimization problems*, Optimization 19(1988), pp. 169–180.
[15] S. KOMLOSI, *Some properties on nondifferentiable pseudoconvex functions*, Math. Programming, 26(1983), pp. 232–237.
[16] B. KUMMER, *An implicit function theorem for $C^{0,1}$ equations and parametric $C^{1,1}$ optimization*, J. of Math. Anal. Appl., 158(1991), pp. 35–46.
[17] G. LEBOURG, *Valeur moyenne pour gradient généralisé*, Comptes Rendus Acad. Csi. Paris, A (1975), pp. 795–797.
[18] D. T. LUC, *On the maximal monotonicity of subdifferentials*, Acta Math. Vietnam., 18 (1993), pp. 99–106.
[19] ———, *Characterizations of quasiconvex functions*, Bull. Austral. Math. Soc., 48(1993), pp. 193–405.
[20] ———, *A strong mean value theorem and applications*, Nonlinear Anal. Appl., to appear.
[21] D. T. LUC AND S. SWAMINATHAN, *A characterization of convex functions*, Nonlinear Anal. Appl., 20(1993), pp. 697–701.
[22] O. L. MANGASARIAN, *Nonlinear Programming*, McGraw-Hill Book Co., New York, 1969.

[23] J.-E. MARTINEZ-LEGAZ, *Weak lower subdifferentials and applications*, Optimization, 21(1990), pp. 321–341.
[24] P. MAZZOLENI, ED., *Generalized Concavity for Economic Applications*, Pisa University, Pisa, 1992.
[25] J. M. ORTEGA AND W. C. RHEINBOLDT, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, London, 1970.
[26] J. P. PENOT, *On the mean value theorem*, Optimization, 19(1988), pp. 147–156.
[27] J. PONSTEIN, *Seven kinds of convexity*, SIAM Rev., 9 (1967), pp. 115–119.
[28] R. T. ROCKAFELLAR, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.
[29] A. SEEGER, *Analyse du Second Ordre de Problemes Non Differentiables*, These, Université Paul Sabatier de Toulouse, 1986.
[30] S. SCHAIBLE, *Generalized convexity of quadratic functions*, in Generalized concavity in optimization and economics, S. Schaible and W. T. Ziemba, eds., Academic Press, New York 1981, pp. 183–197.
[31] S. SCHAIBLE AND W. T. ZIEMBA, *Generalized Concavity in Optimization and Economics*, Academic Press, New York, 1981.
[32] M. STUDNIARSKI, *Mean value theorems and sufficient optimality conditions for nonsmooth functions*, J. Math. Anal. and Appl., 111 (1985), pp. 313–326.
[33] M. STUDNIARSKI, *Necessary and sufficient conditions for isolated local minima of nonsmooth functions*, SIAM J. Control Optim., 24(1986), pp. 1044–1049.
[34] L. WEGGE, *Mean value theorems for convex functions*, J. Math. Econom., 1(1974), pp. 207–208.
[35] D. ZAGRODNY, *Approximate mean value theorem for upper subderivatives*, Nonlinear Anal. Appl., 12(1988), pp. 1413–1428.

SIAM J. OPTIMIZATION
Vol. 5, No. 3, pp. 670–680, August 1995

010

THE LINEAR NONCONVEX GENERALIZED GRADIENT AND LAGRANGE MULTIPLIERS*

JAY S. TREIMAN†

Abstract. A Lagrange multiplier rule that uses small generalized gradients is introduced. It includes both inequality and set constraints. The generalized gradient is the linear generalized gradient. It is smaller than the generalized gradients of Clarke and Mordukhovich but retains much of their nice calculus. Its convex hull is the generalized gradient of Michel and Penot if a function is Lipschitz.

The tools used in the proof of this Lagrange multiplier result are a coderivative, a chain rule, and a scalarization formula for this coderivative. Many smooth and nonsmooth Lagrange multiplier results are corollaries of this result.

It is shown that the technique in this paper can be used for cases of equality, inequality, and set constraints if one considers the generalized gradient of Mordukhovich. An open question is: Does a Lagrange multiplier result hold when one has equality constraints and uses the linear generalized gradient?

Key words. nonsmooth analysis, generalized gradient, co-derivative, Lagrange multipliers

AMS subject classifications. 49J52, 90C30

1. Introduction. Following the work of Mordukhovich [8], [10]–[12] and Ioffe [4], [5], a small nonconvex generalized gradient was defined in [16]. In this paper we consider some properties of the coderivative for the linear generalized gradient and show how these can be used to prove necessary conditions for optimality. More precisely, a Lagrange multiplier theorem is proven that includes inequality and set constraints.

The linear generalized gradient is a smaller version of the generalized gradient of Mordukhovich that retains many of its properties. It has good rules for the generalized gradient of the sum of functions and the maximum of functions. For our purposes it is also important that there is a good chain rule involving a coderivative and a scalarization formula for the coderivative.

Given this nice calculus, we show one can prove a Lagrange multiplier rule for Lipschitz problems. This result includes basic results for convex functions, the Clarke and Mordukhovich generalized gradients, the generalized gradient of Michel and Penot, and Fréchet differentiable functions.

This paper is divided into five sections. Section 1 gives the basic definitions for the linear generalized gradient. Section 2 continues with a review of some of the calculus for the linear generalized gradient, and concludes with a new result that will be used in proving the Lagrange multiplier result. Section 3 contains a proof of the Lagrange multiplier result and several corollaries. Section 4 gives several examples that help place this result, and in §5 the question of including equality constraints is discussed.

2. Definitions and calculus. The basic objects used to define both the linear generalized gradient and the generalized gradient of Mordukhovich are the proximal normal and proximal subgradients. For our purposes the definition is restricted to $\mathbb{R}^n$; however, a similar definition can be used in Banach spaces with a smooth renorm.

DEFINITION 2.1. *Let $C \subset \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$. A $v \in \mathbb{R}^n$ is a* proximal normal *to C at $x \in C$, if, for some $\lambda > 0$*

$$C \cap \bar{B}(x + \lambda v, \lambda \|v\|) = \{x\}.$$

Here $\bar{B}(y, \rho)$ is the closed ball centered at y with radius ρ.

* Received by the editors May 5, 1993; accepted for publication (in revised form) February 21, 1994.
† Department of Mathematics and Statistics, Western Michigan University, Kalamazoo, Michigan 49008 (treiman@math-stat.wmich.edu).

A $w \in \mathbb{R}^n$ *is a* proximal subgradient *to* f *at* x *if*

$$f(y) \geq f(x) + \langle w, y - x\rangle - \mu\|y - x\|^2$$

for all y *in some neighborhood of* x.

These definitions have been used to characterize the generalized gradients and normal cones of Clarke [1]–[3] and Mordukhovich [5], [8], [10], [11], the B-gradient [13]–[15], and to define the linear normal cone and linear generalized gradient [16]. They are also used, through normal cone definitions, to define the coderivative of Mordukhovich [11], [12] and the linear coderivative [16].

An element of the normal cone of Mordukhovich is defined as the limit of a sequence of proximal normals. To define the linear normal cone, a restriction in the convergence of the proximal normals is used.

DEFINITION 2.2. *A sequence of proximal normals* $v^k \to v$ *to a closed set* $C \subset \mathbb{R}^n$ *at* $x^k \to \bar{x}$ *is* linear *if, either* $x^k \neq \bar{x}$ *for all* k, *and for some* $\lambda > 0$ *and all sufficiently large* k,

$$C \cap \bar{B}(x^k + \lambda\|x^k - \bar{x}\|v^k, \lambda\|x^k - \bar{x}\|\|v^k\|) = \{x^k\},$$

or $x^k = \bar{x}$ *for all* k.

With this definition one can define the linear normal cone.

DEFINITION 2.3. *Let* C *be a closed subset of* $\mathbb{R}^n$. *The* linear normal cone *to* C *at* $\bar{x}$ *is*

$$N_\ell(C, \bar{x}) := \text{cl}\,\{v\colon v \text{ is the limit of a linear sequence of proximal normals } v^k \text{ to } C \text{ at } x^k \to \bar{x}\}.$$

To obtain the normal cone of Mordukhovich one simply removes the linear convergence restriction from the sequences of proximal normals. After this change, the closure is not necessary.

The following example shows that the two cones may be different.

Example 2.4. Let $C = \{(x, y) : y \leq \sqrt{|x|}\}$. Then

$$N_\ell(C, (0,0)) = \{(0,0)\},$$

but the normal cone of Mordukhovich (MNC) is

$$N_M(C, (0,0)) = \{(x, 0) : x \in \mathbb{R}\}.$$

Several properties of $N_\ell(C, x)$ are given in [13]. The linear normal cone (LNC) is easily placed with respect to well-known normal cones.

THEOREM 2.5. *Let* C *be a closed subset of* $\mathbb{R}^n$ *and* $x \in C$. *Then*

$$N_\ell(C, x) \subset N_M(C, x) \subset N_C(C, x),$$

where $N_C(C, x)$ *is the Clarke normal cone* (CNC) *to* C *at* x.

If C *is convex then all three of these cones coincide with the normal cone of convex analysis.*

Proof. The first inclusion follows directly from the definitions. The second inclusion follows from the fact that $N_C(C, x) = \text{cl co}\, N_M(C, x)$.

The proof that all three coincide when the set is convex is simple and left to the reader. □

In optimization problems, one considers problems that involve functional constraints. Thus a generalization of the gradient is required. To define this generalization of the gradient,

restricted sequences of proximal subgradients are used. In what follows $x^k \to_f x$ means that $x^k \to x$ and $f(x^k) \to f(x)$.

DEFINITION 2.6. *A sequence of proximal subgradients $v^k \to v$ to f for $\bar{x}$ is* linear *if either there are $x^k \to_f \bar{x}$, $x^k \neq \bar{x}$, and μ, $\delta > 0$ such that*

$$f(x^k + h) \geq f(x^k) + \langle v^k, h\rangle - \frac{\mu}{\|x^k - \bar{x}\|_f}\|h\|^2$$

on $B(x^k, \delta\|x^k - \bar{x}\|_f)$ with $\|x - y\|_f = \|x - y\| + |f(x) - f(y)|$, or v^k is a proximal subgradient to f at $\bar{x}$ for all k.

As with the linear normal cone, the definition of the linear generalized gradient uses this linear convergence.

DEFINITION 2.7. *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$. The* linear generalized gradient (LGG) *to f at $\bar{x}$ is the set*

$$\partial_\ell f(\bar{x}) := \text{cl}\,\{v : v \text{ is the limit of a linear sequence of proximal subgradients to } f \text{ for } \bar{x}\}.$$

To define the generalized gradient of Mordukhovich (MGG) one removes the linearity restriction on the sequences of proximal subgradients. We will denote the MGG by $\partial_M f(x)$. If a function is Lipschitz, the Clarke generalized gradient (CGG) is the closed convex hull of MGG.

In finite dimensions, the closed convex hull of LGG is the generalized gradient of Michel and Penot (GGP) [9] when f is Lipschitz. It has a very nice calculus and is convex. This has advantages and disadvantages. A corollary comparing the best Lagrange multiplier result for this generalized gradient with the main result in this paper follows Theorem 3.1.

The basic optimality condition holds for the LGG. We will need this in the proof of Theorem 3.1.

PROPOSITION 2.8. *If $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is lower semicontinuous* (lsc) *and $\bar{x}$ is a local minimizer of f, then*

$$0 \in \partial_\ell f(\bar{x}).$$

Proof. Simply note that 0 is a proximal subgradient to f at $\bar{x}$. Thus $0 \in \partial_\ell f(\bar{x})$. □

As one hopes, there is a close relationship between the LGG and the LNC. Here $\delta_C(x)$ is the indicator function of C,

$$\delta_C(x) = \begin{cases} 0 & \text{if } x \in C, \\ +\infty & \text{otherwise.} \end{cases}$$

THEOREM 2.9 [16]. *Let C be a closed subset of $\mathbb{R}^n$. Then*

$$\partial_\ell \delta_C(x) = N_\ell(C, x).$$

Even with the restriction to "linear" convergence, the calculus for this generalized gradient is fairly strong. It extends to include lsc functions.

THEOREM 2.10 [16]. *Let f be an* lsc *function from $\mathbb{R}^n$ to $\mathbb{R} \cup \{+\infty\}$ and let g be a Lipschitz function from $\mathbb{R}^n$ to $\mathbb{R}$. Then*

$$\partial_\ell(f + g)(x) \subset \partial_\ell f(x) + \partial_\ell g(x).$$

There is a rule for the nonnegative multiple of a function.

THEOREM 2.11 [16]. *Let f be an* lsc *function from $\mathbb{R}^n$ to $\mathbb{R} \cup \{+\infty\}$ and let $\alpha \geq 0$. Then*

$$\partial_\ell(\alpha f)(x) = \alpha \partial_\ell f(x).$$

Another important result for our purposes is the following chain rule. It involves the linear coderivative.

DEFINITION 2.12 [16]. *Let $G : \mathbb{R}^m \to \mathbb{R}^n$ be a Lipschitz function. The* linear coderivative *of G is the multifunction $D^*_\ell G$ from $\mathbb{R}^m \times \mathbb{R}^n$ to $\mathbb{R}^m$, where*

$$D^*_\ell G(x)(y^*) = \{v^* : (v^*, -y^*) \in N_\ell(\operatorname{gph} G, (\bar{x}, G(\bar{x})))\}.$$

The coderivative of Mordukhovich is given by a similar formula. The *coderivative* of G is the multifunction $D^*_M G$ from $\mathbb{R}^m \times \mathbb{R}^n$ to $\mathbb{R}^m$ where

$$D^*_M G(x)(y^*) = \{v^* : (v^*, -y^*) \in N_M(\operatorname{gph} G, (\bar{x}, G(\bar{x})))\}.$$

There are good chain rules for both of these coderivatives.

THEOREM 2.13 [16]. *Let $G : \mathbb{R}^m \to \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}$ be Lipschitz functions. Then*

$$\partial_\ell f \circ G(x) \subset D^*_\ell G(x) \partial_\ell f(G(x)).$$

The same result holds if the LGG and coderivative are replaced by those of Mordukhovich [10]–[12]. The other result that is central to the proof of the main theorem of this paper is a "scalarization" formula for the coderivative.

PROPOSITION 2.14. *Let $F : \mathbb{R}^m \to \mathbb{R}^n$ be a Lipschitz function. Then*

$$D^*_\ell F(\bar{x})(\bar{y}) = \partial_\ell \langle \bar{y}, F \rangle (\bar{x})$$

for all $y \in \mathbb{R}^n$.

Proof. Only the $\subset$ inclusion is proven. The reverse inclusion is a direct consequence of Theorem 2.13.

Let $v \in D^*_\ell F(\bar{x}(y)$ and let L be a Lipschitz constant for F.

The case where $(v, -y)$ is a proximal normal to the graph of F at $(\bar{x}, F(\bar{x}))$ is easier and is left to the reader.

There is a sequence of proximal normals $(v^k, -y^k) \to (\bar{v}, -\bar{y})$ to the graph of F at points $(x^k, F(x^k)) \to (\bar{x}, F(\bar{x}))$ such that

$$\begin{aligned} &\operatorname{graph} F \cap \bar{B}((x^k, F(x^k)) + (\mu \|x^k - \bar{x}\|_F)(v^k, -y^k), \\ &\mu \|(v^k, -y^k)\| \|x^k - \bar{x}\|_F) = \{(x^k, F(x^k))\} \end{aligned} \tag{2.1}$$

for some $\mu > 0$. Note that for all k, $x^k \neq \bar{x}$.

Fix k. To simplify notation, the superscript k will be dropped and μ will replace $\mu \|(x^k, F(x^k)) - (\bar{x}, F(\bar{x}))\|$ while examining what happens for this fixed k.

By Definition 2.1, we have

$$\mu^2 \|(v, -y)\|^2 \leq \|z - x - \mu v\|^2 + \|F(z) - F(x) + \mu y\|^2$$

for all $z \in \bar{B}(x + \mu v, \mu \|(v, -y)\|)$. Thus

$$\begin{aligned} \mu^2 \|(v, -y)\|^2 &\leq \|z - x - \mu v\|^2 + \|F(z) - F(x) + \mu y\|^2 \\ &\leq \|z - x\|^2 + \mu^2 \|v\|^2 - 2\langle z - x, v \rangle + \|F(z) - F(x)\|^2 \\ &\quad + 2\langle F(z) - F(x), \mu y \rangle + \mu^2 \|y\|^2. \end{aligned}$$

Using the fact that F is Lipschitz and rewriting gives

$$\langle F, y\rangle(z) \geq \langle F, y\rangle(x) + \langle z - x, v\rangle - \frac{1 + L^2}{2\mu}\|z - x\|^2$$

on $B(x + \mu v, \mu\|(v, -y)\|)$. This inequality is also valid on $B(x, \mu(\|(v, -y)\| - \|v\|))$.

Therefore, the v^k form a linear sequence of proximal normals to the Lipschitz function $\langle F, y\rangle$ at $x^k \to \bar{x}$ and the result follows. □

The same result holds for D^*_M and ∂_M.

Combining this result with Theorem 2.10 yields the following corollary.

COROLLARY 2.15. *Let $F : \mathbb{R}^m \to \mathbb{R}^n$ be a Lipschitz function with $F(x) = (f_1(x), f_2(x), \ldots, f_n(x))$. Then*

$$D^*_\ell F(x)(\lambda_1, \lambda_2, \ldots, \lambda_n) \subset \partial_\ell(\lambda_1 f_1)(x) + \partial_\ell(\lambda_2 f_2)(x) + \cdots + \partial_\ell(\lambda_n f_n)(x).$$

Proof. By the above corollary

$$\begin{aligned} D^*_\ell F(x)(\lambda_1, \lambda_2, \ldots, \lambda_n) &= \partial_\ell\langle(\lambda_1, \lambda_2, \ldots, \lambda_n), F\rangle(x) \\ &= \partial_\ell\left[\sum_{i=1}^n \lambda_i f_i\right](x) \\ &\subset \sum_{i=1}^n \partial_\ell(\lambda_i f_i)(x). \end{aligned}$$

This theorem can also be used to prove a result for the LGG of the maximum of a finite number of functions.

THEOREM 2.16. *Let $g_1, g_2, \ldots, g_n$ be a finite collection of Lipschitz functions from $\mathbb{R}^m$ to $\mathbb{R}$. Then*

$$\partial_\ell \max_{i=1,2,\ldots,n} g_i(x) \subset \left\{ \sum_{i \in I(x)} \lambda_i \partial_\ell g_i(x) : \lambda_i \geq 0,\ \lambda_i = 0 \text{ if } i \notin I(x) \text{ and } \sum_{i \in I(x)} \lambda_i = 1 \right\},$$

where $I(x) = \{i : g_i(x) = \max_{j=1,\ldots,n} g_j(x)\}$.

Proof. This is a simple application of the above corollary to the function

$$F(x) = (g_1(x), g_2(x), \ldots, g_n(x))$$

composed with the convex function

$$h(y_1, y_2, \ldots, y_m) = \max_i y_i. \qquad \square$$

3. A Lagrange multiplier result. The problem considered in this section is a constrained optimization problem with inequality and set constraints. The form used is

$$(*) \qquad \min_x g_0(x) \quad \text{subject to} \quad \begin{aligned} &g_i(x) \leq 0 \quad i = 1, 2, \ldots, p, \\ &x \in C. \end{aligned}$$

Here it is assumed that all of the g_i's are Lipschitz functions from $\mathbb{R}^n$ to $\mathbb{R}$ and C is a closed subset of $\mathbb{R}^n$.

The main result is a Lagrange multiplier rule that encompasses many classical results and many results in nonsmooth analysis where equality constraints are not considered.

THEOREM 3.1. *If $\bar{x}$ is a local minimizer for* (*), *then there exist* $\lambda_i \geq 0$, $i = 0, 1, \ldots, p$, *not all zero, such that*

$$\lambda_i g_i(\bar{x}) = 0 \quad \text{for } i = 1, 2, \ldots, p$$

and

$$0 \in \sum_{i=0}^{p} \lambda_i \partial_\ell g_i(\bar{x}) + N_\ell(C, \bar{x}).$$

Proof. Note that if (*) has a local minimum at $\bar{x}$, then $\bar{x}$ is a local minimizer for

$$h(x) = \max\{g_0(x) - g_0(\bar{x}), g_1(x), \ldots, g_p(x)\} + \delta_C(x)$$

and $h(\bar{x}) = 0$. This function can be written as the sum $\phi \circ \Phi(x) + \delta_C(x)$. Here

$$\Phi(x) = (g_0(x) - g_0(\bar{x}),\ g_1(x), g_2(x),\ \ldots\ , g_p(x))$$

and

$$\phi(y) = \max\{y_0, y_1, \ldots, y_p,\}.$$

Applying the scalarization formula, Proposition 2.14, and noting that one of the components of every element of $\partial\phi(y)$ is not zero for any y with $y_0 = 0$ yields the result. Explicitly, since $0 \in \partial_\ell h(\bar{x})$,

$$\begin{aligned}
0 &\in \partial_\ell(\phi \circ \Phi + \delta_C)(\bar{x}) \\
&\subset \partial_\ell \phi \circ \Phi(\bar{x}) + \partial_\ell \delta_C(\bar{x}) \\
&\subset (\cup_{\lambda \in \partial_\ell \phi(\Phi(\bar{x}))} \partial_\ell \langle \lambda, \Phi \rangle(\bar{x})) + N_\ell(C, \bar{x}) \\
&\subset \bigcup_{\lambda \in \partial_\ell \phi(\Phi(\bar{x}))} \left[\partial_\ell(\lambda_0(g_0 - g_0(\bar{x}))(\bar{x}) + \sum_{i=0}^{p} \partial_\ell(\lambda_i g_i)(\bar{x}) + N_\ell(C, \bar{x}) \right].
\end{aligned}$$

This completes the proof. □

The idea of using a chain rule to prove that Lagrange multiplier results in nonsmooth analysis is not new. It has been used by Jourani and Thibault [7].

This can be reduced to known cases in certain situations. Using the fact that if f is Fréchet differentiable at x then $\partial_\ell f(x)$ is $\nabla f(x)$ [16], one can get the following result.

COROLLARY 3.2. *Assume that $\bar{x}$ is a minimizer for* (*). *If $g_0, g_1, \ldots, g_m$ are Lipschitz and Fréchet differentiable at $\bar{x}$, then for some $\lambda_i \geq 0$, $i = 0, 1, \ldots, p$, not all zero,*

$$\lambda_i g(\bar{x}) = 0 \quad \text{for } i = 1, 2, \ldots, p,$$

and

$$0 \in \sum_{i=0}^{p} \lambda_i \nabla g_i(\bar{x}) + N_\ell(C, \bar{x}).$$

Proof. This follows directly from the above theorem using the facts that $\partial_\ell(\alpha f)(x) = \alpha \partial_\ell f(x)$ if $\alpha \geq 0$ and if f is Fréchet differentiable at x, then $\nabla(-f)(x) = -\nabla f(x)$. □

The following is a simplification of the Lagrange multiplier results in [2] and [3]. In this result ∂f is the generalized gradient of Clarke.

COROLLARY 3.3. *If $g_0, g_1, \ldots, g_m$ are Lipschitz, then for some $\lambda_i \geq 0$, $i = 0, 1, \ldots, p$, not all zero,*

$$\lambda_i g(\bar{x}) = 0 \quad \text{for } i = 1, 2, \ldots, m,$$

and

$$0 \in \sum_{i=0}^{p} \lambda_i \partial g_i(\bar{x}) + N_C(C, \bar{x}).$$

Proof. Simply use the facts that $N_\ell(C, x) \subset N_C(C, x)$ and $\partial_\ell f(x) \subset \partial f(x)$ to rewrite Theorem 3.1. □

The same can be done for the MGG.

COROLLARY 3.4 [11]. *If $g_0, g_1, \ldots, g_m$ are Lipschitz, then for some $\lambda_i \geq 0$, $i = 0, \ldots, p$, not all zero,*

$$\lambda_i g(\bar{x}) = 0 \quad \text{for } i = 1, 2, \ldots, m,$$

and

$$0 \in \sum_{i=0}^{p} \lambda_i \partial_M g_i(\bar{x}) + N_M(C, \bar{x}).$$

The following gives the result corresponding to Theorem 3.1 for the GGP [9]. In most ways it is much weaker than the result of Ioffe [6]. Ioffe's multiplier rule includes equality, inequality, and set constraints. His proof relies heavily on the convexity of GGP and uses a convex set constraint.

In what follows, ∂_P denotes the GGP and $N_P(C, x)$ denotes the corresponding normal cone.

COROLLARY 3.5. *If $g_0, g_1, \ldots, g_m$ are Lipschitz, then for some $\lambda_i \geq 0$, $i = 0, \ldots, p$, not all zero,*

$$\lambda_i g(\bar{x}) = 0 \quad \text{for } i = 1, 2, \ldots, m,$$

and

$$0 \in \sum_{i=0}^{p} \lambda_i \partial_P g_i(\bar{x}) + N_P(C, \bar{x}).$$

Proof. One simply uses the fact that the Michel–Penot objects are the closed convex hulls of the "linear" objects. □

4. Examples. In this section two examples are given and the differences between Theorem 3.1 and Corollaries 3.3 and 3.4 are explored. The first example is a simple example that demonstrates how the differences between the normal cones can affect the set of points satisfying the necessary conditions.

Example 4.1. Let C be the union of the pairwise intersections of the four closed balls of radius 1 in $\mathbb{R}^2$ centered at $(1, 0)$, $(0, 1)$, $(-1, 0)$ and $(0, -1)$. Consider the problem

$$\min f(x, y) \quad \text{subject to} \quad (x, y) \in C.$$

For illustration it is assumed that f is continuously differentiable on an open set containing C.

According to the results in §3, the necessary conditions for having a minimum at $(0,0)$ are that

$$0 \in \nabla f(0,0) + N(C,(0,0)),$$

where $N(C,\cdot)$ is one of the Clarke, Mordukhovich, or linear normal cones.

For this example, if f has a local minimum at $(0,0)$, then $\nabla f(0,0) = (0,0)$. One would like this to be reflected in the necessary conditions.

Since $N_C(C,(0,0)) = \mathbb{R}^2$, the necessary condition is satisfied for any f when using the CNC. This is much larger than desired.

The Mordukhovich cone is much better. Here $N_M(C,(0,0))$ is the union of the x-axis and the y-axis. This means that $(0,0)$ is a critical point for f's whose gradient is on one of the axes. This is much better.

In this example the LNC is $\{(0,0)\}$. This implies that the only functions that make $(0,0)$ a critical point are those whose gradient is $(0,0)$. This is as good as is possible.

One can also have cases where the difference between LGG and MGG is as pronounced as that between LGG and CGG in the preceding example.

Example 4.2. Assume that $f : \mathbb{R}^2 \to \mathbb{R}$ is Lipschitz and consider the problem

$$\min f(x,y) \quad \text{subject to} \quad (x,y) \in C,$$

where $C = \mathbb{R}^2 \setminus \bigcup_{i=1}^{\infty} B((i^{-2},0), i^{-4}/4)$.

For this set

$$N_M(C,(0,0)) = N_C(C,(0,0)) = \mathbb{R}^2,$$

whereas

$$N_\ell(C,(0,0)) = \{(0,0)\}.$$

This means that the origin is always a critical point for both MGG and CGG, but is only a critical point for LGG if $0 \in \partial_\ell f(0,0)$.

5. An open question. An interesting open question not answered in this paper is: Does a Lagrange multiplier result for LGG hold if equality constraints are included. The problem we consider is

$$(**) \qquad \min_x g_0(x) \quad \text{subject to} \quad \begin{array}{ll} g_i(x) \le 0 & i = 1,2,\dots,p, \\ g_j(x) = 0 & j = p+1, p+2, \dots, m, \\ x \in C. & \end{array}$$

Here it is assumed that all of the g_i's are Lipschitz functions from $\mathbb{R}^n$ to $\mathbb{R}$ and C is a closed subset of $\mathbb{R}^n$.

The simple proof used in §3 is not valid since the chain rule is not strong enough. This proof does work for MGG since MGG has the following chain rule.

In the following result the singular gradient of Mordukhovich is used. It is the set of limits of $\alpha_k v^k$, where the v^k form a sequence of proximal subgradients to f at $x^k \to \bar{x}$ and $\alpha_k \searrow 0$. We denote this singular generalized gradient by $\partial_M^\infty f(\bar{x})$.

THEOREM 5.1 [11]. *Let $G : \mathbb{R}^m \to \mathbb{R}^n$ be Lipschitz and let $f : \mathbb{R}^n \to \mathbb{R}$ be* lsc. *If*

$$y^* \in \partial_M^\infty f(G(x)) \quad \textit{and} \quad 0 \in D_M^* G(x)(y^*) \quad \textit{implies} \quad y^* = 0,$$

then

$$\partial_M f \circ G(x) \subset D_M^* G(x) \partial_M f(G(x)).$$

Given this, we can prove the following result.

THEOREM 5.2 [11]. *If $\bar{x}$ is a local minimizer for* (**), *then there exist $\lambda_i \geq 0$, $i = 0, 1, \ldots, p$ and λ_j, $j = p+1, \ldots, m$, not all zero, such that*

$$\lambda_i g_i(\bar{x}) = 0 \quad \textit{for} \quad i = 1, 2, \ldots, m$$

and

$$0 \in \sum_{i=0}^{p} \lambda_i \partial_M g_i(\bar{x}) + \sum_{j=p+1}^{m} \partial_M(\lambda_i g_i(\bar{x})) + N_M(C, \bar{x}).$$

Proof. Note that if (*) has a local minimum at $\bar{x}$ then $\bar{x}$ is a local minimizer for the sum $\phi \circ \Phi(x) + \delta_C(x)$. Here

$$\Phi(x) = (g_0(x) - g_0(\bar{x}), g_1(x), g_2(x), \ldots, g_m(x))$$

and

$$\phi(y) = \begin{cases} \max\{y_0, y_1, \ldots, y_p, \} & \text{if } y_{p+1} = \cdots = y_m = 0, \\ \max\{y_0, y_1, \ldots, y_p, |y_{p+1}|, \ldots, |y_m| & \text{otherwise}\}. \end{cases}$$

First note that the zero vector is only in $\partial_M^\infty \phi(0)$, not in $\partial_M \phi(0)$. In addition, the first p components of any vector in $\partial_M \phi(0) \cup \partial_M^\infty \phi(0)$ must be nonnegative.

Applying Theorem 5.1, there is a nonzero $y^* \in \partial_M \phi(0) \cup \partial_M^\infty \phi(0)$ such that either

$$0 \in D_M^* \Phi(\bar{x})(y^*)$$

or

$$0 \in D_M^* \Phi(\bar{x})(y^*) + N_M(C, \bar{x}).$$

Since $0 \in N_M(C, \bar{x})$, the second case includes the first.

Applying the scalarization formula for $D_M^* \Phi$ similar to Proposition 2.14 yields the result. Explicitly, since $0 \in \partial_M h(\bar{x})$,

$$\begin{aligned} 0 &\in \partial_M(\phi \circ \Phi + \delta_C)(\bar{x}) \\ &\subset \partial_M \phi \circ \Phi(\bar{x}) + \partial_M \delta_C(\bar{x}) \\ &\subset (\cup_{\lambda \in \partial_M \phi(\Phi(\bar{x}))} \partial_M \langle \lambda, \Phi \rangle(\bar{x})) + N_M(C, x)\text{bar} \\ &\subset \bigcup_{\lambda \in \partial_M \phi(\Phi(\bar{x}))} \left[\partial_M(\lambda_0(g_0 - g_0(\bar{x}))(\bar{x}) + \sum_{i=0}^{p} \partial_M(\lambda_i g_i)(\bar{x}) + N_M(C, \bar{x}) \right]. \end{aligned}$$

This completes the proof. □

Unfortunately, if the set constraint is included, the above result does not hold for LGG. The following example shows this.

Example 5.3. Consider the problem

$$\min f(x, y) \quad \text{subject to} \quad \begin{aligned} &g(x, y) = x^2 + (y+2)^2 - 4 = 0, \\ &(x, y) \in \bar{B}((0, 1), 1) \cup \bar{B}((0, -1), 1). \end{aligned}$$

The only feasible point for this problem is $(0,0)$.

Here one has that $g(x,y)$ is C^∞ with $\nabla g(0,0) = (0,4)$ and $N_\ell(C,(0,0)) = \{(0,0)\}$.

The desired Lagrange multiplier rule would be that for some $\lambda_0 \geq 0$ and λ_1, not both zero, one has

$$\begin{aligned}(0,0) &\in \lambda_0 \partial_\ell f(0,0) + \partial_\ell(\lambda_1 g)(0,0) + N_\ell(C,(0,0)) \\ &= \lambda_0 \partial_\ell f(0,0) + \lambda_1(0,4).\end{aligned}$$

If one takes $f(x,y) = x$ this would mean that

$$(0,0) = \lambda_0(1,0) + \lambda_1(0,4)$$

or that $\lambda_0 = \lambda_1 = 0$.

This means that a general multiplier rule does not hold with a set constraint and equality constraints even under the condition that all functions are C^∞!

Note 5.4(a). This example shows that it may be necessary to use either the Clarke or Mordukhovich normal cone if one wants to include arbitrary set constraints with equality constraints. The author does not know of any other well-defined normal cones contained in either Clarke or Mordukhovich that will work. The cone of Michel and Penot will not work since it coincides with the LNC in this case.

Note 5.4(b). If one replaces the set constraint by any one of the equivalent functional constraints

$$d(C,(x,y)) = 0,$$

$$d(C,(x,y)) \leq 0,$$

or

$$\min\{(x^2 + (y-1)^2), (x^2 + (y+1)^2)\} - 1 \leq 0,$$

the "multiplier rule" gives that $(0,\ 0)$ is a critical point.

At this point it is still unknown if a Lagrange multiplier result that includes equality constraints holds for LGG.

REFERENCES

[1] F. H. Clarke, *Generalized gradients and applications*, Trans. Amer. Math. Soc., 205 (1975), pp. 247–262.

[2] ———, *Methods of dynamic and nonsmooth analysis*, CBMS–NSF Regional Conference Series in Applied Mathematics, 57, Society for Industrial and Applied Mathematics, Philadelphia, 1989.

[3] ———, *Optimization and nonsmooth analysis*, Classics in Applied Mathematics, 5, Society for Industrial and Applied Mathematics, Philadelphia, 1990.

[4] A. D. Ioffe, *Approximate subdifferentials and applications. I, The finite dimensional theory*, Trans. Amer. Math. Soc., 281 (1984), pp. 389–416.

[5] ———, *Approximate subdifferentials and applications. II, Mathematika*, 33 (1986), pp. 111–128.

[6] ———, *A Lagrange multiplier rule with small convex-valued subdifferentials for nonsmooth problems of mathematical programming involving equality and nonfunctional constraints*, Math. Programming, 58 (1993), pp. 137–145.

[7] A. Jourani and L. Thibault, *Approximations and metric regularity in mathematical programming in Banach spaces*, Math. Oper. Res., 18 (1993), pp. 390–401.

[8] A. Ja. Kruger and B. Sh. Mordukhovich, *Minimization of nonsmooth functionals in optimal control problems*, Izv. Akad. Nauk. SSSR Tekhn. Kibernet, (1978), pp. 176–183. In English: Engrg. Cybernetics, 16(1978), pp. 126–133.

[9] P. MICHEL, J. P. PENOT, *Calcul sous-différential por des functions lipschitziennes et non lipschitziennes*, C. R. Acad. Sci. Paris Ser. I Math., 298 (1984), pp. 269–272.
[10] B. S. MORDUKHOVICH, *Maximum principle in the problem of time optimal control with nonsmooth constraints*, J. Appl. Math. Mech., 40 (1976), pp. 960–969.
[11] ———, *Approximation methods in problems of optimization and control*, Nauka, Moscow 1988. (In Russian.)
[12] ———, *Complete characterization of openness, metric regularity and Lipschitzian properties of multifunction*, to appear.
[13] J. S. TREIMAN, *Shrinking generalized gradients*, Nonlinear Anal., 12 (1988), pp. 1429–1450.
[14] ———, *Finite dimensional optimality conditions*: B-*gradients*, J. Optim. Theory Appl., 62 (1989), pp. 139–150.
[15] ———, *Optimal control with small generalized gradients*, SIAM J. Control Optim., 28 (1990), pp. 720–732.
[16] ———, *The linear nonconvex generalized gradient*, Proc. First Internat. Congress on Nonlinear Anal., to appear.

SIAM J. OPTIMIZATION
Vol. 5, No. 3, pp. 681–694, August 1995

011

ON THE SIMULATION AND CONTROL OF SOME FRICTION CONSTRAINED MOTIONS*

ROLAND GLOWINSKI† AND ANTHONY J. KEARSLEY‡

Abstract. In this paper, some issues involved with numerical simulation and control of some elasto-dynamic systems are discussed. The motivation is the simulation of dry or *Coulomb* friction in the joints that link together remote manipulator systems used in aerospace operations (for example, space shuttle remote manipulator systems). The goal here is to develop numerical techniques to simulate and control these systems, while properly modeling the Coulomb friction. The numerical procedure described employs a finite difference time discretization in conjunction with a vector of multipliers that predicts the friction effect for all time. In addition to this discrete multiplier technique an associated regularization procedure that greatly improves the behavior of these multipliers is also presented. Numerical examples conclude the paper.

Key words. Coulomb friction, direct search methods, nonsmooth optimization

AMS subject classifications. 49J15, 70Q05

1. Introduction.

In this paper, we discuss the simulation and control of some elasto-dynamic systems with dry friction. The phenomenon of dry or *Coulomb* friction has been described and analyzed in [KikO88] and in [CamOK82] (see also [Ren92]). In [Cab81], Coulomb friction is analyzed in the motion of a string. These methods all approach these physical problems as time-dependent variational inequalities (see [DuvL76]). The spatial semidiscretization of these problems gives rise to systems like the one examined here.

We consider the simple time-dependent problem

$$M\ddot{x} + Ax + C\lambda = f, \qquad t \in (0,T], \tag{1.1}$$

$$x(0) = x_0, \qquad \dot{x}(0) = x_1, \tag{1.2}$$

$$\lambda_i(t) = 0 \text{ if } c_{ii} = 0, |\lambda_i(t)| \le 1 \text{ if } c_{ii} > 0 \text{ and } C\lambda(t)\cdot\dot{x}(t) = \sum_{i=1}^{d} c_{ii}|\dot{x}_i(t)|. \tag{1.3}$$

We are using the standard inner product of $\Re^d$, i.e., for $y, z \in \Re^d$ we have $y \cdot z = \sum_{i=1}^{d} y_i z_i$. In this system, $x_i(t)$ denotes the displacement of the ith component at time t. The mass matrix $M \in \Re^{d\times d}$ is a symmetric and positive definite matrix. The stiffness matrix, $A \in \Re^{d\times d}$ is symmetric and positive semidefinite. The friction matrix $C \in \Re^{d\times d}$ is diagonal with only nonnegative diagonal entries. Matrices M, A, and C are all assumed to be constant with respect to time t. The vectors $x, \dot{x}, \ddot{x}, \in \Re^d$ represent displacement, velocity, and acceleration, respectively. Here $\dot{x} = \frac{dx}{dt}$ and

* Received by the editors June 11, 1993; accepted for publication (in revised form) February 15, 1994. The work of these authors was supported by the Texas Board of Higher Education grant 003652156-ARP and the Patricia R. Harris Fellowship at Rice University. Use of the Intel iPSC/860 was provided by the Center for Research on Parallel Computation, National Science Foundation Cooperative Agreement CCR-9120008, and CDA-8619893 with support from the Keck Foundation.

† Department of Mathematics, University of Houston, 4800 Calhoun Road, Houston, Texas 77204-3476.

‡ Department of Computational and Applied Mathematics, P. O. Box 1892, Rice University, Houston, Texas 77251-1892 (kearsley@masc.rice.edu).

$\ddot{x} = \frac{d^2x}{dt^2}$. Also, $x_0, x_1 \in \Re^d$ are the initial displacement and velocity, respectively. The function f models the external forces acting on the system, and we will assume that $f \in L^2(0,T;\Re^d)$. Finally, $\lambda \in L^\infty(0,T;\Re^d)$ is the normalized coefficient of Coulomb friction.

The system described in (1.1)–(1.3) is equivalent to a second order time-dependent variational inequality whose solution requires finding a function $x \in H^2(0,T;\Re^d)$ such that

$$\int_0^T (M\ddot{x} + Ax - f) \cdot (y - \dot{x})dt + \int_0^T (j(y) - j(\dot{x}))dt \geq 0 \quad \forall y \in L^2(0,T;\Re^d), \tag{1.4}$$

$$x(0) = x_0, \qquad \dot{x}(0) = x_1. \tag{1.5}$$

Here we have

$$j(y) = \sum_{i=1}^{d} c_{ii}|y_i| \quad \forall y \in \Re^d. \tag{1.6}$$

The space $H^2(0,T;\Re^d)$ is a Hilbert space with associated inner product

$$\langle y, z \rangle = \sum_{j=0}^{2} \int_0^T \frac{d^j y}{dt^j} \cdot \frac{d^j z}{dt^j} dt. \tag{1.7}$$

The functional $j(\cdot)$ is convex and continuous. It is also nonsmooth, unless $c_{ii} = 0$, for all $i = 1, 2, \ldots, d$.

Existence and uniqueness for the solution of (1.1)–(1.3) can be proved using the methods from [DuvL76]. We will not prove these results here.

2. Simulation. Let $N, \Delta t, T$ be positive scalars denoting the number of time steps in the discretized problem, the length of the time step, and the final time, respectively. Here we have $\Delta t = \frac{T}{N}$. We seek an approximate solution to problem (1.1)–(1.3), say, x^n for every $n\Delta t$. Similarly, we approximate $f(n\Delta t)$ by f^n. In this article, we will assume that f is sufficiently smooth ($f \in C^0([0,T];\Re^d)$) so that we can take $f^n = f(n\Delta t)$. We then approximate system (1.1)–(1.3) by

$$M\frac{1}{\Delta t^2}(x^{n+1} + x^{n-1} - 2x^n) + A(\alpha x^{n+1} + \alpha x^{n-1} + (1-2\alpha)x^n) + C\lambda^n = f^n \quad \forall n \geq 0; \tag{2.1}$$

$$\lambda_i^n = 0 \quad \text{if} \quad c_{ii} = 0, \tag{2.2}$$

$$|\lambda_i^n| \leq 1 \quad \text{and} \quad \lambda_i^n(x_i^{n+1} - x_i^{n-1}) = |x_i^{n+1} - x_i^{n-1}| \quad \text{if} \quad c_{ii} > 0, \tag{2.3}$$

$$x^0 = x_0 \quad \text{and} \quad x^1 - x^{-1} = 2x_1\Delta t, \tag{2.4}$$

$$\lambda_i^0 = 0 \quad \text{if either} \quad c_{ii} = 0 \quad \text{or} \quad \dot{x}_i^0(0) = 0 \quad \text{and} \quad c_{ii} > 0, \tag{2.5}$$

$$\lambda_i^0 = \quad \text{sign}\,(\dot{x}_i(0)) \quad \text{if} \quad \dot{x}_i(0) \neq 0 \quad \text{and} \quad c_{ii} > 0. \tag{2.6}$$

Discretization (2.1)–(2.6) has been investigated in [DeaGKN90] for $\alpha \in [0, \frac{1}{2}]$ and stability properties of the scheme were established. Some convergence properties were also established.

THEOREM 2.1 (Dean, Glowinski, Kuo, Nasser). *Suppose* (x, λ) *is the solution of* (2.1)–(2.6). *Let* $\lambda_{\Delta t}$ *be the function of* t, *defined by*

$$\lambda_{\Delta t}(t) = \lambda^0 \quad \textit{when} \quad t \in \left[0, \frac{\Delta t}{2}\right) \tag{2.7}$$

$$\lambda_{\Delta t}(t) = 0 \quad \textit{when} \quad t \in \left(T - \frac{\Delta t}{2}, T\right] \tag{2.8}$$

$$\lambda_{\Delta t}(t) = \lambda^n \quad \textit{when} \quad t \in \left(n\Delta t - \frac{\Delta t}{2}, n\Delta t + \frac{\Delta t}{2}\right). \tag{2.9}$$

Then (2.1)–(2.6) *is an unconditionally stable scheme if* $\alpha \in [\frac{1}{4}, \frac{1}{2}]$ *and the following convergence results hold:*

1. $\lim_{\Delta t \to 0} \max_{0 \le n \le N} \|x^n - x(n\Delta t)\| = 0$;
2. $\lim_{\Delta t \to 0} \max_{0 \le n \le N-1} \|\frac{1}{\Delta t}(x^{n+1} - x^n) - \dot{x}((n + \frac{1}{2})\Delta t)\| = 0$;
3. $\lim_{\Delta t \to 0} \lambda_{\Delta t} = \lambda$, *weakly* * *in* $L^\infty(0, T; \Re^d)$.

Remark 2.1. If $\alpha \in [0, \frac{1}{4})$ the above convergence results still hold, the stability condition in that case being

$$\Delta t < [(\frac{1}{4} - \alpha)\delta_M]^{-\frac{1}{2}}$$

with δ_M the largest eigenvalue of the matrix $M^{-1}A$.

Following [DeaGKN92], we can solve the nonlinear system (2.1)–(2.6) by observing that the conditions on λ in (2.1)–(2.6) can be rewritten as

$$\lambda^n = P_\Lambda\left(\lambda^n + \frac{r}{2\Delta t} C(x^{n+1} - x^{n-1})\right) \quad \forall r \ge 0 \quad \forall n \ge 0. \tag{2.10}$$

The projector $P_\Lambda : \Re^d \to \Lambda$, is the orthogonal projection operator from $\Re^d$ onto the set Λ, where Λ is a closed convex set defined by

$$\Lambda = \{\lambda : \lambda \in \Re^d, |\lambda_i| \le 1 \quad \text{and} \quad \lambda_i = 0 \quad \text{if} \quad c_{ii} = 0\}. \tag{2.11}$$

We have for all $\mu \in \Re^d$, $P_\Lambda(\mu) = z$ with $z_i = 0$ if $c_{ii} = 0$ and $z_i = \min\{1, \max(\mu_i, -1)\}$ if $c_{ii} > 0$.

We can numerically solve the class of problems defined by (2.1)–(2.6) using the techniques developed and described in [Glo84].

In the particular case of problem (2.1)–(2.6) and for $\alpha = \frac{1}{4}$, these techniques yield the following algorithm for computing the pair (x^{n+1}, λ^n):

$$\lambda^{n,0} \in \Lambda; \tag{2.12}$$

for $k \geq 0$, $\lambda^{n,k}$ being known in Λ, solve the linear system

$$\left(M + \frac{\Delta t^2}{4}A\right) x^{n+1,k} = \Delta t^2(f^n - C\lambda^{n,k}) + M(2x^n - x^{n-1}) - \frac{\Delta t^2}{4}A(2x^n + x^{n-1}) \tag{2.13}$$

and update $\lambda^{n,k}$ by

$$\lambda^{n,k+1} = P_\Lambda\left(\lambda^{n,k} + \frac{r}{2\Delta t}C(x^{n+1,k} - x^{n-1})\right). \tag{2.14}$$

We stop iterating when

$$\frac{\|x^{n+1,k} - x^{n+1,k-1}\|}{\|x^{n+1,k-1}\| + 1} \leq \eta \tag{2.15}$$

for some well chosen norm $\|\cdot\|$ and parameter η.

When applying algorithm (2.12)–(2.15) to the solution of system (2.1)–(2.6), for some test problem (1.1)–(1.3), the numerical experiments show very good convergence properties for the approximate displacement and velocity (see, for example, Figs. 1, 2, 5 and 6). However, the approximate multiplier $\lambda_{\Delta t}$ displays violent oscillations (as shown in Figs. 3 and 7); naturally, this behavior is compatible with the convergence Theorem 2.1 which only guarantees weak convergence for $\lambda_{\Delta t}$. Some applications, however, require more accurate discrete multipliers since, after all, these multipliers measure friction forces. It is for this reason that a regularization procedure has been developed in [DeaGKN92]. This procedure is based on observing first that (2.14) is a discrete form of

$$\lambda = P_\Lambda(\lambda + rC\dot{x}), \tag{2.16}$$

then by regularizing (2.16) via

$$\epsilon\dot{\lambda} + \lambda = P_\Lambda(\lambda + rC\dot{x}), \tag{2.17}$$

with $\epsilon > 0$, and finally by discretizing (2.17) by

$$\epsilon\left(\frac{\lambda^n - \lambda^{n-1}}{\Delta t}\right) + \lambda^n = P_\Lambda\left(\lambda^n + rC\frac{x^{n+1} - x^{n-1}}{2\Delta t}\right). \tag{2.18}$$

This procedure can be viewed as a dynamical Tychonoff regularization procedure. It is quite clear that relation (2.17) implies that the regularized multiplier is a Lipschitz continuous function of t. It can be shown that if $\lim_{\Delta t \to 0} \frac{\epsilon}{\Delta t} = 0$, then the convergence results of Theorem 2.1 still hold.

In practice, inspired by (2.18), we shall use the regularized variant of algorithm (2.12)–(2.15) obtained by replacing (2.14) by

$$\lambda^{n,k+1} = \frac{\Delta t}{\epsilon + \Delta t}P_\Lambda\left(\lambda^{n,k} + \frac{r}{2\Delta t}C(x^{n+1,k} - x^{n-1,k})\right) + \frac{\epsilon}{\epsilon + \Delta t}\lambda^{n-1}; \tag{2.19}$$

the parameter ϵ must be chosen so that $\lim_{\Delta t \to 0} \frac{\epsilon}{\Delta t} = 0$.

The above regularization procedure has a double effect.

- It substantially improves, for the same value of r, the convergence of algorithm (2.12)–(2.15).

• It improves the convergence of $\lambda_{\Delta t}$ to λ without affecting the convergence of the approximate displacement and velocity to their respective limit, this is seen in Table 5. Indeed from the figures and Table 5 we can conjecture that the weak-* convergence of the discrete multipliers has been replaced by a strong convergence of the discrete regularized multipliers $L^s(0,T;\Re^d)$ for all $s \in [1,+\infty)$. Proving this result would be of interest.

Let δ be the largest eigenvalue of the matrix $([M + \frac{\Delta t^2}{4}A]^{-1}C^2)$. Numerical observations suggest that a value of $r \approx \frac{2}{\delta \Delta t}$ yields the best results. While apparently this value of r is optimal, $r \in (0, \frac{4}{\delta \Delta t})$ is necessary to guarantee the important fixed-point property that yields convergence of the pair $(x^{n+1,k}, \lambda^{n,k})$.

3. Control. There are many practical applications where we are interested in controlling the time evolution of the system modelled by (1.1)–(1.3). In particular we may require x and/or $\dot{x}$ to have preassigned values at some final time $t = T$.

This is certainly applicable to the control of remote manipulator systems, where final velocities must be kept small. Here we choose target velocity $\dot{x}_T$ and target position x_T. The cost function employed will use penalty terms to attain target states. The resulting problem will look like

$$M\ddot{x} + Ax + C\lambda = f + Bv, \qquad t \in (0,T], \tag{3.1}$$

$$x(0) = x_0, \qquad \dot{x}(0) = x_1, \tag{3.2}$$

$$\lambda_i(t) = 0 \text{ if } c_{ii} = 0, \qquad |\lambda_i(t)| \leq 1 \text{ if } c_{ii} > 0, \text{ and} \tag{3.3}$$

$$C\lambda(t) \cdot \dot{x}(t) = \sum_{i=1}^{d} c_{ii}|\dot{x}_i(t)|,$$

where we want to solve

$$\min_{v \in \Re^d} J(v) = \int_0^T Nv \cdot v dt \tag{3.4}$$

subject to

$$x(T) = x_T, \tag{3.5}$$

$$\dot{x}(T) = \dot{x}_T. \tag{3.6}$$

The vector, $v \in \Re^p$, is the vector of control variables. The matrix $B \in \Re^{d \times p}$ will dictate how the control is administered to the system. Likewise, the matrix $N \in \Re^{p \times p}$, is a matrix that may be different from the identity if some parts of the system are more difficult to control than others. For simplicity we will assume that $p = d$ and that both B and N are multiples of the identity matrix. The algorithm used to solve this *nonsmooth* nonlinear programming problem (3.4)–(3.6) will be discussed in §4.

4. Optimization technique. Problem (3.4)–(3.6) is an equality constrained nonlinear programming problem. The equality constraints are *not* differentiable with respect to the control variables, v, prohibiting the use of very fast algorithms that require smoothness of the constraints (e.g., see [GilMW81]). In fact, we expect that at the solution of (3.4)–(3.6), say v^*, the constraints will not be differentiable with respect to v. Whenever the velocity of any of the components of our dynamical system vanishes, the derivative with respect to v will not exist. For this reason we employed an algorithm that used no derivative information.

In [DenT91] an algorithm for unconstrained optimization that requires no derivative information was suggested. While algorithms for unconstrained minimization that required no derivative information (usually referred to as *direct search* methods) are not new (see, for example, [Cea71] and the references therein), a global first-order stationary point convergence theory for this particular method has only recently been developed in [Tor93]. A parallel implementation of the algorithm has also been developed and tested (see [Tor92]). Recently, this algorithm and its implementation have been modified to handle constraints [KeaTT93]. The method samples points on an evolving simplex, moving to the vertex on the simplex with value closest to optimality. The simplex then expands or contracts depending on where the point closest to optimality was located. The contractions reduce the size of the simplex; the procedure is halted when the lengths of the edges in the simplex fall below a tolerance set by the user. Direct search methods of this sort typically do not demonstrate rapid local convergence, but they are extremely robust and far less susceptible than faster higher-order methods to the difficulties introduced when functions are nonsmooth or the data is noisy.

Because we are dealing with only equality constraints, minimization problem (3.4)–(3.6) can be handled in a straightforward manner. After each simplex is constructed, a penalty function is evaluated. Let ρ_1 and ρ_2 be two large constants. Then we employ a penalty function, such as

$$P(v;\rho_1,\rho_2) = \int_0^T Nv \cdot v dt + \rho_1 \|x(T) - x_T\|_2 + \rho_2 \|\dot{x}(T) - \dot{x}_T\|_2. \tag{4.1}$$

We can now minimize this penalty function and the new problem is now a continuous, nonsmooth, nonconvex unconstrained optimization problem. Early numerical testing included trying various updating strategies for the penalty parameters ρ_1 and ρ_2. Most of our attempts at dynamically adjusting these parameters for penalty function (4.1) failed to significantly improve the performance of our direct search method algorithm. The dynamic penalty parameter updating schemes,

$$\rho_1^+ \leftarrow \min\{10^6, (\rho_1^c + \|x(T) - x_T\|_\infty)\}, \tag{4.2}$$

$$\rho_2^+ \leftarrow \min\{10^6, (\rho_2^c + \|\dot{x}(T) - \dot{x}_T\|_\infty)\} \tag{4.3}$$

for ρ_1 and ρ_2 were employed for comparison purposes only and remained fairly ineffective when compared to the performance with constant $\rho_1 = \rho_2 = 10^6$ (see Tables 2 and 4). The superscript c denotes a quantity associated with the minimizer on the previous simplex.

More complicated penalty functions for (PDS) involving penalty parameter schemes, dynamical choices of norms for penalization, and exploiting feasibility are

discussed in [KeaTT93]. One such suggestion is to use the penalty function,

$$\Pi_K(v;\rho_3) = \int_0^T Nv \cdot v dt + \rho_3(G_K(v)) + Z_K(v, v^c), \tag{4.4}$$

where $Z_K(v, v^c)$ is

(4.5)

$$Z_K(v, v^c) = \min\left\{\frac{G_K(v^c)}{\max\{G_K(v), 10^{-8}\}}, 1\right\} \max\left\{\int_0^T Nv^c \cdot v^c dt - \int_0^T Nv \cdot v dt, 0\right\}$$

and $G_K(v)$ is

$$G_K(v) = \|(x(T) - x_T, \dot{x}(T) - \dot{x}_T)^T\|_K. \tag{4.6}$$

Here v^c denotes the value of control variables yielding smallest penalty function value on the preceding simplex. The augmentation function $Z_K(v, v^c)$ incorporates a balance between decreasing infeasibility and moving towards optimality. The choice of norm, K,

$$K = \begin{cases} \infty & \text{if} \quad G_2(v^c) \geq 2, \\ 2 & \text{if} \quad 1 < G_2(v^c) < 2, \\ 1 & \text{if} \quad G_2(v^c) \leq 1, \end{cases} \tag{4.7}$$

depends on the distance from feasibility. The parameter ρ_3 can be chosen initially to be a large constant or updated dynamically. The updating strategy employed here combines (4.2) and (4.3)

$$\rho_3^+ \leftarrow \frac{1}{2}(\rho_1^+ + \rho_2^+). \tag{4.8}$$

It is worth commenting that consistent performance of (PDS) with respect to the penalty parameters ρ_1, ρ_2, and ρ_3 when minimizing penalty functions (4.1) and (4.6) is characteristic of direct search methods. This is inherent in the way search directions and trial points are generated independently of any quantity dependent on the penalty parameters. Moreover, a new point is selected from the set of trial points by having the smallest associated objective function value. There are no minimum decrease or maximum allowable change conditions enforced. In this way, there is no ill conditioning introduced into the problem by penalty parameters that are too large. The price paid for this robustness is, as mentioned above, no rapid local convergence.

5. Numerical results. The test problems presented here are small ones, and are intended only to demonstrate the effectiveness of the simulation method and regularization. Actually, much larger problems have been solved using similar techniques, as shown for example in [DeaGKN90] where an algorithm like (2.12)–(2.15) has been used to simulate the motion of an elastic string in the presence of dry friction; employing the additional regularization procedure discussed here has been straightforward and the corresponding results will be given in a forthcoming article [GloK95].

In order to make our first test problem more significant, we have chosen for A the 3×3 Hilbert matrix because it is a small matrix with a large condition number (≈ 524) and more importantly it strongly couples the components of the state variable $x(t)$, implying a fairly complicated dynamics particularly for the friction multipliers.

We have also chosen a final time, T, relatively small. For this problem we prescribed a desired final position, but no target velocity.

The second test problem can be physically motivated by considering a spring-mass system consisting of $(d+1)$ springs with known spring constants and d known masses. The springs are suspended between two fixed ends and connected by exactly one of the masses between every two springs. The spring constants determine the stiffness matrix A and the masses determine the matrix M.

We solve the initial value problem (3.1)–(3.3) to evaluate the objective function (3.4) and the constraints (3.5) and (3.6). The following numerical experiments demonstrate the behavior of the algorithm and specifically the multipliers, with and without the regularization. For simplicity we took $d = 3$, with the following values and functions:

TEST PROBLEM (TP1)

- $x_0 = (1,2,3)^T$ (initial positions),
- $x_1 = (1,1,1)^T$ (initial velocities),
- $m_{11} = 4\ m_{22} = 5\ m_{33} = 6$ (diagonal matrix),
- $a_{ij} = \frac{1}{(i+j)-1}$ (Hilbert matrix),
- $c_{ii} = 10$ (diagonal matrix),
- $f_i(t) = -100e^{-10t}$ (external forces),
- $\Delta t = .0125$ (time step),
- $T = 5$ (final time),
- $\epsilon = (\Delta t)^{6/5}$,
- $r = \frac{2}{\delta \Delta t}$,
- $x_T = (2,0,-2)^T$ (desired final position),
- $n_{ii} = \frac{1}{3}$ (diagonal matrix),
- $b_{ii} = 1$ (identity matrix).

TEST PROBLEM (TP2)

- $x_0 = (1,3,5)^T$ (initial positions),
- $x_1 = (2,2,2)^T$ (initial velocities),
- $m_{11} = 1\ m_{22} = 2\ m_{33} = 3$ (diagonal matrix),
- $\kappa_1 = 1\ \kappa_2 = 2\ \kappa_3 = 3\ \kappa_4 = 4$ (spring constants),
- $a_{11} = 3\ a_{22} = 4\ a_{33} = 5\ a_{23} = a_{32} = -3\ a_{12} = a_{21} = -2$ (stiffness matrix),
- $c_{11} = 6\ c_{22} = 5\ c_{33} = 4$ (diagonal matrix),
- $f_i(t) = -10\sin(\frac{3\pi t}{5})e^{-t^2}$ (external forces),
- $\Delta t = .01$ (time step),
- $T = 10$ (final time),
- $\epsilon = (\Delta t)^{6/5}$,
- $r = \frac{2}{\delta \Delta t}$,
- $x_T = (5,10,15)^T$ (desired final position),
- $\dot{x}_T = (0,0,0)^T$ (desired final velocity),
- $n_{ii} = \frac{1}{3}$ (diagonal matrix),
- $b_{ii} = 1$ (identity matrix).

The control problems (4.1) and (4.6) were solved on an Intel iPSC/860, using 8 nodes simultaneously to evaluate 216 function evaluations per processor for each

TABLE 1
Problem 1. *Simulation.*

Multiplier scheme	Fixed point iterations (FIP)	Average: (FIP) per time step	Control present on RHS?
Not regularized	35633	89.0825	no
Not regularized	16676	41.6900	yes
Regularized	5135	12.8375	no
Regularized	4184	10.4600	yes

TABLE 2
Problem 1. *Control.*

Penalty scheme	Penalty constant	Multipliers scheme	(PDS) simplexes	No. IVPs solved	Error $E^1_\infty(v^*)$	Elapsed time (msec)
$P(v;\rho)$	Constant	Not regularized	21	36624	7.82E-8	1878366
$P(v;\rho)$	Constant	Regularized	21	36624	7.82E-8	1141314
$P(v;\rho)$	Dynamic	Not regularized	21	36624	7.79E-8	1878338
$P(v;\rho)$	Dynamic	Regularized	21	36624	7.80E-8	1141328
$\Pi(v;\rho)$	Constant	Not regularized	19	33136	2.01E-8	1707533
$\Pi(v;\rho)$	Constant	Regularized	19	33136	1.99E-8	1040543
$\Pi(v;\rho)$	Dynamic	Not regularized	19	33136	2.08E-8	1707433
$\Pi(v;\rho)$	Dynamic	Regularized	19	33136	1.94E-8	1040461

TABLE 3
Problem 2. *Simulation.*

Multiplier scheme	Fixed point iterations (FIP)	Average: (FIP) per time step	Control present on RHS?
Not regularized	20876	20.876	no
Not regularized	11517	11.517	yes
Regularized	3238	3.238	no
Regularized	3142	3.142	yes

TABLE 4
Problem 2. *Control.*

Penalty scheme	Penalty constant	Multipliers scheme	(PDS) simplexes	No. IVPs solved	Error $E^1_\infty(v^*)$	Elapsed time (msec)
$P(v,\rho)$	Constant	Not regularized	23	40112	2.54E-9	2309235
$P(v,\rho)$	Constant	Regularized	23	40112	2.54E-9	1424193
$P(v,\rho)$	Dynamic	Not regularized	23	40112	2.56E-9	2309188
$P(v,\rho)$	Dynamic	Regularized	23	40112	2.56E-9	1424085
$\Pi(v,\zeta)$	Constant	Not regularized	20	34880	6.03E-8	2010565
$\Pi(v,\zeta)$	Constant	Regularized	20	34880	6.02E-8	1241696
$\Pi(v,\zeta)$	Dynamic	Not regularized	20	34880	6.11E-8	2010213
$\Pi(v,\zeta)$	Dynamic	Regularized	20	34880	6.11E-8	1241428

TABLE 5
Difference between regularization and no reglarization.

Problem	Control present on RHS?	Position discrepancy $D(x_R, x_{NR})$	Velocity discrepancy $D(\dot{x}_R, \dot{x}_{NR})$
Test Problem 1	yes	4.4E-8	2.8E-5
Test Problem 1	no	1.4E-7	3.8E-4
Test Problem 2	yes	6.1E-7	5.7E-2
Test Problem 2	no	8.9E-8	3.2E-3

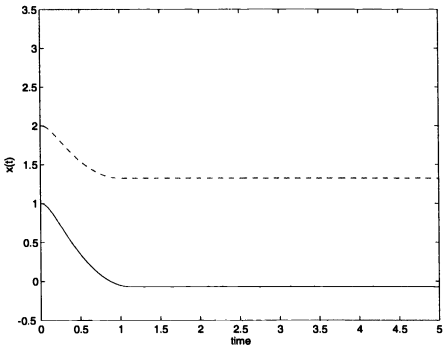


FIG. 1. (TP1) *Displacements of uncontrolled system.*

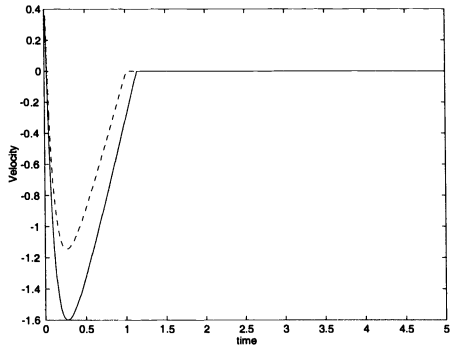


FIG. 2. (TP1) *Velocities of uncontrolled system.*

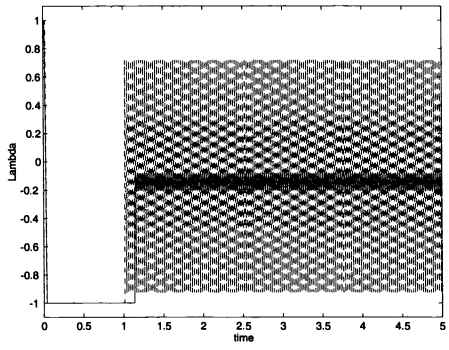


FIG. 3. (TP1) *Multipliers of uncontrolled system* (NR).

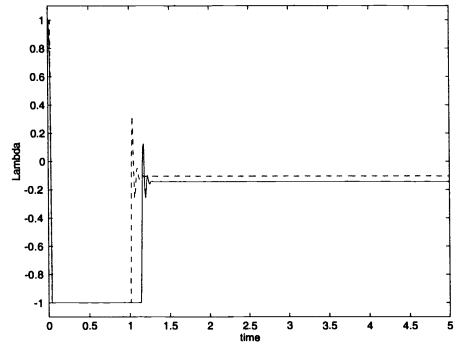


FIG. 4. (TP1) *Multipliers of uncontrolled system* (R).

simplex generated. The minimization procedure was halted in the event that the largest edge of a generated simplex was less than 10^{-10}.

The simulation was performed on a SUN SPARCstation (1+), in double precision arithmetic (IEEE 64-bit floating-point arithmetic). The fixed-point iterations were applied until (using stopping criteria (2.15)) convergence was detected (with a value of $\eta = 1.5 \times 10^{-8} \approx \sqrt{\epsilon_{\text{machine}}}$ and the norm used was $\|\cdot\| = \|\cdot\|_2$).

Tables 1–4 illustrate how effective the regularization procedure was. In the simulation, the number of fixed-point iterations decreases drastically when the regularization is employed (see Tables 2–4). While the regularization could not decrease the number of simplexes generated by (PDS) and hence, could not decrease the number of initial value problems (3.1)–(3.3) that needed to be solved, it did result in a decrease in the elapsed time. The dynamical updating of the penalty parameter had little influence on the performance of (PDS). Employing the more sophisticated penalty function (4.6) did consistently result in fewer (PDS) simplexes. The relative error in the final position, $E^1_\infty(v)$,

$$E^1_\infty(v) = \frac{\|x(T) - x_T\|_\infty}{\max\{1, \|x_T\|_\infty\}} \tag{5.1}$$

was recorded in the numerical results. The relative error associated with the velocities,

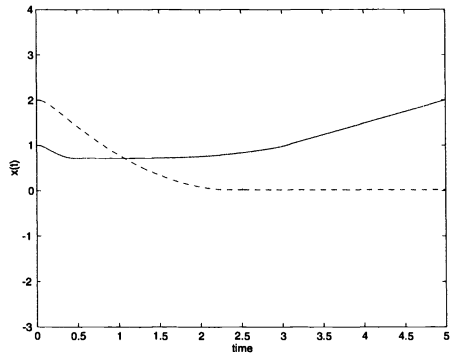


FIG. 5. (TP1) *Displacements of controlled system.*

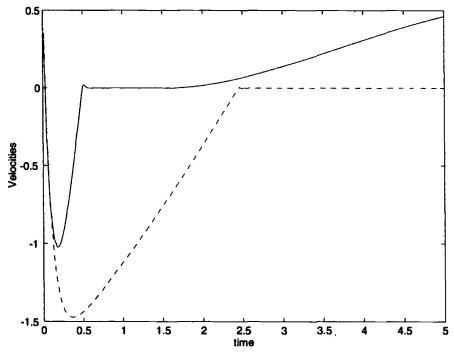


FIG. 6. (TP1) *Velocities of controlled system.*

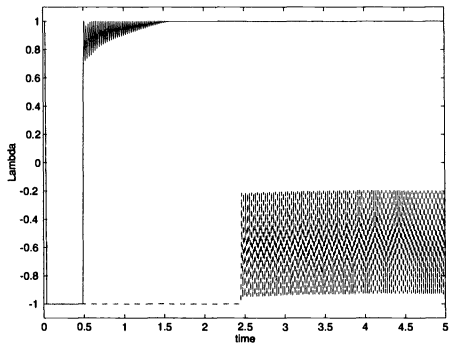


FIG. 7. (TP1) *Multipliers of controlled system* (NR).

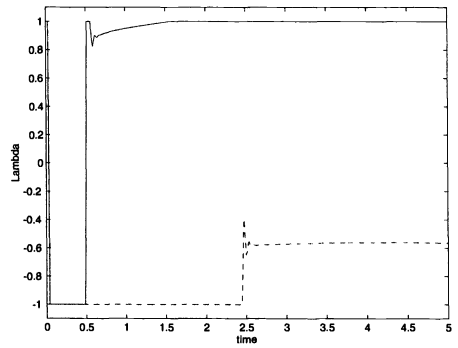


FIG. 8. (TP1) *Multipliers of controlled system* (R).

$E^2_\infty(v)$,

$$E^2_\infty(v) = \frac{\|\dot{x}(T) - \dot{x}_T\|_\infty}{\max\{1, \|\dot{x}_T\|_\infty\}} \tag{5.2}$$

was not reported in the tables. For the first test problem (TP1) no final velocities were prescribed, and for the second problem the desired velocities ($\dot{x}_T = (0,0,0)^T$) were satisfied exactly.

The effect of regularizing on the positions and velocities of our test problems is summarized in Table 5. The maximum relative difference between regularized and unregularized position and velocity is recorded. More precisely, the discrepancies

$$D(x_R, x_{NR}) = \frac{\|\max_n |x_{NR}(n\Delta t) - x_R(n\Delta t)|\|}{1 + \max_n |x_R(n\Delta t)|} \quad \text{and}$$

$$D(\dot{x}_R, \dot{x}_{NR}) = \frac{\|\max_n |\dot{x}_{NR}(n\Delta t) - \dot{x}_R(n\Delta t)|\|}{1 + \max_n |\dot{x}_R(n\Delta t)|}$$

are in the third and fourth columns, respectively, of Table 5. The subscripts R and NR denote regularized and nonregularized values respectively.

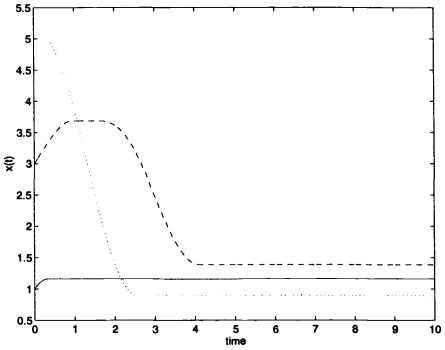


FIG. 9. (TP2) *Displacements of uncontrolled system.*

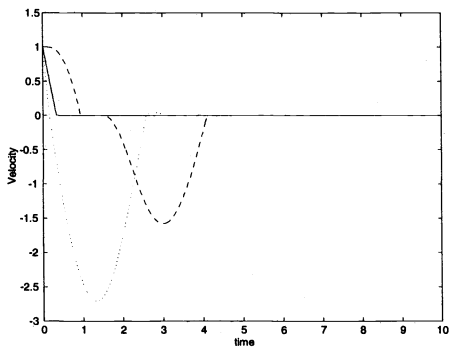


FIG. 10. (TP2) *Velocities of uncontrolled system.*

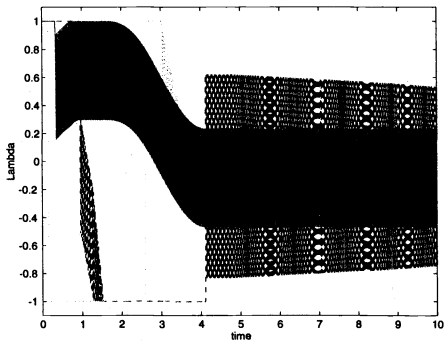


FIG. 11. (TP2) *Multipliers of uncontrolled system* (NR).

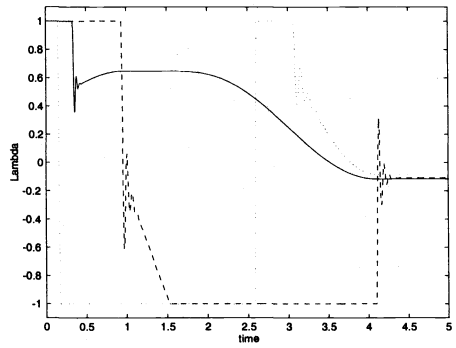


FIG. 12. (TP2) *Multipliers of uncontrolled system* (R).

Position, velocity, and the Coulomb multipliers are plotted versus time. The first set of figures (Figs. 1–4 and Figs. 9–12) represents the system coming to rest without any control present and hence the velocities of all three components vanish quite rapidly. In the second set of figures (Figs. 5–8 and Figs. 13–16) the control is activated, and the final positions of the components are the desired targets.

A significant decrease in computation was seen when the regularization (using (2.19) instead of (2.14)) was employed. The oscillating behavior of the multipliers was virtually extinguished by the regularization procedure both in the controlled system and the uncontrolled system. An interesting observation is that the values of the regularized multipliers were always close to the time averaged values of the unregularized multipliers, as observed in [DeaGKN92]. The external forces were chosen to change the sign of the velocity in the interval $t \in (0, T]$. The only nonsmooth behavior in the multipliers occurred at the jumps, where the velocity of the system changed sign. Since we are most interested in behavior of these systems when velocities become small or vanish, dealing with the nonsmooth behavior of the multipliers was the primary difficulty in these simulations.

The graphs associated with test problem one and test problem two are labeled (TP1) and (TP2), respectively. Similarly, the graphs of the Coulomb multipliers calculated with the regularization procedure are labeled with (R) and those graphs calculated without regularization are labeled by (NR). In the uncontrolled system, the

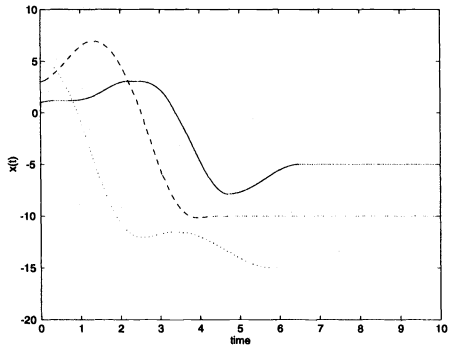


FIG. 13. (TP2) *Displacements of controlled system.*

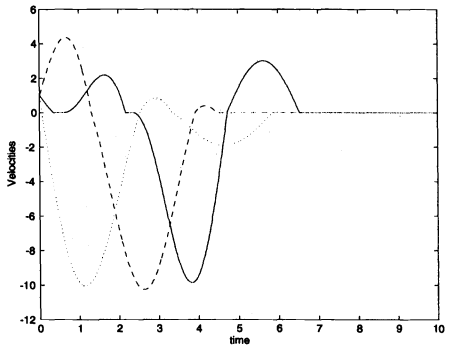


FIG. 14. (TP2) *Velocities of controlled system.*

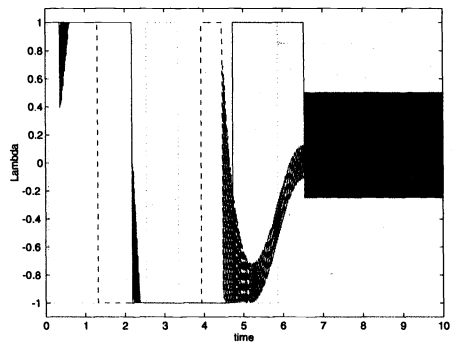


FIG. 15. (TP2) *Multipliers of controlled system* (NR).

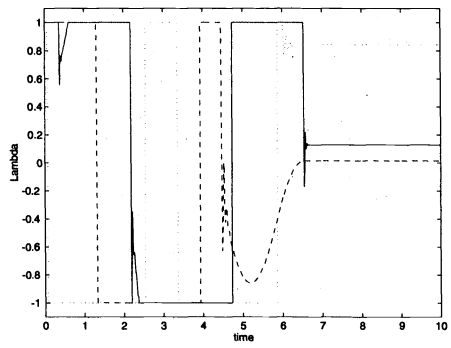


FIG. 16. (TP2) *Multipliers of controlled system* (R).

friction damped the displacements to zero very rapidly. Here the weak convergence of the multipliers is evident, and the benefit of the regularization is clear. In the controlled systems the displacements did reach the correct targets, both with and without regularization as seen in Figs. 1, 5, 9, and 13. Again, we see the weak convergence of the multiplier corresponding to the one component whose velocity vanishes for t large enough.

The smooth behavior of the multipliers when velocities are far from zero and the effectiveness of the regularization suggest that a combination of smooth and nonsmooth minimizations may prove useful in the study of these motions. In particular, using smooth or higher order minimization techniques (like those in [DenS83]) for (4.1) or (4.6) when velocities are far from zero seems appropriate. When velocities are small, more robust methods like [DenT91] may yield minima of a nonsmooth function that describes the system more accurately than a smooth analog. Perhaps the development of an algorithm for the minimization of (4.1) or (4.6) that could harness the strengths of both types of minimization techniques is possible.

Acknowledgments. The authors would like to thank the Texas Board of Higher Education and the Patricia R. Harris Fellowship at Rice University for very generous support. We must also thank the Center for Research on Parallel Computation for the use of the Intel iPSC/860. This work was facilitated by numerous helpful discussions with Professor E. J. Dean, Professor J. E. Dennis, Dr. Y. M. Kuo, Dr. G. Nasser,

Professor R. A. Tapia, and Dr. V. J. Torczon. We wish to thank Andrea Reiff for helpful suggestions regarding efficient use of (PDS). Finally we must also thank Professor C. T. Kelley and two very thorough anonymous referees whose reports contributed greatly to the presentation and substance of this paper.

REFERENCES

[Cab81] H. CABANNES, *Study of the motions of a vibrating string subject to solid friction,* Math. Meth. Appl. Sci., 3 (1981), pp. 287–300.

[CamOK82] L. T. CAMPOS, J. T. ODEN, AND N. KIKUCHI, *A numerical analysis of a class of contact problems with friction in elastodynamics,* Comput. Meth. Appl. Mech. Engrg., 34 (1982), pp. 821–845.

[Cea71] J. CÉA, *Optimisation: Théorie et Algorithmes,* Dunod, Paris, 1971.

[DeaGKN90] E. J. DEAN, R. GLOWINSKI, Y. M. KUO, AND G. NASSER, *On the discretization of some second order in time differential equations. Applications to nonlinear wave problems,* in Computational techniques in identification and control of flexible flight structures, A.V. Balakrishnan, ed., Optimization Sofware Inc., 1990, pp. 199–246.

[DeaGKN92] E. J. DEAN, R. GLOWINSKI, Y. M. KUO, AND G. NASSER, *Multiplier techniques for some dynamical systems with dry friction,* C. R. Acad. Sci. (Paris), 314 (1992), pp. 153–159.

[DenS83] J. E. DENNIS AND R. B. SCHNABEL, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations,* Prentice-Hall, Englewood Cliffs, NJ, 1983.

[DenT91] J. E. DENNIS AND V. TORCZON, *Direct search methods on parallel machines,* SIAM J. Optim., 1 (1991), pp. 448–474.

[DuvL76] G. DUVAUT AND J. L. LIONS, *Inequalities in Mechanics and Physics,* Springer-Verlag, Berlin, 1976.

[GilMW81] P. E. GILL, W. MURRAY, AND M. H. WRIGHT, *Practical Optimization.* Academic Press, New York, 1981.

[Glo84] R. GLOWINSKI, *Numerical Methods for Nonlinear Variational Problems,* Springer-Verlag, New York, 1984.

[GloK95] R. GLOWINSKI AND A. J. KEARSLEY, *Numerical control of a vibrating string subject to dry friction,* manuscript.

[KeaTT93] A. J. KEARSLEY, R. A. TAPIA, AND V. TORCZON, *On the use of parallel direct search methods for nonlinear programming problems,* Tech. Report 93-33, Dept. of Computational and Applied Mathematics, Rice University, Houston TX, 1993.

[KikO88] N. KIKUCHI AND J. T. ODEN, *Contact Problems in Elasticity: A Study of Variational Inequalities and Finite Element Methods,* Society for Industrial and Applied Mathematics, Philadelphia, 1988.

[Lio71] J. L. LIONS, *Optimal Control of Systems Governed by Partial Differential Equations,* Springer-Verlag. Berlin, 1971.

[Lio88] ———, *Exact controllability, stabilization and perturbations for distributed systems,* SIAM Rev., 30 (1988), pp. 1–68.

[Ren92] M. RENARDY, *Ill-posedness at the boundary for elastic solids sliding under Coulomb friction,* J. Elasticity, 27 (1992), pp. 281–287.

[Tor92] V. TORCZON, PDS: *Direct search methods for unconstrained optimization on either sequential or parallel machines,* Tech. Report 92-9, Dept. of Computational and Applied Mathematics, Rice University, Houston TX, 1993. ACM Transactions on Mathematical Software, submitted.

[Tor93] V. TORCZON, *On the convergence of pattern search algorithms,* Tech. Report 93-10, Department of Computational and Applied Mathematics, Rice University, Houston TX, 1993, SIAM J. Optim, submitted.

SIAM J. OPTIMIZATION
Vol. 5, No. 4, pp. 695–739, November 1995

SOME CONVERGENCE PROPERTIES OF THE MODIFIED LOG BARRIER METHOD FOR LINEAR PROGRAMMING*

M.J.D. POWELL†

Abstract. On each iteration of the modified log barrier method for linear programming, a vector of variables is calculated by the unconstrained minimization of a fixed positive multiple of the objective function minus a linear combination of the logarithms of the constraint residuals plus positive constants. Thus, in contrast to interior point methods, all the logarithms are finite whenever the vector of variables is feasible. The parameters of the calculation have the property that the coefficients of the linear combination of logarithms can be viewed as estimates of Lagrange multipliers of Karush–Kuhn–Tucker conditions of the given problem. Initially these coefficients have any positive values, and then their adjustment is derived from the zero gradient that occurs at the end of each unconstrained minimization calculation. These techniques provide a simple and interesting algorithm for linear programming that was proposed by Polyak. We study its convergence properties, finding that the sequence of calculated variables converges to a solution. Furthermore, the values of the objective function tend to optimality and any constraint violations tend to zero at R-linear rates. These conclusions are valid even when there are many solutions. Indeed, our only assumption is that the feasible region of the linear programming problem is bounded and nonempty, which is far less restrictive than the assumptions that are usually made in theoretical investigations of the modified log barrier method.

Key words. convergence theory, linear programming, modified log barrier method

AMS subject classifications. 65K05, 90C05

1. Introduction.

Log barrier methods for linear programming are often highly efficient in practice, being used, for example, in the very successful software that has been developed by Lustig, Marsten, and Shanno (1991). Furthermore, it is known that they enjoy polynomial time convergence properties, which are reviewed well by Gonzaga (1991). In order to describe the main idea of these methods, we consider the minimization of the linear function

$$\underline{c}^T\underline{x}, \quad \underline{x}\in\mathcal{R}^n, \tag{1.1}$$

subject to the linear inequality constraints

$$\underline{a}_k^T\underline{x} \geq b_k, \quad k=1,2,\ldots,m, \tag{1.2}$$

where the components of the vectors $\underline{c}$ and $\{\underline{a}_k : k=1,2,\ldots,m\}$ and the numbers $\{b_k : k=1,2,\ldots,m\}$ are data. It is required that the feasible region, $\mathcal{S}_0$ say, is bounded and has a nonempty interior. Then, for every value of the parameter σ, the function

$$\Phi_\sigma(\underline{x}) = \sigma\,\underline{c}^T\underline{x} - \sum_{k=1}^{m}\log(\underline{a}_k^T\underline{x}-b_k), \quad \underline{x}\in\mathcal{S}_0, \tag{1.3}$$

is strictly convex and has a unique minimizer, $\underline{x}(\sigma)$ say, which is an interior point of $\mathcal{S}_0$. A log barrier method follows approximately the trajectory $\{\underline{x}(\sigma) : 0<\sigma<\infty\}$ in $\mathcal{S}_0$ by calculating an estimate of $\underline{x}(\sigma)$ for a sequence of values of σ that diverges to infinity, because this trajectory leads to a solution of the linear programming problem.

* Received by the editors November 25, 1992; accepted for publication (in revised form) May 2, 1994.

† Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Silver Street, Cambridge CB3 9EW, England (mjdp@amtp.cam.ac.uk).

An important ingredient of an efficient implementation is to ensure that the increases in σ have the property that, when calculating $\underline{x}(\sigma)$ for a new value of σ, it is suitable to apply an iterative procedure whose initial vector of variables is the $\underline{x}(\sigma)$ that was found for the previous value of σ. It would be very wasteful, however, to make changes to σ that are far smaller than necessary, because usually a large final value of σ is required in order that the final $\underline{x}(\sigma)$ is acceptably close to a solution of the given calculation. Another important practical consideration is that, if one takes a global view of the function (1.3) when σ is very large, then the graph of Φ_σ has sharp corners near the boundary of $\mathcal{S}_0$, which can be particularly severe near the vertices of the feasible region. These corners introduce several difficulties into the minimization of Φ_σ, such as ill-conditioning of the second derivative matrix $\nabla^2\Phi_\sigma$ at $\underline{x}(\sigma)$.

These features of log barrier methods were studied more than 20 years ago for general inequality constraints (Fiacco and McCormick (1968)). Then, in the 1970's, the augmented Lagrangian method and its extensions tended to be preferred, because the use of estimates of Lagrange multipliers can avoid the need for a parameter that must diverge to infinity. These developments are described in several books, such as Fletcher (1987) and Gill, Murray, and Wright (1981). In the augmented Lagrangian method, however, one loses a strong advantage of the function (1.3), which is that the log terms provide barriers that prevent constraint violations.

Therefore Polyak (1992) has developed a way of combining the merits of the augmented Lagrangian approach with terms that restrict constraint violations, namely, the "modified log barrier method." In order to avoid a construction that is analogous to σ tending to infinity, it is necessary for the modified barriers to be outside the boundary of the original feasible region. Specifically, instead of working with expression (1.3), one applies an algorithm for unconstrained minimization to the function

$$\Phi^{(\ell)}(\underline{x}) = \sigma\,\underline{c}^T\underline{x} - \sum_{k=1}^{m} \lambda_k^{(\ell)} \log[\,\sigma(\underline{a}_k^T\underline{x} - b_k) + 1\,], \quad \underline{x} \in \mathcal{S}_1, \tag{1.4}$$

where the new parameters $\{\lambda_k^{(\ell)} : k = 1, 2, \ldots, m\}$ are all positive, and where $\mathcal{S}_1$ is the set of vectors $\underline{x}$ in $\mathcal{R}^n$ such that the numbers $\{\sigma(\underline{a}_k^T\underline{x} - b_k) + 1 : k = 1, 2, \ldots, m\}$ are all nonnegative. We see that the log terms of expression (1.4) allow the original constraints (1.2) to be violated by at most σ^{-1}. Therefore, if σ is a moderate positive number, any solution of the given linear programming problem is well away from the boundary of $\mathcal{S}_1$. Let $\underline{x}^{(\ell)}$ be the minimizer of $\Phi^{(\ell)}$. It will be shown that, for any fixed positive value of σ, there exists a sequence of parameter vectors $\{\underline{\lambda}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ in $\mathcal{R}_+^m$ such that $\underline{x}^{(\ell)}$ converges to a solution of the given calculation as $\ell \to \infty$. Here the subscript $+$ on $\mathcal{R}^m$ indicates that all the components of each $\underline{\lambda}^{(\ell)}$ are positive. We note also that the function

$$\sigma\,\underline{c}^T\underline{x} - \sum_{k=1}^{m} \lambda_k^{(\ell)} \log[\,\underline{a}_k^T\underline{x} - b_k + \sigma^{-1}\,], \quad \underline{x} \in \mathcal{S}_1, \tag{1.5}$$

differs from $\Phi^{(\ell)}$ by a term that is independent of $\underline{x}$, because it follows that the minimization of $\Phi^{(\ell)}$ is comparable to the minimization of Φ_σ if σ is large and if all the components of $\underline{\lambda}^{(\ell)}$ are one.

The definition of $\underline{x}^{(\ell)}$ implies that the gradient vector $\nabla\Phi^{(\ell)}(\underline{x}^{(\ell)})$ is zero, so we have the equation

$$\underline{c} = \sum_{k=1}^{m} \lambda_k^{(\ell)} \frac{\underline{a}_k}{\sigma(\underline{a}_k^T\underline{x}^{(\ell)} - b_k) + 1}. \tag{1.6}$$

Moreover, if $\underline{x}^{(*)}$ is any solution of the given linear programming problem, then the well-known KKT (Karush–Kuhn–Tucker) conditions for optimality provide the identity

$$\underline{c} = \sum_{k=1}^{m} \lambda_k^{(*)} \underline{a}_k, \tag{1.7}$$

where $\lambda_k^{(*)}$ is zero if $\underline{a}_k^T \underline{x}^{(*)} - b_k$ is positive and where the remaining components of $\underline{\lambda}^{(*)} \in \mathcal{R}^m$ are nonnegative. We see that the condition (1.6) reduces to (1.7) if $\underline{\lambda}^{(\ell)} = \underline{\lambda}^{(*)}$ and if $\underline{x}^{(\ell)}$ satisfies the condition $\underline{a}_k^T \underline{x}^{(\ell)} = b_k$ for all values of k such that $\lambda_k^{(*)}$ is nonzero. Therefore it is appropriate to regard $\underline{\lambda}^{(\ell)}$ as an approximation to a vector of Lagrange multipliers. Alternatively, one can regard $\underline{\lambda}^{(\ell)}$ as a vector of variables of the dual linear programming problem, but our view may be more helpful to any extensions of the given theory to nonlinear constraints that are not necessarily convex.

If σ is much larger than the components of $\underline{\lambda}^{(\ell)} \in \mathcal{R}_+^m$, then usually $\underline{x}^{(\ell)}$ is close to a solution of the linear programming problem, because it is the minimizer of the function (1.5). It follows from a comparison of expressions (1.6) and (1.7) that the "updating formula"

$$\lambda_k^{(\ell+1)} = \lambda_k^{(\ell)} / [\sigma(\underline{a}_k^T \underline{x}^{(\ell)} - b_k) + 1], \quad k = 1, 2, \ldots, m, \tag{1.8}$$

is often an excellent way of generating $\underline{\lambda}^{(\ell+1)}$ from $\underline{\lambda}^{(\ell)}$ in practice. Furthermore, $\underline{\lambda}^{(\ell+1)}$ inherits positive components from $\underline{\lambda}^{(\ell)}$ because $\underline{x}^{(\ell)}$ is an interior point of $\mathcal{S}_1$. The use of this formula is recommended by Polyak (1992). He proves that, if it is applied recursively for a suitable fixed value of σ, then the points $\{\underline{x}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ and the sequence $\{\underline{\lambda}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ converge to a solution of the given optimization problem and an optimal vector of Lagrange multipliers, respectively, assuming that the following conditions are satisfied. 1. The linear programming problem has a unique solution $\underline{x}^{(*)}$; 2. the feasible region $\mathcal{S}_0$ is bounded and has a nonempty interior; 3. the constraints (1.2) have the nondegeneracy property that exactly n of them are satisfied as equations at $\underline{x} = \underline{x}^{(*)}$; 4. the "strict complementarity condition" holds, which means that $\lambda_k^{(*)}$ is positive in (1.7) if $\underline{a}_k^T \underline{x}^{(*)} - b_k$ is zero; and 5. the fixed parameter σ is sufficiently large. We will find that most of these conditions are unnecessary, but it should be mentioned that Polyak (1992) addresses nonlinear objective and constraint functions, so some of his results for linear programming applications are corollaries of theorems that do not take advantage of linearity, and also he studies rates of convergence.

Therefore we consider the following procedure for minimizing the linear function (1.1) subject to the constraints (1.2).

THE ALGORITHM. *Initially the components of* $\underline{\lambda}^{(1)} \in \mathcal{R}^m$ *and the parameter* σ *have any positive values. Then an infinite sequence of iterations is begun,* ℓ *being the index of the iteration, so* $\ell = 1$ *is set initially. For each* ℓ*, the vector* $\underline{x}^{(\ell)}$ *is calculated by minimizing the function* (1.4). *Then* $\underline{\lambda}^{(\ell+1)}$ *is defined by the updating formula* (1.8), *which completes the* ℓ*th iteration.*

We are going to investigate the properties of this algorithm assuming only that $\mathcal{S}_0$ is bounded and nonempty. Hence in §2 we deduce that the calculations are well defined for all positive integers ℓ. It is important to study the convergence of the sequences $\{\underline{x}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ and $\{\underline{\lambda}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$, because the modified log barrier

method may provide substantial improvements to algorithms for linear programming, by gaining some of the advantages that augmented Lagrangian methods have over the original use of log barrier functions.

Let us compare our assumption that $\mathcal{S}_0$ is bounded and nonempty with the five conditions of Polyak's analysis that have been noted. It is highly useful to remove his conditions 1, 3, and 4, because degenerate calculations do occur in practice, and it is important that they should not cause failure in linear programming software that is intended for general applications. Deleting the nonempty interior assertion of condition 2 is also valuable, because sometimes expressing an equality constraint as two inequalities is more convenient than satisfying the equality by eliminating a variable. Furthermore, our absence of restrictions on $\underline{\lambda}^{(1)}$ and σ, except for positivity, may assist the choice of suitable initial values of these parameters. It should be noted, however, that the rate of convergence of the iterative procedure is usually very slow when σ is small. Nevertheless, the fact that one can prove convergence under such weak assumptions is likely to be of interest to many theoreticians.

The main purpose of §2 is to establish a consequence of the updating formula (1.8) that provides the backbone of our analysis. It is that the sequence

$$\phi^{(\ell)} = \Phi^{(\ell)}(\underline{x}^{(\ell)}), \quad \ell = 1, 2, 3, \ldots, \tag{1.9}$$

increases monotonically and is bounded above by $\sigma\, \underline{c}^T \underline{x}^{(*)}$, where $\underline{x}^{(*)}$ is any solution of the linear programming problem. This result is analogous to a duality property of the augmented Lagrangian method.

Let $\mathcal{K}^{(*)}$ be the set of indices of the inequality constraints that are satisfied as equations at every solution of the linear programming problem. It is proved in §3 that, for each k in $\mathcal{K}^{(*)}$, the sequence $\{\underline{a}_k^T \underline{x}^{(\ell)} - b_k : \ell = 1, 2, 3, \ldots\}$ tends to zero as $\ell \to \infty$. It follows that, if the given calculation has a unique solution $\underline{x}^{(*)}$, then the points $\{\underline{x}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ must converge to $\underline{x}^{(*)}$ as required, because uniqueness implies that the vectors $\{\underline{a}_k : k \in \mathcal{K}^{(*)}\}$ span $\mathcal{R}^n$. This analysis also establishes the convergence of the sequence $\{\underline{\lambda}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ of Lagrange multiplier estimates, the limiting value of $\lambda_k^{(\ell)}$ being positive if and only if k is in $\mathcal{K}^{(*)}$.

On the other hand, when the linear programming problem has several solutions, then the conditions

$$\lim_{\ell \to \infty} \underline{a}_k^T \underline{x}^{(\ell)} = b_k, \quad k \in \mathcal{K}^{(*)}, \tag{1.10}$$

are not sufficient to imply that the sequence $\{\underline{x}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ converges. In this case, therefore, the work of §4 shows that the residuals $\{\underline{a}_k^T \underline{x}^{(\ell)} - b_k : \ell = 1, 2, 3, \ldots\}$ tend to a positive limit for at least one value of k that is not in $\mathcal{K}^{(*)}$. Furthermore, if this addition to the properties (1.10) does not establish a limit of the points $\{\underline{x}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$, then the argument of §4 can be applied recursively to find yet more values of k such that the numbers $\{\underline{a}_k^T \underline{x}^{(\ell)} - b_k : \ell = 1, 2, 3, \ldots\}$ converge. Thus we conclude eventually that the calculated vectors $\{\underline{x}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ tend to a feasible point, which is a solution of the given linear programming problem due to some of the theory of §§2 and 3.

In the analysis of §4, there is a set $\mathcal{K} \supset \mathcal{K}^{(*)}$ of constraint indices, such that, for each k in $\mathcal{K}$, it is known that the residuals $\{\underline{a}_k^T \underline{x}^{(\ell)} - b_k : \ell = 1, 2, 3, \ldots\}$ tend to a limit, but the vectors $\{\underline{a}_k : k \in \mathcal{K}\}$ do not span $\mathcal{R}^n$. Therefore it is helpful to take the following view of the minimization of the function (1.4) for each ℓ. We express $\underline{x}^{(\ell)}$ in

the form

$$\underline{x}^{(\ell)} = \underline{y}^{(\ell)} + \underline{z}^{(\ell)}, \tag{1.11}$$

where $\underline{y}^{(\ell)}$ is in the linear space, $\mathcal{Y}$ say, that is spanned by $\{\underline{a}_k : k \in \mathcal{K}\}$, and where $\underline{z}^{(\ell)}$ is in the orthogonal complement of $\mathcal{Y}$. Furthermore, we suppose that $\underline{y}^{(\ell)}$ is known before $\underline{z}^{(\ell)}$ is calculated. Thus $\underline{z}^{(\ell)}$ is defined by an unconstrained minimization calculation that depends on σ, $\underline{\lambda}^{(\ell)}$, and $\underline{y}^{(\ell)}$. Now $\underline{y}^{(\ell)}$ is a perturbation of a known vector, where the perturbation tends to zero as $\ell \to \infty$. Furthermore, except for the consequences of the perturbations, the properties of the sequence $\{\underline{z}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ are analogous to the properties of $\{\underline{x}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ that are found in §3. Therefore much of the analysis of §4 is an extension of the theory of §3 that accommodates the perturbations. We protect the reader from many of the complications of the extension by placing the proofs of some of the lemmas of §4 in an Appendix.

We draw conclusions from our theory in §5, noting that $\{\underline{x}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ tends to a limit, $\hat{\underline{x}}^{(*)}$ say, that is independent of the initial parameters σ and $\underline{\lambda}^{(1)}$. Thus we confirm an observation of Polyak (private communication) that the rate of convergence of this sequence to $\hat{\underline{x}}^{(*)}$ can be controlled by the value of σ.

It has been anticipated that most readers will not have time to consider the arguments that address the difficult cases when the linear programming problem has more than one solution. Therefore it is advisable to skip both §4 and the Appendix initially. Thus one can gain an understanding of the method of analysis that makes it easier to study the treatment of degeneracies. Furthermore, the presentation allows one to read §4 completely before referring to the Appendix.

2. Some properties of the algorithm. The first result of this section is that the vector of variables $\underline{x}^{(\ell)}$ that minimizes the function (1.4) is well defined on each iteration of the algorithm. Second, we consider the relevance of the numbers (1.9), noting that they are bounded above by σ times the optimal value of the objective function, and that they tend to this bound as $\ell \to \infty$ if the sequences $\{\underline{x}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ and $\{\underline{\lambda}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ converge to optimal vectors of variables and Lagrange multipliers, respectively. Third, we find that the algorithm provides the strict inequality

$$\phi^{(\ell+1)} > \phi^{(\ell)} \tag{2.1}$$

on every iteration, except in the highly degenerate case when $\underline{x}^{(\ell)}$ is on the boundaries of *all* the constraints (1.2). Thus we deduce that, if the calculated sequences $\{\underline{x}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ and $\{\underline{\lambda}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ converge, then their limits satisfy the KKT conditions for optimality. The section ends by mentioning some of the difficulties that must be overcome by the analysis of §§3 and 4.

LEMMA 2.1. *The assumption that the set $\mathcal{S}_0$ is bounded and nonempty implies that $\mathcal{S}_1$ is bounded and has a nonempty interior, these sets being defined in* §1. *Furthermore, for any positive values of σ and $\{\lambda_k^{(\ell)} : k = 1, 2, \ldots, m\}$, the function* (1.4) *is strictly convex and has a unique minimizer $\underline{x}^{(\ell)}$, which is an interior point of $\mathcal{S}_1$.*

Proof. The set $\mathcal{S}_1 \subset \mathcal{R}^n$ is convex. Therefore, if it is unbounded, there exists a nonzero vector $\underline{v}$ such that $\underline{x} + \alpha \underline{v}$ is in $\mathcal{S}_1$ for every $\underline{x}$ in $\mathcal{S}_1$ and for every positive multiplier α. It follows from the definition of $\mathcal{S}_1$ that $\underline{v}$ satisfies the conditions

$$\underline{a}_k^T \underline{v} \geq 0, \quad k = 1, 2, \ldots, m. \tag{2.2}$$

Let $\underline{x}$ be in the nonempty subset $\mathcal{S}_0$ of $\mathcal{S}_1$. Then the inequalities (2.2) imply that $\underline{x}+\alpha\underline{v}$ is in $\mathcal{S}_0$ for every positive α. This contradiction of the boundedness of $\mathcal{S}_0$ implies that $\mathcal{S}_1$ is bounded too. Furthermore, the set $\mathcal{S}_0$ provides a nonempty interior of $\mathcal{S}_1$.

The function (1.4) is strictly convex if its second derivative matrix

$$\nabla^2\Phi^{(\ell)}(\underline{x}) = \sum_{k=1}^{m} \lambda_k^{(\ell)} \frac{\sigma^2 \underline{a}_k \underline{a}_k^T}{[\sigma(\underline{a}_k^T \underline{x} - b_k)+1]^2} \tag{2.3}$$

is positive definite at all interior points of $\mathcal{S}_1$. Because the components of $\underline{\lambda}^{(\ell)}$ are all positive, we see that this matrix has no negative eigenvalues and that it is singular if and only if a nonzero vector $\underline{v}$ satisfies the equations

$$\underline{a}_k^T \underline{v} = 0, \quad k=1,2,\ldots,m. \tag{2.4}$$

We found in the previous paragraph, however, that these equations lead to a contradiction, which establishes the required strict convexity of $\Phi^{(\ell)}$. Furthermore, since $\Phi^{(\ell)}$ is continuous on the interior of $\mathcal{S}_1$ and tends to infinity at the boundary of $\mathcal{S}_1$, it has a minimizer, which is unique due to the strict convexity. Therefore $\underline{x}^{(\ell)}$ is a well-defined interior point of $\mathcal{S}_1$. □

It follows from Lemma 2.1 and the updating formula (1.8) that, if the components of $\underline{\lambda}^{(\ell)}$ are positive, then not only is $\underline{x}^{(\ell)}$ well defined, but also the components of $\underline{\lambda}^{(\ell+1)}$ are positive. Thus the calculations of the algorithm we are studying can continue for an infinite number of iterations.

LEMMA 2.2. *Let $\underline{x}^{(*)}$ be any solution of the given linear programming problem. Then the numbers* (1.9) *satisfy the condition*

$$\phi^{(\ell)} \leq \sigma\, \underline{c}^T \underline{x}^{(*)}, \quad \ell=1,2,3,\ldots . \tag{2.5}$$

Furthermore, if $\{\underline{x}^{(\ell)} : \ell=1,2,3,\ldots\}$ converges to $\underline{x}^{(*)}$ and if $\{\underline{\lambda}^{(\ell)} : \ell=1,2,3,\ldots\}$ converges to a Lagrange multiplier vector, $\underline{\lambda}^{(*)}$ say, of the KKT *conditions at $\underline{x}^{(*)}$, then the sequence* (1.9) *has the property*

$$\lim_{\ell\to\infty} \phi^{(\ell)} = \sigma\, \underline{c}^T \underline{x}^{(*)}. \tag{2.6}$$

Proof. It follows from the definitions of $\underline{x}^{(\ell)}$ and $\Phi^{(\ell)}$ that we have the bound

$$\begin{aligned} \phi^{(\ell)} \leq \Phi^{(\ell)}(\underline{x}^{(*)}) &= \sigma\, \underline{c}^T \underline{x}^{(*)} - \sum_{k=1}^{m} \lambda_k^{(\ell)} \log[\sigma(\underline{a}_k^T \underline{x}^{(*)} - b_k)+1] \\ &\leq \sigma\, \underline{c}^T \underline{x}^{(*)}, \qquad \ell=1,2,3,\ldots, \end{aligned} \tag{2.7}$$

where the last line depends on the constraints (1.2) on $\underline{x}^{(*)}$ and on the positivity of $\underline{\lambda}^{(\ell)}$. Therefore condition (2.5) is satisfied. Moreover, because the right-hand side of expression (1.4) is a continuous function of $\underline{x}$ and $\underline{\lambda}^{(\ell)}$ in neighbourhoods of $\underline{x}^{(*)}$ and $\underline{\lambda}^{(*)}$, respectively, the assumed limits $\underline{x}^{(\ell)} \to \underline{x}^{(*)}$ and $\underline{\lambda}^{(\ell)} \to \underline{\lambda}^{(*)}$ imply the equation

$$\begin{aligned} \lim_{\ell\to\infty} \phi^{(\ell)} = \lim_{\ell\to\infty} \Phi^{(\ell)}(\underline{x}^{(\ell)}) &= \sigma\, \underline{c}^T \underline{x}^{(*)} - \sum_{k=1}^{m} \lambda_k^{(*)} \log[\sigma(\underline{a}_k^T \underline{x}^{(*)} - b_k)+1] \\ &= \sigma\, \underline{c}^T \underline{x}^{(*)}, \end{aligned} \tag{2.8}$$

where the last line depends on the fact that, due to the KKT conditions, $\lambda_k^{(*)}$ is zero if $\underline{a}_k^T \underline{x}^{(*)} - b_k$ is nonzero. Therefore the lemma is true. □

Lemma 2.2 is useful because it tells us that, if we wish to achieve the limits of the sequences $\{\underline{x}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ and $\{\underline{\lambda}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ that are assumed, then we should satisfy (2.6). Therefore, in view of inequality (2.5), we want the numbers $\{\phi^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ to converge to as large a value as possible. Fortunately, the updating formula (1.8) gives the following properties.

LEMMA 2.3. *Every iteration of the algorithm provides the inequality*

$$\phi^{(\ell+1)} - \phi^{(\ell)} \geq w_1 \sum_{k=1}^{m} \lambda_k^{(\ell+1)} [\, \sigma(\underline{a}_k^T \underline{x}^{(\ell)} - b_k) \,]^2, \tag{2.9}$$

where w_1 is a positive constant. Furthermore, $\phi^{(\ell+1)}$ also satisfies the bound

$$\sigma\, \underline{c}^T \underline{x}^{(*)} - \phi^{(\ell+1)} \leq \sum_{k=1}^{m} \lambda_k^{(\ell+1)} \sigma(\underline{a}_k^T \underline{x}^{(*)} - b_k), \tag{2.10}$$

where $\underline{x}^{(*)}$ is any solution of the linear programming problem.

Proof. Equations (1.6) and (1.8) imply the identity

$$\underline{c} = \sum_{k=1}^{m} \lambda_k^{(\ell+1)} \underline{a}_k. \tag{2.11}$$

Therefore the definitions of $\phi^{(\ell+1)}$ and $\underline{x}^{(\ell+1)}$ and an elementary property of the log function provide the relation

$$\begin{aligned} \phi^{(\ell+1)} &= \sigma\, \underline{c}^T \underline{x}^{(\ell+1)} - \sum_{k=1}^{m} \lambda_k^{(\ell+1)} \log[\, \sigma(\underline{a}_k^T \underline{x}^{(\ell+1)} - b_k) + 1 \,] \\ &\geq \sigma\, \underline{c}^T \underline{x}^{(\ell+1)} - \sum_{k=1}^{m} \lambda_k^{(\ell+1)} [\, \sigma(\underline{a}_k^T \underline{x}^{(\ell+1)} - b_k) \,] \\ &= \sigma \sum_{k=1}^{m} \lambda_k^{(\ell+1)} b_k. \end{aligned} \tag{2.12}$$

Thus, using the definitions of $\phi^{(\ell)}$ and $\underline{x}^{(\ell)}$, (2.11) again, and formula (1.8), we find the inequality

$$\begin{aligned} \phi^{(\ell+1)} - \phi^{(\ell)} &= \phi^{(\ell+1)} - \sigma\, \underline{c}^T \underline{x}^{(\ell)} + \sum_{k=1}^{m} \lambda_k^{(\ell)} \log[\, \sigma(\underline{a}_k^T \underline{x}^{(\ell)} - b_k) + 1 \,] \\ &\geq \sum_{k=1}^{m} \lambda_k^{(\ell)} \log[\, \sigma(\underline{a}_k^T \underline{x}^{(\ell)} - b_k) + 1 \,] - \sum_{k=1}^{m} \lambda_k^{(\ell+1)} \sigma(\underline{a}_k^T \underline{x}^{(\ell)} - b_k) \\ &= \sum_{k=1}^{m} \lambda_k^{(\ell+1)} \Big\{ [\, \sigma(\underline{a}_k^T \underline{x}^{(\ell)} - b_k) + 1 \,] \log[\, \sigma(\underline{a}_k^T \underline{x}^{(\ell)} - b_k) + 1 \,] - \sigma(\underline{a}_k^T \underline{x}^{(\ell)} - b_k) \Big\}. \end{aligned} \tag{2.13}$$

We write this condition in the form

$$\phi^{(\ell+1)} - \phi^{(\ell)} \geq \sum_{k=1}^{m} \lambda_k^{(\ell+1)} \psi\Big(\sigma(\underline{a}_k^T \underline{x}^{(\ell)} - b_k) \Big), \tag{2.14}$$

where ψ is the function of one variable

$$\psi(\theta) = (\theta+1)\log(\theta+1) - \theta, \quad -1<\theta\leq\bar{\theta}, \tag{2.15}$$

and where $\bar{\theta}$ is a constant upper bound on the numbers $\{\sigma(\underline{a}_k^T\underline{x} - b_k) : k = 1, 2, \ldots, m\}$ for every $\underline{x}$ in $\mathcal{S}_1$. Noting that ψ has the derivatives

$$\psi'(\theta) = \log(\theta+1) \quad \text{and} \quad \psi''(\theta) = (\theta+1)^{-1}, \tag{2.16}$$

we use a Taylor series expansion to deduce the bound

$$\psi(\theta) = \psi(0) + \theta\,\psi'(0) + \tfrac{1}{2}\theta^2\psi''(\xi) \geq \tfrac{1}{2}\theta^2(\bar{\theta}+1)^{-1}, \tag{2.17}$$

ξ being a point in the interval $[\min\{0,\theta\}, \max\{0,\theta\}]$. Therefore expression (2.14) implies the relation

$$\phi^{(\ell+1)} - \phi^{(\ell)} \geq \tfrac{1}{2}(\bar{\theta}+1)^{-1}\sum_{k=1}^{m}\lambda_k^{(\ell+1)}[\sigma(\underline{a}_k^T\underline{x}^{(\ell)} - b_k)]^2, \tag{2.18}$$

so inequality (2.9) is satisfied when w_1 is the constant $\frac{1}{2}(\bar{\theta}+1)^{-1}$. We complete the proof by deducing the property (2.10) from the fact that expressions (2.11) and (2.12) provide the inequality

$$\sigma\,\underline{c}^T\underline{x}^{(*)} - \phi^{(\ell+1)} \leq \sigma\sum_{k=1}^{m}\lambda_k^{(\ell+1)}\underline{a}_k^T\underline{x}^{(*)} - \sigma\sum_{k=1}^{m}\lambda_k^{(\ell+1)}b_k. \qquad \square \tag{2.19}$$

The second assertion of Lemma 2.3 will be useful in §3, while the first assertion establishes the strict inequality (2.1), unless all the constraint residuals $\{\underline{a}_k^T\underline{x}^{(\ell)} - b_k : k = 1, 2, \ldots, m\}$ are zero. It now follows from the bound (2.5) that the differences $\{\phi^{(\ell+1)} - \phi^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ tend to zero. Hence condition (2.9) implies that the algorithm gives the limits

$$\lim_{\ell\to\infty}\lambda_k^{(\ell+1)}(\underline{a}_k^T\underline{x}^{(\ell)} - b_k)^2 = 0, \quad k=1, 2, \ldots, m. \tag{2.20}$$

These limits and the updating formula (1.8) make us hopeful that the algorithm will converge satisfactorily, because they provide the following lemma.

LEMMA 2.4. *If the sequences* $\{\underline{x}^{(\ell)} : \ell=1, 2, 3, \ldots\}$ *and* $\{\underline{\lambda}^{(\ell)} : \ell=1, 2, 3, \ldots\}$ *are convergent, their limits being* $\hat{\underline{x}}^{(*)}$ *and* $\underline{\lambda}^{(*)}$ *say, then* $\hat{\underline{x}}^{(*)}$ *is a solution of the linear programming problem and* $\underline{\lambda}^{(*)}$ *is a Lagrange multiplier vector of the* KKT *conditions at* $\hat{\underline{x}}^{(*)}$.

Proof. For each constraint index k, formula (1.8) implies the equation

$$\lambda_k^{(\ell+1)} = \lambda_k^{(1)} \Big/ \prod_{j=1}^{\ell}[\sigma(\underline{a}_k^T\underline{x}^{(j)} - b_k) + 1]. \tag{2.21}$$

Therefore, if the limit $\underline{a}_k^T\hat{\underline{x}}^{(*)} - b_k$ of the constraint residuals $\{\underline{a}_k^T\underline{x}^{(\ell)} - b_k : \ell=1, 2, 3, \ldots\}$ were negative, then the sequence $\{\lambda_k^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ would diverge, which is a contradiction. It follows that the point $\hat{\underline{x}}^{(*)}$ is feasible. The updating formula also provides the relation (2.11), so (1.7) holds. Moreover, we deduce from expression (2.20) that $\underline{a}_k^T\hat{\underline{x}}^{(*)} - b_k$ is zero if $\lambda_k^{(*)}$ is positive. Moreover, because each $\underline{\lambda}^{(\ell)}$ is in $\mathcal{R}_+^m$,

none of the components of $\underline{\lambda}^{(*)}$ is negative. These remarks show that $\underline{\hat{x}}^{(*)}$ and $\underline{\lambda}^{(*)}$ satisfy the KKT conditions. Since these conditions are both necessary and sufficient for optimality in the case of a linear programming problem, the lemma is true. □

Our main task is to show that the conclusion of Lemma 2.4 is always valid under our very weak assumption that $\mathcal{S}_0$ is bounded and nonempty. Therefore §§3 and 4 prove the convergence of the calculated points $\{\underline{x}^{(\ell)} : \ell=1,2,3,\ldots\}$ and the Lagrange multiplier estimates $\{\underline{\lambda}^{(\ell)} : \ell=1,2,3,\ldots\}$. This rather long analysis must overcome the following difficulties.

The updating formula (1.8) would cause $\lambda_k^{(\ell+1)}$ to be much larger than $\lambda_k^{(\ell)}$ if the denominator $[\sigma(\underline{a}_k^T\underline{x}^{(\ell)}-b_k)+1]$ were close to zero, but we must establish that $\{\underline{\lambda}^{(\ell)} : \ell=1,2,3,\ldots\}$ tends to a limit. In particular, suppose that the equality constraint $\underline{a}^T\underline{x}=b$ is expressed as the two inequalities

$$\underline{a}_1^T\underline{x} \geq b_1 \quad \text{and} \quad \underline{a}_2^T\underline{x} \geq b_2, \tag{2.22}$$

where $\underline{a}_1=\underline{a}$, $b_1=b$, $\underline{a}_2=-\underline{a}$, and $b_2=-b$. Then the updating formula gives the identity

$$\lambda_1^{(\ell+1)}\lambda_2^{(\ell+1)} = \lambda_1^{(\ell)}\lambda_2^{(\ell)}/[1-\sigma^2(\underline{a}^T\underline{x}^{(\ell)}-b)^2]. \tag{2.23}$$

Thus the products $\{\lambda_1^{(\ell)}\lambda_2^{(\ell)} : \ell=1,2,3,\ldots\}$ increase monotonically, so no recovery can occur if a large increase in the product is caused by a small denominator. Therefore our analysis must show that the total damage from the denominators being less than one does not prevent the convergence of the Lagrange multiplier estimates as $\ell\to\infty$.

Another difficulty is nonuniqueness of the solution of the linear programming problem, which occurs, for example, when the objective function (1.1) is identically zero and the constraints (1.2) are satisfied by many values of $\underline{x}$. We must identify the appropriate limit of the sequence $\{\underline{x}^{(\ell)} : \ell=1,2,3,\ldots\}$ in this case. Furthermore, our analysis will imply that the second derivative matrix (2.3) tends to singularity if the linear programming problem has more than one solution, due to the components of $\underline{\lambda}^{(\ell)}$ that converge to zero.

Difficulties arise also from the shifts in the log barrier functions, because this construction keeps the calculated points $\{\underline{x}^{(\ell)} : \ell=1,2,3,\ldots\}$ in $\mathcal{S}_1$, but we require the limit points of the sequence to be in the original feasible region $\mathcal{S}_0$. For example, consider the linear programming problem

$$\begin{aligned} &\text{Minimize} \quad \underline{c}^T\underline{x}=x_2, \qquad \underline{x}\in\mathcal{R}^2, \\ &\text{subject to} \quad x_2\geq 0, \ -x_2\geq -1, \ x_1+x_2\geq -1, \ -x_1+x_2\geq -1, \end{aligned} \tag{2.24}$$

which has the set of solutions $\underline{x}^{(*)}\in\{(\theta,0)^T : -1\leq\theta\leq 1\}$. Let $\sigma=1$, let $\lambda_1^{(\ell)}\approx 1$, and let the remaining Lagrange multiplier estimates be very small. Then the first component of the vector equation (1.6) takes the form

$$\lambda_3^{(\ell)}/(x_1^{(\ell)}+x_2^{(\ell)}+2) - \lambda_4^{(\ell)}/(-x_1^{(\ell)}+x_2^{(\ell)}+2) = 0, \tag{2.25}$$

which gives the value

$$x_1^{(\ell)} = (\lambda_3^{(\ell)}-\lambda_4^{(\ell)})\,(x_2^{(\ell)}+2)/(\lambda_3^{(\ell)}+\lambda_4^{(\ell)}). \tag{2.26}$$

Since our assumptions on $\underline{\lambda}^{(\ell)}$ imply $x_2^{(\ell)}\approx 0$, it follows that $x_1^{(\ell)}$ depends mainly on the ratio $\lambda_3^{(\ell)}/\lambda_4^{(\ell)}$. Thus we achieve the required feasibility condition $-1\leq x_1^{(\ell)}\leq 1$

in the limit $\ell \to \infty$ if and only if the algorithm tends to give the property $\frac{1}{3} \leq \lambda_3^{(\ell)}/\lambda_4^{(\ell)} \leq 3$. In fact both $\lambda_3^{(\ell)}$ and $\lambda_4^{(\ell)}$ converge to zero, so this example shows that, in addition to proving the convergence of the sequence $\{\underline{\lambda}^{(\ell)} : \ell = 1,2,3,\ldots\}$, our analysis must take account of the ratios of the components of $\underline{\lambda}^{(\ell)}$ that tend to zero. Therefore we require some properties of the algorithm that are stronger than the assertions of Lemmas 2.2 and 2.3. Actually, this linear programming problem is solved satisfactorily, because (2.25) shows that the updating formula (1.8) provides $\lambda_4^{(\ell+1)} = \lambda_3^{(\ell+1)}$, and then formula (2.26) gives $x_1^{(\ell)} = 0$, $\ell \geq 2$.

3. The constraints that are active at all solutions. The linear programming problem that we are considering has at least one solution due to the assumption that the feasible region $\mathcal{S}_0$ is bounded and nonempty. We let $\mathcal{X}^{(*)}$ be the set of optimal vectors of variables, and, as in §1, we let $\mathcal{K}^{(*)}$ be the set of indices of the constraints that hold as equations at every point of $\mathcal{X}^{(*)}$, so we have the identities

$$\underline{a}_k^T \underline{x}^{(*)} = b_k, \quad k \in \mathcal{K}^{(*)}, \quad \underline{x}^{(*)} \in \mathcal{X}^{(*)}. \tag{3.1}$$

The analysis of this section will prove that the algorithm has the property

$$\lim_{\ell \to \infty} \underline{a}_k^T \underline{x}^{(\ell)} = b_k, \quad k \in \mathcal{K}^{(*)}. \tag{3.2}$$

We will find also that the Lagrange multiplier estimates $\{\underline{\lambda}^{(\ell)} : \ell = 1,2,3,\ldots\}$ converge to a limit $\underline{\lambda}^{(*)}$ whose components satisfy the conditions

$$\lambda_k^{(*)} > 0, \quad k \in \mathcal{K}^{(*)}, \quad \text{and} \quad \lambda_k^{(*)} = 0, \quad k \notin \mathcal{K}^{(*)}. \tag{3.3}$$

These assertions depend on the existence of strictly complementary solutions of the linear programming problem and its dual, which is proved in Corollary 2A of Goldman and Tucker (1956) for a different formulation of the problem. Specifically, the following properties of $\mathcal{K}^{(*)}$ will be required.

LEMMA 3.1. *If $\mathcal{K}^{(*)}$ is nonempty, then there exist multipliers $\{\mu_k : k \in \mathcal{K}^{(*)}\}$ that are all positive and that give the formula*

$$\underline{c} = \sum_{k \in \mathcal{K}^{(*)}} \mu_k \underline{a}_k. \tag{3.4}$$

Moreover, if some of the constraint indices are not in $\mathcal{K}^{(*)}$, then there exists a solution of the linear programming problem, $\hat{\underline{x}}^{(*)}$ say, that satisfies the conditions

$$\underline{a}_k^T \hat{\underline{x}}^{(*)} \geq b_k + \hat{h}, \quad k \in \{1,2,\ldots,m\} \backslash \mathcal{K}^{(*)}, \tag{3.5}$$

where $\hat{h}$ is a positive constant.

Proof. We pick any strictly complementary solution of the linear programming problem and its dual, say $\hat{\underline{x}}^{(*)} \in \mathcal{R}^n$ and $\underline{\mu} \in \mathcal{R}^m$, and we define $\mathcal{L}^{(*)}$ to be the set $\{k : \mu_k > 0\}$. Then the equation

$$\underline{c} = \sum_{k \in \mathcal{L}^{(*)}} \mu_k \underline{a}_k \tag{3.6}$$

holds, and, if some of the constraint indices are not in $\mathcal{L}^{(*)}$, we can let $\hat{h}$ be the greatest number that is allowed by the inequalities

$$\underline{a}_k^T \hat{\underline{x}}^{(*)} \geq b_k + \hat{h}, \quad k \in \{1,2,\ldots,m\} \backslash \mathcal{L}^{(*)}. \tag{3.7}$$

It follows from strict complementarity that $\hat{h}$ is positive. Therefore the theorem is true if $\mathcal{L}^{(*)}$ is the same as the set $\mathcal{K}^{(*)}$ that has been defined already.

Expression (3.7) with $\hat{h}>0$ and $\underline{\hat{x}}^{(*)}\in\mathcal{X}^{(*)}$ imply that, if $k\notin\mathcal{L}^{(*)}$, then $k\notin\mathcal{K}^{(*)}$, so we have the property $\mathcal{K}^{(*)}\subset\mathcal{L}^{(*)}$. Moreover, if $\underline{x}^{(*)}$ is any optimal vector of variables, then optimality, (3.6), and the KKT conditions at $\underline{\hat{x}}^{(*)}$ give the identity

$$(3.8)\quad 0=\underline{c}^T\underline{x}^{(*)}-\underline{c}^T\underline{\hat{x}}^{(*)}=\sum_{k\in\mathcal{L}^{(*)}}\mu_k\,(\underline{a}_k^T\underline{x}^{(*)}-\underline{a}_k^T\underline{\hat{x}}^{(*)})=\sum_{k\in\mathcal{L}^{(*)}}\mu_k\,(\underline{a}_k^T\underline{x}^{(*)}-b_k).$$

Furthermore, the feasibility of $\underline{x}^{(*)}$ implies that the factors $\{\underline{a}_k^T\underline{x}^{(*)}-b_k : k\in\mathcal{L}^{(*)}\}$ are nonnegative and we recall that the multipliers $\{\mu_k : k\in\mathcal{L}^{(*)}\}$ are strictly positive. Therefore the constraint residuals $\{\underline{a}_k^T\underline{x}^{(*)}-b_k : k\in\mathcal{L}^{(*)}\}$ are zero for every $\underline{x}^{(*)}\in\mathcal{X}^{(*)}$, which is the condition $\mathcal{L}^{(*)}\subset\mathcal{K}^{(*)}$. It follows that $\mathcal{L}^{(*)}$ is the set $\mathcal{K}^{(*)}$ as required. □

It would not be helpful to deduce the assertion (3.2) from our analysis if the set $\mathcal{K}^{(*)}$ were empty. This cannot happen if $\underline{c}$ is nonzero, because we are able to satisfy (3.4). It also cannot happen if $\underline{c}$ is zero and $\mathcal{S}_0$ has no interior, because now a nontrivial, nonnegative linear combination of the constraint gradients $\{\underline{a}_k : k=1,2,\ldots,m\}$ vanishes, the multipliers of the combination being admissible as Lagrange multipliers of KKT conditions, so again $\mathcal{L}^{(*)}=\mathcal{K}^{(*)}$ is nonempty. The set $\mathcal{K}^{(*)}$ is empty, however, when $\underline{c}$ vanishes and $\mathcal{S}_0$ has an interior, because the interior points of $\mathcal{S}_0$ are in $\mathcal{X}^{(*)}$. In this case, therefore, we apply the algorithm of §1 to the linear programming problem that is the subject of the following lemma.

LEMMA 3.2. *Let the objective function of the original linear programming problem be identically zero and let the constraints* (1.2) *define a feasible region* $\mathcal{S}_0$ *that has a nonempty interior. Furthermore, let* $\hat{h}$ *have the largest value that is allowed by the statement of Lemma* 3.1. *We introduce the quantities*

$$(3.9)\qquad \hat{\underline{a}}_k=(\sigma\hat{h}+1)^{-1}\underline{a}_k\quad\text{and}\quad\hat{b}_k=(\sigma\hat{h}+1)^{-1}(b_k+\hat{h}),\quad k=1,2,\ldots,m.$$

Then the feasible region, $\hat{\mathcal{S}}_0$ *say, of the linear programming problem*

$$(3.10)\qquad \begin{aligned}&\text{Minimize}\quad \underline{c}^T\underline{x}\ (\equiv 0),\qquad \underline{x}\in\mathcal{R}^n,\\ &\text{subject to}\quad \hat{\underline{a}}_k^T\underline{x}\geq\hat{b}_k,\qquad k=1,2,\ldots,m,\end{aligned}$$

is nonempty but has no interior. We apply the algorithm of §1 *to the new problem, using the old value of* σ *and starting with the Lagrange multiplier estimates* $\hat{\underline{\lambda}}^{(1)}=(\sigma\hat{h}+1)\underline{\lambda}^{(1)}$, *and we let* $\{\underline{\hat{x}}^{(\ell)} : \ell=1,2,3,\ldots\}$ *and* $\{\hat{\underline{\lambda}}^{(\ell)} : \ell=1,2,3,\ldots\}$ *be the vectors of variables and Lagrange multiplier estimates that are generated. Then these vectors are related to the original ones by the identities*

$$(3.11)\qquad \hat{\underline{\lambda}}^{(\ell)}=(\sigma\hat{h}+1)^{\ell}\underline{\lambda}^{(\ell)}\quad\text{and}\quad\underline{\hat{x}}^{(\ell)}=\underline{x}^{(\ell)},\quad \ell=1,2,3,\ldots\ .$$

Proof. Let $\underline{\hat{x}}^{(*)}$ satisfy the conditions (3.5) for the above value of $\hat{h}$. Then, because $\mathcal{K}^{(*)}$ is empty, expression (3.9) provides the bounds

$$(3.12)\quad \hat{\underline{a}}_k^T\underline{\hat{x}}^{(*)}=(\sigma\hat{h}+1)^{-1}\underline{a}_k^T\underline{\hat{x}}^{(*)}\geq(\sigma\hat{h}+1)^{-1}(b_k+\hat{h})=\hat{b}_k,\quad k=1,2,\ldots,m,$$

which shows that the constraints of the problem (3.10) are consistent. Moreover, if $\underline{x}$ were an interior point of $\hat{\mathcal{S}}_0$, then it would satisfy the strict inequalities

$$(3.13)\qquad \underline{a}_k^T\underline{x}=(\sigma\hat{h}+1)\hat{\underline{a}}_k^T\underline{x}>(\sigma\hat{h}+1)\hat{b}_k=b_k+\hat{h},\quad k=1,2,\ldots,m,$$

which contradicts the maximal property of the given choice of $\hat{h}$, the optimality of the objective function at $\underline{x}$ being trivial because $\underline{c}$ is zero. Therefore $\hat{\mathcal{S}}_0$ is nonempty but has no interior as required.

We establish the relations (3.11) by induction, knowing that the first one is true when $\ell=1$. Therefore we assume that $\hat{\underline{\lambda}}^{(\ell)}$ has the value $(\sigma\hat{h}+1)^{\ell}\underline{\lambda}^{(\ell)}$ for general ℓ. Then $\hat{\underline{x}}^{(\ell)}$ is the minimizer of the function

$$\hat{\Phi}^{(\ell)}(\underline{x}) = \sigma\,\underline{c}^T\underline{x} - (\sigma\hat{h}+1)^{\ell}\sum_{k=1}^{m}\lambda_k^{(\ell)}\log[\,\sigma(\underline{\hat{a}}_k^T\underline{x}-\hat{b}_k)+1\,], \quad \underline{x}\in\hat{\mathcal{S}}_1, \tag{3.14}$$

where $\underline{x}$ is in $\hat{\mathcal{S}}_1$ if and only if the inequalities

$$\sigma(\underline{\hat{a}}_k^T\underline{x}-\hat{b}_k)+1 \geq 0, \quad k=1,2,\ldots,m, \tag{3.15}$$

hold. Now the definitions (3.9) are chosen to provide the identities

$$\sigma(\underline{\hat{a}}_k^T\underline{x}-\hat{b}_k)+1 = (\sigma\hat{h}+1)^{-1}[\,\sigma(\underline{a}_k^T\underline{x}-b_k)+1\,], \quad k=1,2,\ldots,m, \tag{3.16}$$

for every $\underline{x}$, so the sets $\hat{\mathcal{S}}_1$ and $\mathcal{S}_1$ are the same. Furthermore, remembering $\underline{c}=0$, it follows from the definitions (1.4) and (3.14) that we can write $\hat{\Phi}^{(\ell)}$ in the form

$$\hat{\Phi}^{(\ell)}(\underline{x}) = (\sigma\hat{h}+1)^{\ell}\Phi^{(\ell)}(\underline{x}) + (\sigma\hat{h}+1)^{\ell}\sum_{k=1}^{m}\lambda_k^{(\ell)}\log(\sigma\hat{h}+1), \quad \underline{x}\in\mathcal{S}_1. \tag{3.17}$$

Hence the vectors that minimize $\hat{\Phi}^{(\ell)}$ and $\Phi^{(\ell)}$, namely, $\hat{\underline{x}}^{(\ell)}$ and $\underline{x}^{(\ell)}$ are equal for the current ℓ. Then the updating formula (1.8) gives the new Lagrange multiplier estimates

$$\begin{aligned}\hat{\lambda}_k^{(\ell+1)} &= \hat{\lambda}_k^{(\ell)}/\,[\,\sigma(\underline{\hat{a}}_k^T\underline{x}^{(\ell)}-\hat{b}_k)+1\,] \\ &= (\sigma\hat{h}+1)^{\ell}\lambda_k^{(\ell)}/\left\{(\sigma\hat{h}+1)^{-1}[\,\sigma(\underline{a}_k^T\underline{x}^{(\ell)}-b_k)+1\,]\right\} \\ &= (\sigma\hat{h}+1)^{\ell+1}\lambda_k^{(\ell+1)}, \quad k=1,2,\ldots,m, \end{aligned} \tag{3.18}$$

which completes the inductive argument that establishes the assertions (3.11). Therefore the lemma is true. □

The last part of expression (3.11) shows that, in order to assist our analysis of the sequence $\{\underline{x}^{(\ell)} : \ell=1,2,3,\ldots\}$ that is defined in §1, we can replace the original linear programming problem by the calculation (3.10) if $\underline{c}$ is zero and if $\mathcal{S}_0$ has interior points. Therefore we assume without loss of generality that the set $\mathcal{K}^{(*)}$ is nonempty. Further use will be made of Lemma 3.2 in §4.

Our analysis will require a bound on the Lagrange multiplier estimates $\{\underline{\lambda}^{(\ell)} : \ell=1,2,3,\ldots\}$. It is easy to deduce uniform boundedness if $\mathcal{S}_0$ has an interior point, $\hat{\underline{x}}$ say. Indeed, Lemma 2.3 and the definition of $\phi^{(\ell)}$ imply the inequalities

$$\begin{aligned}\phi^{(1)} &\leq \phi^{(\ell)} \leq \Phi^{(\ell)}(\hat{\underline{x}}) \\ &= \sigma\,\underline{c}^T\hat{\underline{x}} - \sum_{k=1}^{m}\lambda_k^{(\ell)}\log[\,\sigma(\underline{a}_k^T\hat{\underline{x}}-b_k)+1\,], \quad k=1,2,\ldots,m. \end{aligned} \tag{3.19}$$

Therefore the conditions

$$\lambda_k^{(\ell)} \leq (\sigma\,\underline{c}^T\hat{\underline{x}}-\phi^{(1)})\,/\log[\,\sigma(\underline{a}_k^T\hat{\underline{x}}-b_k)+1\,], \quad k=1,2,\ldots,m, \tag{3.20}$$

are satisfied for every positive integer ℓ. Next we derive an adequate bound for the general case when the interior of $\mathcal{S}_0$ may be empty.

LEMMA 3.3. *There exists a positive constant w_2 that provides the property*

$$\lambda_k^{(\ell)} \leq w_2 \ell, \quad k=1,2,\ldots,m, \quad \ell=1,2,3,\ldots . \tag{3.21}$$

Proof. The updating formula (1.8) allows the first statement of Lemma 2.3 to be expressed in the form

$$\sum_{k=1}^{m} \lambda_k^{(\ell+1)} [\sigma(\underline{a}_k^T \underline{x}^{(\ell)} - b_k)]^2 = \sum_{k=1}^{m} \frac{(\lambda_k^{(\ell+1)} - \lambda_k^{(\ell)})^2}{\lambda_k^{(\ell+1)}} \leq w_1^{-1}(\phi^{(\ell+1)} - \phi^{(\ell)}). \tag{3.22}$$

We introduce the monotonically increasing numbers

$$\rho^{(\ell)} = \max_{j=1,2,\ldots,\ell} \max_{k=1,2,\ldots,m} \lambda_k^{(j)}, \quad \ell=1,2,3,\ldots . \tag{3.23}$$

They satisfy the inequality

$$\frac{(\rho^{(\ell+1)} - \rho^{(\ell)})^2}{\rho^{(\ell+1)}} \leq \sum_{k=1}^{m} \frac{(\lambda_k^{(\ell+1)} - \lambda_k^{(\ell)})^2}{\lambda_k^{(\ell+1)}}, \tag{3.24}$$

which is trivial if $\rho^{(\ell+1)} = \rho^{(\ell)}$, and otherwise condition (3.24) follows from the remark

$$\frac{(\rho^{(\ell+1)} - \rho^{(\ell)})^2}{\rho^{(\ell+1)}} = \frac{(\lambda_k^{(\ell+1)} - \rho^{(\ell)})^2}{\lambda_k^{(\ell+1)}} \leq \frac{(\lambda_k^{(\ell+1)} - \lambda_k^{(\ell)})^2}{\lambda_k^{(\ell+1)}}, \tag{3.25}$$

where k is such that $\rho^{(\ell+1)} = \lambda_k^{(\ell+1)}$. Hence expression (3.22) provides the bound

$$(\rho^{(\ell+1)} - \rho^{(\ell)})^2 / \rho^{(\ell+1)} \leq w_1^{-1}(\phi^{(\ell+1)} - \phi^{(\ell)}). \tag{3.26}$$

Now the Cauchy–Schwarz inequality gives the relation

$$\rho^{(\ell+1)} - \rho^{(1)} = \sum_{j=1}^{\ell} (\rho^{(j+1)} - \rho^{(j)}) \leq \left[\sum_{j=1}^{\ell} (\rho^{(j+1)} - \rho^{(j)})^2 / \rho^{(j+1)} \sum_{j=1}^{\ell} \rho^{(j+1)} \right]^{\frac{1}{2}}. \tag{3.27}$$

By squaring both sides, by dropping the $(\rho^{(1)})^2$ term from the left, by bounding $\sum_{j=1}^{\ell} \rho^{(j+1)}$ by $\ell \rho^{(\ell+1)}$, and by dividing by $\rho^{(\ell+1)}$, we find the condition

$$\begin{aligned} \rho^{(\ell+1)} - 2\rho^{(1)} &\leq \ell \sum_{j=1}^{\ell} (\rho^{(j+1)} - \rho^{(j)})^2 / \rho^{(j+1)} \\ &\leq \ell w_1^{-1}(\phi^{(\ell+1)} - \phi^{(1)}), \quad \ell=1,2,3,\ldots, \end{aligned} \tag{3.28}$$

where the last line depends on expression (3.26). Therefore the positivity of $\rho^{(1)}$ and the first statement of Lemma 2.2 imply the bound

$$\rho^{(\ell+1)} \leq (\ell+1)\rho^{(1)} + \ell w_1^{-1}(\sigma \underline{c}^T \underline{x}^{(*)} - \phi^{(1)}), \quad \ell=1,2,3,\ldots . \tag{3.29}$$

Furthermore, this bound is trivial when $\ell = 0$. It follows from the definition (3.23) that the lemma is true when w_2 is the constant $\rho^{(1)} + w_1^{-1}(\sigma \underline{c}^T \underline{x}^{(*)} - \phi^{(1)})$. □

We can now establish that property (3.2) is achieved by a subsequence of the integers ℓ. The proof of this result depends on the positive constants $\{\mu_k : k\in\mathcal{K}^{(*)}\}$ that are introduced in Lemma 3.1.

LEMMA 3.4. *The algorithm provides the limit*

$$\lim_{\ell\to\infty}\frac{1}{\ell}\sum_{j=1}^{\ell}\left\{\sum_{k\in\mathcal{K}^{(*)}}(\underline{a}_k^T\underline{x}^{(j)}-b_k)^2\right\}=0. \tag{3.30}$$

Proof. Let $\Pi^{(\ell)}$ be the product

$$\Pi^{(\ell)}=\prod_{k\in\mathcal{K}^{(*)}}(\lambda_k^{(\ell)})^{\mu_k},\quad \ell=1,2,3,\ldots, \tag{3.31}$$

and let $\bar{\theta}$ be the constant that is defined just after (2.15). It follows from the updating formula (1.8) and the inequality $\log(\theta+1)\leq\theta-\frac{1}{2}\theta^2(\bar{\theta}+1)^{-2}$, $-1<\theta\leq\bar{\theta}$, that we have the relation

$$\begin{aligned}\log(\Pi^{(\ell+1)}/\Pi^{(\ell)}) &= -\sum_{k\in\mathcal{K}^{(*)}}\mu_k\log[\,\sigma(\underline{a}_k^T\underline{x}^{(\ell)}-b_k)+1\,]\\ &\geq -\sum_{k\in\mathcal{K}^{(*)}}\mu_k\,\sigma(\underline{a}_k^T\underline{x}^{(\ell)}-b_k)+\tfrac{1}{2}(\bar{\theta}+1)^{-2}\sum_{k\in\mathcal{K}^{(*)}}\mu_k\,[\sigma(\underline{a}_k^T\underline{x}^{(\ell)}-b_k)\,]^2. \qquad (3.32)\end{aligned}$$

Furthermore, let $\underline{x}^{(*)}$ be any solution of the original linear programming problem. Then the definition of $\mathcal{K}^{(*)}$ and (3.4) give the identity

$$\sum_{k\in\mathcal{K}^{(*)}}\mu_k\,\sigma(\underline{a}_k^T\underline{x}^{(\ell)}-b_k)=\sigma\sum_{k\in\mathcal{K}^{(*)}}\mu_k(\underline{a}_k^T\underline{x}^{(\ell)}-\underline{a}_k^T\underline{x}^{(*)})=\sigma\,\underline{c}^T(\underline{x}^{(\ell)}-\underline{x}^{(*)}). \tag{3.33}$$

Furthermore, (2.11) and the feasibility of $\underline{x}^{(*)}$ imply the bound

$$\underline{c}^T(\underline{x}^{(\ell)}-\underline{x}^{(*)})=\sum_{k=1}^{m}\lambda_k^{(\ell+1)}\underline{a}_k^T(\underline{x}^{(\ell)}-\underline{x}^{(*)})\leq\sum_{k=1}^{m}\lambda_k^{(\ell+1)}(\underline{a}_k^T\underline{x}^{(\ell)}-b_k). \tag{3.34}$$

Therefore condition (3.32) provides the inequality

$$\begin{aligned}&\sum_{k\in\mathcal{K}^{(*)}}\mu_k\,[\,\sigma(\underline{a}_k^T\underline{x}^{(\ell)}-b_k)\,]^2\\ &\qquad\leq 2\,(\bar{\theta}+1)^2\left\{\log(\Pi^{(\ell+1)}/\Pi^{(\ell)})+\sum_{k=1}^{m}\lambda_k^{(\ell+1)}[\,\sigma(\underline{a}_k^T\underline{x}^{(\ell)}-b_k)\,]\right\}. \qquad (3.35)\end{aligned}$$

Now Cauchy–Schwarz and Lemmas 2.3 and 3.3 yield the relation

$$\begin{aligned}\sum_{k=1}^{m}\lambda_k^{(\ell+1)}[\,\sigma(\underline{a}_k^T\underline{x}^{(\ell)}-b_k)\,] &\leq\left\{\sum_{k=1}^{m}\lambda_k^{(\ell+1)}[\,\sigma(\underline{a}_k^T\underline{x}^{(\ell)}-b_k)\,]^2\sum_{k=1}^{m}\lambda_k^{(\ell+1)}\right\}^{\frac{1}{2}}\\ &\leq[\,m(w_2/w_1)\,(\ell+1)\,(\phi^{(\ell+1)}-\phi^{(\ell)})\,]^{\frac{1}{2}}. \qquad (3.36)\end{aligned}$$

Furthermore, if p and q are any integers that satisfy $1 \leq p < q$, another application of Cauchy–Schwarz implies the property

$$\sum_{\ell=p}^{q-1} (\ell+1)^{1/2} (\phi^{(\ell+1)} - \phi^{(\ell)})^{1/2} \leq \left\{ \sum_{\ell=p}^{q-1} (\ell+1) \sum_{\ell=p}^{q-1} (\phi^{(\ell+1)} - \phi^{(\ell)}) \right\}^{\frac{1}{2}}$$

$$\leq q \, (\phi^{(q)} - \phi^{(p)})^{\frac{1}{2}}. \tag{3.37}$$

It follows from expressions (3.35) and (3.36) that we have the inequality

$$\sum_{\ell=p}^{q-1} \left\{ \sum_{k \in \mathcal{K}^{(*)}} \mu_k \, [\, \sigma(\underline{a}_k^T \underline{x}^{(\ell)} - b_k) \,]^2 \right\}$$

$$\leq 2 \, (\bar{\theta}+1)^2 \left\{ \log(\Pi^{(q)} / \Pi^{(p)}) + q \, [\, m(w_2/w_1) \, (\phi^{(q)} - \phi^{(p)}) \,]^{\frac{1}{2}} \right\}. \tag{3.38}$$

Let ϵ be any positive number. Then, because the sequence $\{\phi^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ increases monotonically and is bounded above, there exists a fixed integer L that gives the condition

$$2 \, (\bar{\theta}+1)^2 [\, m(w_2/w_1) \, (\phi^{(\ell)} - \phi^{(L)}) \,]^{\frac{1}{2}} \leq \epsilon, \quad \ell \geq L. \tag{3.39}$$

Thus, when ℓ exceeds L, expression (3.38) provides the inequality

$$\sum_{j=1}^{\ell} \left\{ \sum_{k \in \mathcal{K}^{(*)}} \mu_k \, [\, \sigma(\underline{a}_k^T \underline{x}^{(j)} - b_k) \,]^2 \right\}$$

$$= \sum_{j=1}^{L-1} \left\{ \sum_{k \in \mathcal{K}^{(*)}} \mu_k \, [\, \sigma(\underline{a}_k^T \underline{x}^{(j)} - b_k) \,]^2 \right\} + \sum_{j=L}^{\ell} \left\{ \sum_{k \in \mathcal{K}^{(*)}} \mu_k \, [\, \sigma(\underline{a}_k^T \underline{x}^{(j)} - b_k) \,]^2 \right\}$$

$$\leq 2 \, (\bar{\theta}+1)^2 \left\{ \log \left(\frac{\Pi^{(\ell+1)}}{\Pi^{(1)}} \right) + L \left[m \frac{w_2}{w_1} (\phi^{(L)} - \phi^{(1)}) \right]^{\frac{1}{2}} \right\} + (\ell+1)\epsilon. \tag{3.40}$$

Moreover, the definition (3.31) and Lemma 3.3 imply the bound

$$\log \Pi^{(\ell+1)} \leq \sum_{k \in \mathcal{K}^{(*)}} \mu_k \, [\, \log w_2 + \log(\ell+1) \,]. \tag{3.41}$$

Therefore, by dividing both sides of expression (3.40) by ℓ and by taking the limit as $\ell \to \infty$, we deduce the condition

$$\lim_{\ell \to \infty} \frac{1}{\ell} \sum_{j=1}^{\ell} \left\{ \sum_{k \in \mathcal{K}^{(*)}} \mu_k \, [\, \sigma(\underline{a}_k^T \underline{x}^{(j)} - b_k) \,]^2 \right\} \leq \epsilon. \tag{3.42}$$

Because σ and $\{\mu_k : k \in \mathcal{K}^{(*)}\}$ are positive constants, the required property (3.30) is now a consequence of the fact that ϵ can be arbitrarily small. □

We complete the analysis of this section by combining Lemma 3.4 with some properties of the sequence $\{\sigma \, \underline{c}^T \hat{\underline{x}}^{(*)} - \phi^{(\ell)} : \ell = 1, 2, 3, \ldots\}$, where $\hat{\underline{x}}^{(*)}$ is introduced

in Lemma 3.1 if some of the constraint indices are not in $\mathcal{K}^{(*)}$, and otherwise $\hat{\underline{x}}^{(*)}$ is the unique solution of the linear programming problem. Lemmas 2.2 and 2.3 show that this sequence is nonnegative and monotonically decreasing. Furthermore, the following assertion will help us to deduce that the sequence converges to zero at a rate that is at least R-linear.

LEMMA 3.5. *There exist constants w_3 and w_4, satisfying $w_3>0$ and $0<w_4<1$, such that, if the conditions*

$$|\underline{a}_k^T \underline{x}^{(\ell)} - b_k| \leq w_3, \quad k \in \mathcal{K}^{(*)}, \tag{3.43}$$

hold for some positive integer ℓ, then the algorithm provides the inequality

$$\sigma\, \underline{c}^T \hat{\underline{x}}^{(*)} - \phi^{(\ell+1)} \leq w_4 \left(\sigma\, \underline{c}^T \hat{\underline{x}}^{(*)} - \phi^{(\ell)} \right). \tag{3.44}$$

Proof. The right-hand side of expression (2.10) is zero if $\mathcal{K}^{(*)}$ includes all the constraint indices, in which case the assertion (3.44) is an immediate consequence of the bounds (2.5) and (2.10), the value of $\sigma\, \underline{c}^T \hat{\underline{x}}^{(*)} - \phi^{(\ell+1)}$ being zero. Therefore we assume for the remainder of the proof that condition (3.5) is satisfied.

Let $\mathcal{S}^{(*)}$ be the set

$$\mathcal{S}^{(*)} = \{\underline{x} : \underline{a}_k^T \underline{x} = b_k,\ k \in \mathcal{K}^{(*)}\} \subset \mathcal{R}^n, \tag{3.45}$$

so it includes the solutions of the linear programming problem. Furthermore, for every $\underline{x}$ in $\mathcal{R}^n$, let $\underline{s}(\underline{x})$ be the orthogonal projection of $\underline{x}$ into $\mathcal{S}^{(*)}$. We claim that there exists a positive constant w_5 that provides the property

$$\|\underline{x} - \underline{s}(\underline{x})\|_2 \leq w_5 \max\{|\underline{a}_k^T \underline{x} - b_k| : k \in \mathcal{K}^{(*)}\}, \quad \underline{x} \in \mathcal{R}^n. \tag{3.46}$$

Indeed, if we let $\mathcal{A}^{(*)} \subset \mathcal{R}^n$ be the linear space spanned by $\{\underline{a}_k : k \in \mathcal{K}^{(*)}\}$, and if the Euclidean length of $\underline{y} \in \mathcal{A}^{(*)}$ is one, then an elementary argument using continuity and compactness gives the inequality

$$\max\{|\underline{a}_k^T \underline{y}| : k \in \mathcal{K}^{(*)}\} \geq \epsilon, \tag{3.47}$$

where ϵ is a positive number that is independent of $\underline{y}$. Thus, using homogeneity to remove the restriction $\|\underline{y}\|_2 = 1$, we deduce the relation

$$\|\underline{y}\|_2 \leq \epsilon^{-1} \max\{|\underline{a}_k^T \underline{y}| : k \in \mathcal{K}^{(*)}\}, \quad \underline{y} \in \mathcal{A}^{(*)}. \tag{3.48}$$

Now the orthogonal projection construction of $\underline{s}(\underline{x})$ causes $\underline{x} - \underline{s}(\underline{x})$ to be in $\mathcal{A}^{(*)}$, in addition to providing the equations $\{\underline{a}_k^T \underline{s}(\underline{x}) = b_k : k \in \mathcal{K}^{(*)}\}$. Therefore, by substituting $\underline{x} - \underline{s}(\underline{x})$ for $\underline{y}$ in expression (3.48), we find that condition (3.46) holds when w_5 has the value ϵ^{-1}.

We are going to require the bound

$$\left| \sum_{k \notin \mathcal{K}^{(*)}} \lambda_k^{(\ell+1)} \underline{a}_k^T \left[\underline{x}^{(\ell)} - \underline{s}(\underline{x}^{(\ell)}) \right] \right| \leq \tfrac{1}{2} \hat{h} \sum_{k \notin \mathcal{K}^{(*)}} \lambda_k^{(\ell+1)} \tag{3.49}$$

when condition (3.43) holds, where $\hat{h}$ occurs in expression (3.5). Therefore we note that Cauchy–Schwarz and inequalities (3.46) and (3.43) provide the relation

$$\left| \sum_{k \notin \mathcal{K}^{(*)}} \lambda_k^{(\ell+1)} \underline{a}_k^T [\underline{x}^{(\ell)} - \underline{s}(\underline{x}^{(\ell)})] \right| \leq \|\underline{x}^{(\ell)} - \underline{s}(\underline{x}^{(\ell)})\|_2 \sum_{k \notin \mathcal{K}^{(*)}} \lambda_k^{(\ell+1)} \|\underline{a}_k\|_2$$

$$\leq w_5 w_3 \sum_{k \notin \mathcal{K}^{(*)}} \lambda_k^{(\ell+1)} \|\underline{a}_k\|_2. \tag{3.50}$$

It follows that we can achieve the bound (3.49) by letting w_3 be the constant

$$w_3 = \tfrac{1}{2}\hat{h} / [\, w_5 \max\{\|\underline{a}_k\|_2 : k \notin \mathcal{K}^{(*)}\}\,]. \tag{3.51}$$

Expressions (3.5) and (3.49) yield the inequality

$$\sum_{k \notin \mathcal{K}^{(*)}} \lambda_k^{(\ell+1)} [\, \underline{a}_k^T \underline{x}^{(\ell)} - b_k\,] \geq \sum_{k \notin \mathcal{K}^{(*)}} \lambda_k^{(\ell+1)} [\, \underline{a}_k^T \underline{x}^{(\ell)} - \underline{a}_k^T \hat{\underline{x}}^{(*)}\,] + \hat{h} \sum_{k \notin \mathcal{K}^{(*)}} \lambda_k^{(\ell+1)}$$

$$\geq \sum_{k \notin \mathcal{K}^{(*)}} \lambda_k^{(\ell+1)} [\, \underline{a}_k^T \underline{s}(\underline{x}^{(\ell)}) - \underline{a}_k^T \hat{\underline{x}}^{(*)}\,] + \tfrac{1}{2}\hat{h} \sum_{k \notin \mathcal{K}^{(*)}} \lambda_k^{(\ell+1)}, \tag{3.52}$$

and our next task is to show that the first of the two sums in the last line is zero. Indeed, because both $\underline{s}(\underline{x}^{(\ell)})$ and $\hat{\underline{x}}^{(*)}$ are in the set (3.45) and because (2.11) and (3.4) are satisfied, we find the identity

$$\sum_{k \notin \mathcal{K}^{(*)}} \lambda_k^{(\ell+1)} \underline{a}_k^T [\, \underline{s}(\underline{x}^{(\ell)}) - \hat{\underline{x}}^{(*)}\,] = \sum_{k=1}^{m} \lambda_k^{(\ell+1)} \underline{a}_k^T [\, \underline{s}(\underline{x}^{(\ell)}) - \hat{\underline{x}}^{(*)}\,]$$

$$= \underline{c}^T [\, \underline{s}(\underline{x}^{(\ell)}) - \hat{\underline{x}}^{(*)}\,] = \sum_{k \in \mathcal{K}^{(*)}} \mu_k \underline{a}_k^T [\, \underline{s}(\underline{x}^{(\ell)}) - \hat{\underline{x}}^{(*)}\,] = 0. \tag{3.53}$$

Therefore expression (3.52) and Cauchy–Schwarz give the condition

$$\tfrac{1}{2}\hat{h} \sum_{k \notin \mathcal{K}^{(*)}} \lambda_k^{(\ell+1)} \leq \left\{ \sum_{k \notin \mathcal{K}^{(*)}} \lambda_k^{(\ell+1)} [\, \underline{a}_k^T \underline{x}^{(\ell)} - b_k\,]^2 \sum_{k \notin \mathcal{K}^{(*)}} \lambda_k^{(\ell+1)} \right\}^{\frac{1}{2}}, \tag{3.54}$$

which can be written in the form

$$(\tfrac{1}{2}\sigma\hat{h})^2 \sum_{k \notin \mathcal{K}^{(*)}} \lambda_k^{(\ell+1)} \leq \sum_{k \notin \mathcal{K}^{(*)}} \lambda_k^{(\ell+1)} [\, \sigma(\underline{a}_k^T \underline{x}^{(\ell)} - b_k)\,]^2. \tag{3.55}$$

The relation (3.55), the positivity of w_1, the nonnegativity of the products $\{\lambda_k^{(\ell+1)} [\, \sigma(\underline{a}_k^T \underline{x}^{(\ell)} - b_k)\,]^2 : k \in \mathcal{K}^{(*)}\}$, and the first part of Lemma 2.3 imply the bound

$$w_1(\tfrac{1}{2}\sigma\hat{h})^2 \sum_{k \notin \mathcal{K}^{(*)}} \lambda_k^{(\ell+1)} \leq w_1 \sum_{k=1}^{m} \lambda_k^{(\ell+1)} [\, \sigma(\underline{a}_k^T \underline{x}^{(\ell)} - b_k)\,]^2 \leq \phi^{(\ell+1)} - \phi^{(\ell)}, \tag{3.56}$$

while the second part of Lemma 2.3 and $\hat{\underline{x}}^{(*)} \in \mathcal{S}^{(*)}$ provide the condition

$$\sigma\, \underline{c}^T \hat{\underline{x}}^{(*)} - \phi^{(\ell+1)} \leq \sigma \max_{j \notin \mathcal{K}^{(*)}} (\underline{a}_j^T \hat{\underline{x}}^{(*)} - b_j) \sum_{k \notin \mathcal{K}^{(*)}} \lambda_k^{(\ell+1)}. \tag{3.57}$$

By combining expressions (3.56) and (3.57) in the way that eliminates the sum $\sum_{k\notin\mathcal{K}^{(*)}}\lambda_k^{(\ell+1)}$, we find the inequality

$$\tfrac{1}{4}w_1\sigma\hat{h}^2\Big[\max_{j\notin\mathcal{K}^{(*)}}(\underline{a}_j^T\hat{\underline{x}}^{(*)}-b_j)\Big]^{-1}\Big(\sigma\,\underline{c}^T\hat{\underline{x}}^{(*)}-\phi^{(\ell+1)}\Big)\le w_1(\tfrac{1}{2}\sigma\hat{h})^2\sum_{k\notin\mathcal{K}^{(*)}}\lambda_k^{(\ell+1)}$$

$$\le\phi^{(\ell+1)}-\phi^{(\ell)}=\Big(\sigma\,\underline{c}^T\hat{\underline{x}}^{(*)}-\phi^{(\ell)}\Big)-\Big(\sigma\,\underline{c}^T\hat{\underline{x}}^{(*)}-\phi^{(\ell+1)}\Big). \tag{3.58}$$

It follows that the lemma is true when w_4 is the number

$$w_4=\Big\{1+\tfrac{1}{4}w_1\sigma\hat{h}^2\,[\max_{j\notin\mathcal{K}^{(*)}}(\underline{a}_j^T\hat{\underline{x}}^{(*)}-b_j)]^{-1}\Big\}^{-1}. \tag{3.59}$$

□

The main conclusion that we draw from Lemmas 3.4 and 3.5 is given next.

LEMMA 3.6. *The positive, monotonically decreasing sequence* $\{\sigma\,\underline{c}^T\hat{\underline{x}}^{(*)}-\phi^{(\ell)}:\ell=1,2,3,\ldots\}$ *has the property*

$$\sigma\,\underline{c}^T\hat{\underline{x}}^{(*)}-\phi^{(\ell)}\le w_6w_7^\ell,\quad\ell=1,2,3,\ldots, \tag{3.60}$$

where w_6 *and* w_7 *are constants that satisfy* $w_6>0$ *and* $0<w_7<1$.

Proof. We deduce from Lemma 3.4 that we can pick a fixed positive integer L that provides the bound

$$\sum_{j=1}^{\ell}\left\{\sum_{k\in\mathcal{K}^{(*)}}(\underline{a}_k^T\underline{x}^{(j)}-b_k)^2\right\}\le\tfrac{1}{2}\ell\,w_3^2,\quad\ell\ge L, \tag{3.61}$$

where w_3 is introduced in the statement of Lemma 3.5. This relation implies that the conditions

$$|\underline{a}_k^T\underline{x}^{(j)}-b_k|\le w_3,\quad k\in\mathcal{K}^{(*)}, \tag{3.62}$$

are achieved by at least half of the integers j in the range $[1,\ell]$. Therefore Lemma 3.5 and the monotonicity of the sequence $\{\sigma\,\underline{c}^T\hat{\underline{x}}^{(*)}-\phi^{(\ell)}:\ell=1,2,3,\ldots\}$ yield the bound

$$\sigma\,\underline{c}^T\hat{\underline{x}}^{(*)}-\phi^{(\ell+1)}\le w_4^{\ell/2}\Big(\sigma\,\underline{c}^T\hat{\underline{x}}^{(*)}-\phi^{(1)}\Big),\quad\ell\ge L. \tag{3.63}$$

Thus, remembering $w_4<1$, we have the relation

$$\sigma\,\underline{c}^T\hat{\underline{x}}^{(*)}-\phi^{(\ell)}\le w_4^{(\ell-L)/2}\Big(\sigma\,\underline{c}^T\hat{\underline{x}}^{(*)}-\phi^{(1)}\Big),\quad\ell>L, \tag{3.64}$$

which has the advantage of also being true when $\ell\le L$, due to the monotonicity property that has just been mentioned. It follows that we can satisfy inequality (3.60) by setting the values

$$w_6=w_4^{-L/2}\Big(\sigma\,\underline{c}^T\hat{\underline{x}}^{(*)}-\phi^{(1)}\Big)\quad\text{and}\quad w_7=w_4^{\frac{1}{2}}, \tag{3.65}$$

which completes the proof. □

We are now ready to justify (3.2). We emphasize the importance of this result by presenting it as a theorem.

THEOREM 3.7. *Let the algorithm of* §1 *be applied to minimize the function* (1.1) *subject to the constraints* (1.2), *where the feasible region is bounded and nonempty, and let* k *be the index of any constraint that is satisfied as an equation at all solutions of this problem. Then the sequence of calculated variables gives the limit*

$$\lim_{\ell\to\infty} \underline{a}_k^T \underline{x}^{(\ell)} = b_k. \tag{3.66}$$

Furthermore, this limit has the R-linear convergence property

$$|\underline{a}_k^T \underline{x}^{(\ell)} - b_k| \leq w_8 w_9^{\ell}, \quad \ell = 1, 2, 3, \ldots, \tag{3.67}$$

where w_8 *and* w_9 *are constants that satisfy* $w_8 > 0$ *and* $0 < w_9 < 1$.

Proof. Lemma 3.4 allows us to pick a fixed integer L that provides the condition

$$\sum_{j=1}^{\ell} (\underline{a}_k^T \underline{x}^{(j)} - b_k)^2 \leq [\tfrac{1}{2}\sigma^{-1} \log(1/w_7)]^2 \ell, \quad \ell \geq L, \tag{3.68}$$

where k is defined in the statement of the theorem and where w_7 occurs in expression (3.60). Thus we will obtain a lower bound on $\lambda_k^{(\ell+1)}$ that allows the required results to be deduced from inequalities (2.9) and (3.60).

The updating formula (1.8), the concavity of the log function, and Cauchy–Schwarz imply the relation

$$\log(\lambda_k^{(\ell+1)} / \lambda_k^{(1)}) = -\sum_{j=1}^{\ell} \log[\sigma(\underline{a}_k^T \underline{x}^{(j)} - b_k) + 1] \geq -\sigma \sum_{j=1}^{\ell} (\underline{a}_k^T \underline{x}^{(j)} - b_k)$$

$$\geq -\sigma \left[\ell \sum_{j=1}^{\ell} (\underline{a}_k^T \underline{x}^{(j)} - b_k)^2 \right]^{\frac{1}{2}}. \tag{3.69}$$

Therefore our choice of L gives the inequality

$$\log(\lambda_k^{(\ell+1)} / \lambda_k^{(1)}) \geq -\tfrac{1}{2}\ell \log(1/w_7), \quad \ell \geq L, \tag{3.70}$$

which is equivalent to the bound

$$\lambda_k^{(\ell+1)} \geq w_7^{\ell/2} \lambda_k^{(1)}, \quad \ell \geq L. \tag{3.71}$$

Hence, by retaining only one term from the sum of expression (2.9) and by invoking Lemma 2.2, we find the property

$$(\underline{a}_k^T \underline{x}^{(\ell)} - b_k)^2 \leq (w_1 \sigma^2 \lambda_k^{(\ell+1)})^{-1} (\phi^{(\ell+1)} - \phi^{(\ell)})$$

$$\leq (w_1 \sigma^2 w_7^{\ell/2} \lambda_k^{(1)})^{-1} \left(\sigma \underline{c}^T \hat{\underline{x}}^{(*)} - \phi^{(\ell)} \right), \quad \ell \geq L. \tag{3.72}$$

It follows from Lemma 3.6 that we have the relation

$$(\underline{a}_k^T \underline{x}^{(\ell)} - b_k)^2 \leq w_6 (w_1 \sigma^2 \lambda_k^{(1)})^{-1} w_7^{\ell/2}, \quad \ell \geq L. \tag{3.73}$$

We set $w_9 = w_7^{1/4}$ and choose a value of w_8 that is no less than $[w_6 (w_1 \sigma^2 \lambda_k^{(1)})^{-1}]^{1/2}$, in order that the required condition (3.67) is satisfied for $\ell \geq L$. Furthermore, by

increasing w_8 if necessary, we can accommodate the values of ℓ that are less than L. Thus inequality (3.67) is achieved, which implies the limit (3.66). □

We also offer a formal statement and proof of expression (3.3).

THEOREM 3.8. *Let the algorithm of* §1 *be applied to minimize the function* (1.1) *subject to the constraints* (1.2), *where the feasible region is bounded and nonempty. Then the sequence* $\{\underline{\lambda}^{(\ell)} : \ell=1,2,3,\ldots\}$ *of Lagrange multiplier estimates converges to a limit,* $\underline{\lambda}^{(*)}$ *say. Furthermore, the components of* $\underline{\lambda}^{(*)}$ *satisfy the conditions* (3.3).

Proof. Let k be a constraint index that is in $\mathcal{K}^{(*)}$. We consider the sequence

$$\lambda_k^{(\ell+1)} = \lambda_k^{(\ell)} / [\sigma(\underline{a}_k^T \underline{x}^{(\ell)} - b_k) + 1], \quad \ell = 1,2,3,\ldots, \tag{3.74}$$

that is given by the updating formula (1.8). Because every denominator and $\lambda_k^{(1)}$ are positive, it is elementary that this sequence has a nonzero limit if and only if the sum $\sum_{\ell=1}^{\infty} |\underline{a}_k^T \underline{x}^{(\ell)} - b_k|$ is finite. We see that this convergence property is implied by inequality (3.67).

Alternatively, if k is a constraint index that is not in $\mathcal{K}^{(*)}$, then the definition of $\mathcal{K}^{(*)}$ implies that we can pick a solution $\hat{\underline{x}}^{(*)}$ of the linear programming problem at which $\underline{a}_k^T \hat{\underline{x}}^{(*)} - b_k$ is positive. Then inequality (3.19) and Lemma 3.6 give the bound

$$\lambda_k^{(\ell)} \leq \left(\sigma \underline{c}^T \hat{\underline{x}}^{(*)} - \phi^{(\ell)}\right) / \log[\sigma(\underline{a}_k^T \hat{\underline{x}}^{(*)} - b_k) + 1] \leq w_6 w_7^{\ell} / \log[\sigma(\underline{a}_k^T \hat{\underline{x}}^{(*)} - b_k) + 1]. \tag{3.75}$$

It follows that $\{\lambda_k^{(\ell)} : \ell=1,2,3,\ldots\}$ tends to zero as required. □

It is mentioned in §1 that the convergence of the sequence $\{\underline{x}^{(\ell)} : \ell=1,2,3,\ldots\}$ can be deduced from the limits (3.2) if the given linear programming problem has a unique solution. In order to prove this assertion, we let $\underline{x}^{(*)}$ be the solution and we suppose that the sequence fails to converge to $\underline{x}^{(*)}$. Then, because the sequence is bounded and is in $\mathcal{R}^n$, it has a limit point, $\underline{x}^{(\infty)}$ say, that is different from $\underline{x}^{(*)}$. Furthermore, the definition of $\mathcal{K}^{(*)}$, expression (3.2), and the feasibility of $\underline{x}^{(*)}$ provide the conditions

$$\underline{a}_k^T \underline{x}^{(*)} = b_k = \underline{a}_k^T \underline{x}^{(\infty)}, \quad k \in \mathcal{K}^{(*)}, \quad \text{and} \quad \underline{a}_k^T \underline{x}^{(*)} > b_k, \quad k \notin \mathcal{K}^{(*)}. \tag{3.76}$$

It follows that we can pick a small positive value of θ so that the vector $\check{\underline{x}} = \underline{x}^{(*)} + \theta(\underline{x}^{(\infty)} - \underline{x}^{(*)})$ is feasible. Furthermore, (3.4) and (3.76) imply $\underline{c}^T \check{\underline{x}} = \underline{c}^T \underline{x}^{(*)}$. Therefore $\check{\underline{x}}$ is another optimal vector of variables. This contradiction establishes the limit $\underline{x}^{(\ell)} \to \underline{x}^{(*)}$ as $\ell \to \infty$.

On the other hand, if the linear programming problem has more than one solution, then the set (3.45) includes more than one point, while condition (3.2) states just that the limit points of the sequence $\{\underline{x}^{(\ell)} : \ell = 1,2,3,\ldots\}$ are in $\mathcal{S}^{(*)}$. Therefore we do not yet know if the sequence is convergent. Furthermore, we do not even know if its limit points satisfy the given constraints (1.2). The lengthy analysis of §4 and the appendix will answer these questions. It can be skipped by any reader who prefers not to study linear programming calculations that have many solutions.

4. The limiting values of the other constraints. It is convenient to introduce a notation for the constraint indices k whose residuals $\{\underline{a}_k^T \underline{x}^{(\ell)} - b_k : \ell=1,2,3,\ldots\}$ are known to converge satisfactorily. Specifically, we say that a set $\mathcal{K}$ of constraint indices is "established" when we have proved that it has the following three properties.

(i) For each $k \in \mathcal{K}$, the sequence $\{\underline{a}_k^T \underline{x}^{(\ell)} - b_k : \ell = 1, 2, 3, \ldots\}$ tends to a nonnegative limit, $r_k^{(*)}$ say. (ii) These sequences have the R-linear convergence rate

$$|\underline{a}_k^T \underline{x}^{(\ell)} - b_k - r_k^{(*)}| \leq w_8 w_9^{\ell}, \quad \ell = 1, 2, 3, \ldots, \quad k \in \mathcal{K}, \tag{4.1}$$

where w_8 and w_9 are numbers that are independent of ℓ, satisfying the strict inequalities $w_8 > 0$ and $0 < w_9 < 1$. (iii) There exists a solution $\underline{\hat{x}}^{(*)}$ of the original linear programming problem that satisfies the conditions

$$\begin{aligned} &\underline{a}_k^T \underline{\hat{x}}^{(*)} - b_k = r_k^{(*)}, && k \in \mathcal{K}, \\ &\underline{a}_k^T \underline{\hat{x}}^{(*)} \geq b_k, && k \in \{1, 2, \ldots, m\} \backslash \mathcal{K}. \end{aligned} \tag{4.2}$$

It follows from the analysis of §3 that $\mathcal{K}^{(*)}$ is "established." Indeed, (3.2) provides the limits $\{r_k^{(*)} = 0 : k \in \mathcal{K}^{(*)}\}$, Theorem 3.7 gives the convergence rate (4.1), and expression (4.2) is a consequence of the second part of Lemma 3.1 and the definition of $\mathcal{K}^{(*)}$. Here we are using the elementary fact that, because the number of constraints is finite, we can let w_8 and w_9 be independent of k in the statement of Theorem 3.7.

Alternatively, if $\mathcal{K}^{(*)}$ is empty, we let $\hat{\mathcal{K}}^{(*)}$ be the set of indices of the constraints that hold as equations at all solutions of the linear programming problem (3.10), and we show that $\hat{\mathcal{K}}^{(*)}$ is "established." In this case Lemma 3.2 and Theorem 3.7 imply the bounds

$$|\underline{\hat{a}}_k^T \underline{x}^{(\ell)} - \hat{b}_k| \leq w_8 w_9^{\ell}, \quad \ell = 1, 2, 3, \ldots, \quad k \in \hat{\mathcal{K}}^{(*)}. \tag{4.3}$$

Therefore, by substituting the definitions (3.9), we find the inequalities

$$|\underline{a}_k^T \underline{x}^{(\ell)} - b_k - \hat{h}| \leq (\sigma \hat{h} + 1) w_8 w_9^{\ell}, \quad \ell = 1, 2, 3, \ldots, \quad k \in \hat{\mathcal{K}}^{(*)}, \tag{4.4}$$

which provide the positive limits $\{r_k^{(*)} = \hat{h} : k \in \hat{\mathcal{K}}^{(*)}\}$, and which establish the conditions (4.1), because we are allowed to replace w_8 by $(\sigma \hat{h} + 1)$ times its original value. Furthermore, if $\underline{\hat{x}}^{(*)}$ is any solution of the problem (3.10), then the definitions (3.9) give the relation

$$\underline{a}_k^T \underline{\hat{x}}^{(*)} = (\sigma \hat{h} + 1) \underline{\hat{a}}_k^T \underline{\hat{x}}^{(*)} \geq (\sigma \hat{h} + 1) \hat{b}_k = b_k + \hat{h}, \quad k = 1, 2, \ldots, m. \tag{4.5}$$

Therefore, because this last inequality holds as an equation if k is in $\hat{\mathcal{K}}^{(*)}$, the conditions (4.2) are satisfied by this choice of $\underline{\hat{x}}^{(*)}$, which completes the proof that $\hat{\mathcal{K}}^{(*)}$ is "established."

The purpose of most of the analysis in this section is to show that, if $\mathcal{K}$ is any "established" set that does not include all of the constraint indices, then we can form a larger "established" set by adding one or more constraint indices to $\mathcal{K}$. It will follow by induction that $\{1, 2, \ldots, m\}$ is "established." Then we let $\underline{\hat{x}}^{(*)}$ be the solution of the original linear programming problem that occurs in expression (4.2), because the definition of $r_k^{(*)}$ gives the limits

$$\lim_{\ell \to \infty} (\underline{a}_k^T \underline{x}^{(\ell)} - b_k) = \underline{a}_k^T \underline{\hat{x}}^{(*)} - b_k, \quad k = 1, 2, \ldots, m. \tag{4.6}$$

Thus we deduce our main result, namely, that the sequence $\{\underline{x}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ converges to $\underline{\hat{x}}^{(*)}$. Indeed, if this conclusion were false, then the sequence would have

at least one other limit point, $\underline{x}^{(\infty)}$ say, and expression (4.6) would imply that all of the scalar products $\{\underline{a}_k^T(\hat{\underline{x}}^{(*)}-\underline{x}^{(\infty)}) : k=1,2,\ldots,m\}$ were zero. Then the point $\hat{\underline{x}}^{(*)}+\alpha(\hat{\underline{x}}^{(*)}-\underline{x}^{(\infty)})$ would be feasible for all real values of the multiplier α, which would contradict the boundedness of the feasible region. Therefore it is sufficient to prove that "established" sets with fewer than m elements can be enlarged. Often the following result can be applied.

LEMMA 4.1. *Let $\mathcal{K}$ be any "established" set. If there exists a constraint index j that is not in $\mathcal{K}$, and if $\underline{a}_j$ is in the linear space that is spanned by the vectors $\{\underline{a}_k : k\in\mathcal{K}\}$, then the set $\mathcal{K}\cup\{j\}$ is also "established."*

Proof. We write $\underline{a}_j$ in the form

$$\underline{a}_j = \sum_{k\in\mathcal{K}} \theta_k \underline{a}_k \tag{4.7}$$

in order to derive the relation

$$\underline{a}_j^T\underline{x}^{(\ell)} - \sum_{k\in\mathcal{K}} \theta_k(b_k+r_k^{(*)}) = \sum_{k\in\mathcal{K}} \theta_k(\underline{a}_k^T\underline{x}^{(\ell)}-b_k-r_k^{(*)}). \tag{4.8}$$

Thus expression (4.1) implies the bound

$$|\underline{a}_j^T\underline{x}^{(\ell)}-b_j-r_j^{(*)}| \le \sum_{k\in\mathcal{K}} |\theta_k|\, w_8 w_9^{\ell}, \quad \ell=1,2,3,\ldots, \tag{4.9}$$

where $r_j^{(*)}$ is the number

$$r_j^{(*)} = \sum_{k\in\mathcal{K}} \theta_k(b_k+r_k^{(*)}) - b_j. \tag{4.10}$$

Inequality (4.9) shows that $r_j^{(*)}$ is the limit of the sequence $\{\underline{a}_j^T\underline{x}^{(\ell)}-b_j : \ell=1,2,3,\ldots\}$ as required. Furthermore, (4.10), (4.2), and (4.7) provide the identity

$$r_j^{(*)} = \sum_{k\in\mathcal{K}} \theta_k\, \underline{a}_k^T\hat{\underline{x}}^{(*)} - b_j = \underline{a}_j^T\hat{\underline{x}}^{(*)}-b_j, \tag{4.11}$$

so it follows from the second line of expression (4.2) that $r_j^{(*)}$ is nonnegative. It also follows that the conditions (4.2) remain valid if $\mathcal{K}$ is augmented by j. Moreover, we deduce from the bound (4.9) that the conditions (4.1) also admit this augmentation, because we are allowed to overwrite w_8 by the number $\max[\,w_8,\ \sum_{k\in\mathcal{K}} |\theta_k|\, w_8\,]$. Therefore the lemma is true. □

Given an "established" set $\mathcal{K}$, which can be either $\mathcal{K}^{(*)}$ or $\hat{\mathcal{K}}^{(*)}$ initially according to the second and third paragraphs of this section, we enlarge it as much as possible by applying Lemma 4.1 recursively. Thus the resultant set $\mathcal{K}$ contains all the constraint indices, or the set

$$\check{\mathcal{K}} = \{1,2,\ldots,m\}\backslash\mathcal{K} \tag{4.12}$$

is nonempty, and none of the vectors $\{\underline{a}_k : k\in\check{\mathcal{K}}\}$ is in the span of the vectors $\{\underline{a}_k : k\in\mathcal{K}\}$. We investigate this latter case by applying the method that is mentioned briefly in the penultimate paragraph of §1.

Therefore we let $\hat{n}$ be the dimension of the space that is spanned by $\{\underline{a}_k : k\in\mathcal{K}\}$, we assume without loss of generality that $\mathcal{K}$ includes the first $\hat{n}$ integers and that the

vectors $\{\underline{a}_k : k=1,2,\ldots,\hat{n}\}$ are linearly independent, and we let A be the $n\times\hat{n}$ matrix whose columns are these vectors. Furthermore, we let Z be an $n\times\check{n}$ matrix whose columns are both orthonormal and orthogonal to the columns of A, where $\check{n}$ is the integer $n-\hat{n}$. It is straightforward to verify that these choices imply the formula

$$(4.13)\qquad \left(\begin{array}{c|c} A & Z \end{array} \right)^{-1} = \left(\begin{array}{c} (A^TA)^{-1}A^T \\ \hline Z^T \end{array} \right).$$

It follows that we can express each $\underline{x}^{(\ell)}$ in the form

$$(4.14)\qquad \underline{x}^{(\ell)} = \left(\begin{array}{c|c} A & Z \end{array} \right) \left(\begin{array}{c} (A^TA)^{-1}A^T \\ \hline Z^T \end{array} \right) \underline{x}^{(\ell)} = A\underline{\alpha}^{(\ell)} + Z\underline{\beta}^{(\ell)} = \underline{y}^{(\ell)} + \underline{z}^{(\ell)},$$

where $\underline{\alpha}^{(\ell)}$ and $\underline{\beta}^{(\ell)}$ are the vectors

$$(4.15)\qquad \underline{\alpha}^{(\ell)} = (A^TA)^{-1}A^T\underline{x}^{(\ell)} \in \mathcal{R}^{\hat{n}} \quad \text{and} \quad \underline{\beta}^{(\ell)} = Z^T\underline{x}^{(\ell)} \in \mathcal{R}^{\check{n}},$$

and where the last part of the identity (4.14) agrees with (1.11).

Now we recall from §1 that the algorithm generates $\underline{x}^{(\ell)}$ by minimizing the function (1.4). Therefore the least value of the expression

$$(4.16)\qquad \sigma\,\underline{c}^T(A\underline{\alpha}+Z\underline{\beta}) - \sum_{k=1}^{m} \lambda_k^{(\ell)} \log[\,\sigma(\underline{a}_k^TA\,\underline{\alpha}+\underline{a}_k^TZ\underline{\beta}-b_k)+1\,], \quad A\underline{\alpha}+Z\underline{\beta}\in\mathcal{S}_1,$$

occurs when $\underline{\alpha}$ and $\underline{\beta}$ have the values $\underline{\alpha}^{(\ell)}$ and $\underline{\beta}^{(\ell)}$ respectively. We also recall that we take the view that $\underline{\alpha}^{(\ell)}$ is calculated before $\underline{\beta}^{(\ell)}$, so we deduce from expression (4.16) that $\underline{\beta}^{(\ell)}$ is the vector of variables that minimizes the function

$$(4.17)\qquad \sigma\,\check{\underline{c}}^T\underline{\beta} - \sum_{k=1}^{m} \lambda_k^{(\ell)} \log[\,\sigma(\check{\underline{a}}_k^T\underline{\beta}+\underline{a}_k^TA\,\underline{\alpha}^{(\ell)}-b_k)+1\,], \quad \underline{\beta}\in\check{\mathcal{S}}_1^{(\ell)},$$

where we are using the notation

$$(4.18)\qquad \check{\underline{c}} = Z^T\underline{c} \in \mathcal{R}^{\check{n}} \quad \text{and} \quad \check{\underline{a}}_k = Z^T\underline{a}_k \in \mathcal{R}^{\check{n}}, \quad k=1,2,\ldots,m,$$

and where $\check{\mathcal{S}}_1^{(\ell)}$ is the set of values of $\underline{\beta}$ such that $A\underline{\alpha}^{(\ell)}+Z\underline{\beta}$ is in $\mathcal{S}_1$.

Next we reformulate the calculation of $\underline{\beta}^{(\ell)}$ in a way that takes advantage of the fact that $\mathcal{K}$ is an established set. First, we delete from expression (4.17) the values of k in $\mathcal{K}$, because in this case the choice of Z implies that $\check{\underline{a}}_k$ is zero. We could also delete the term $\check{\underline{c}}^T\underline{\beta}$, since (3.4) and $\mathcal{K}^{(*)}\subset\mathcal{K}$ imply that $\check{\underline{c}}$ is zero, but we retain this term in order to provide more generality in the lemmas that are proved later. Second, letting $\underline{b}$ and $\underline{r}^{(*)}$ be the vectors in $\mathcal{R}^{\hat{n}}$ whose components are $\{b_k : k=1,2,\ldots,\hat{n}\}$ and $\{r_k^{(*)} : k=1,2,\ldots,\hat{n}\}$, we deduce from the conditions (4.1) that the moduli of the components of the vector

$$(4.19)\qquad \underline{\delta}^{(\ell)} = A^T\underline{x}^{(\ell)} - \underline{b} - \underline{r}^{(*)}$$

are at most $w_8w_9^{\ell}$. Therefore we substitute the identity

$$(4.20)\qquad \underline{a}_k^TA\,\underline{\alpha}^{(\ell)} = \underline{a}_k^TA\,(A^TA)^{-1}A^T\underline{x}^{(\ell)} = \underline{a}_k^TA\,(A^TA)^{-1}(\underline{b}+\underline{r}^{(*)}+\underline{\delta}^{(\ell)})$$

into expression (4.17). It follows that $\underline{\beta}^{(\ell)}$ can be calculated by minimizing the function

$$\sigma\,\check{\underline{c}}^T\underline{\beta}-\sum_{k\in\check{\mathcal{K}}}\lambda_k^{(\ell)}\log[\sigma(\check{\underline{a}}_k^T\underline{\beta}-\check{b}_k)+1+\epsilon_k^{(\ell)}],\quad \underline{\beta}\in\check{\mathcal{S}}_1^{(\ell)}, \tag{4.21}$$

where $\check{\mathcal{K}}$ is still the set (4.12) and where $\check{b}_k$ and $\epsilon_k^{(\ell)}$ have the values

$$\check{b}_k=b_k-\underline{a}_k^TA(A^TA)^{-1}(\underline{b}+\underline{r}^{(*)})\quad\text{and}\quad \epsilon_k^{(\ell)}=\sigma\underline{a}_k^TA(A^TA)^{-1}\underline{\delta}^{(\ell)},\quad k\in\check{\mathcal{K}}. \tag{4.22}$$

Our notation has been chosen to show the similarity between expressions (1.4) and (4.21). Indeed, the main difference is that the original shifts of 1 have been replaced by shifts of $1+\epsilon_k^{(\ell)}$, and we see that the perturbations satisfy the bounds

$$|\epsilon_k^{(\ell)}|\le w_{10}w_9^{\ell},\quad k\in\check{\mathcal{K}},\quad \ell=1,2,3,\ldots, \tag{4.23}$$

where w_{10} is a positive constant. On the other hand, the definitions (4.18) and (4.22) cause the quantities $\check{\underline{c}}$, $\{\check{\underline{a}}_k : k\in\check{\mathcal{K}}\}$ and $\{\check{b}_k : k\in\check{\mathcal{K}}\}$ to be independent of ℓ, while the values of $\lambda_k^{(\ell)}$ in expression (4.21) are the same as the ones that occurred originally. Furthermore, because the only changes to the arguments of the log functions are due to new notation, the updating formula (1.8) gives the equation

$$\lambda_k^{(\ell+1)}=\lambda_k^{(\ell)}/[\sigma(\check{\underline{a}}_k^T\underline{\beta}^{(\ell)}-\check{b}_k)+1+\epsilon_k^{(\ell)}],\quad k\in\check{\mathcal{K}}. \tag{4.24}$$

Therefore, apart from the changes in the shifts that tend to zero at an R-linear rate as $\ell\to\infty$, one can take the view that the sequence $\{\underline{\beta}^{(\ell)} : \ell=1,2,3,\ldots\}$ is calculated by applying the algorithm of §1 to the problem

$$\begin{aligned}&\text{Minimize}\quad \check{\underline{c}}^T\underline{\beta}, && \underline{\beta}\in\mathcal{R}^{\check{n}},\\ &\text{subject to}\quad \check{\underline{a}}_k^T\underline{\beta}\ge\check{b}_k, && k\in\check{\mathcal{K}}.\end{aligned} \tag{4.25}$$

We prefer to accommodate the changes in shifts by applying a generalization of the algorithm of §1 to the original linear programming problem instead of working with the notation of the previous two paragraphs. Moreover, it is convenient to assume $w_{10}\le 1$ in condition (4.23), which does not lose generality because we can let the initial value of ℓ be the least positive integer L that satisfies $w_{10}w_9^{L-1}\le 1$. Therefore we study the following algorithm.

GENERALIZED ALGORITHM. *Let the parameters σ and η satisfy $\sigma>0$ and $0\le\eta<1$ and let the components of $\underline{\lambda}^{(1)}\in\mathcal{R}^m$ have any positive values. Then an infinite sequence of iterations is begun, ℓ being the index of the iteration, so $\ell=1$ is set initially. For each ℓ, the vector $\underline{x}^{(\ell)}$ is calculated by minimizing the function*

$$\check{\Phi}^{(\ell)}(\underline{x})=\sigma\,c^T\underline{x}-\sum_{k=1}^{m}\lambda_k^{(\ell)}\log[\sigma(\underline{a}_k^T\underline{x}-b_k)+1+\epsilon_k^{(\ell)}],\quad \underline{x}\in\mathcal{S}_1^{(\ell)}, \tag{4.26}$$

where the real parameters $\{\epsilon_k^{(\ell)} : k=1,2,\ldots,m\}$ can take any values that satisfy the conditions

$$|\epsilon_k^{(\ell)}|\le\eta^{\ell},\quad k=1,2,\ldots,m,\quad \ell=1,2,3,\ldots, \tag{4.27}$$

and where $\mathcal{S}_1^{(\ell)}$ is the set

$$\mathcal{S}_1^{(\ell)}=\{\underline{x} : \sigma(\underline{a}_k^T\underline{x}-b_k)+1+\epsilon_k^{(\ell)}\ge 0,\ k=1,2,\ldots,m\}. \tag{4.28}$$

Then $\underline{\lambda}^{(\ell+1)}$ *is defined by the updating formula*

$$\lambda_k^{(\ell+1)} = \lambda_k^{(\ell)} / [\sigma(\underline{a}_k^T \underline{x}^{(\ell)} - b_k) + 1 + \epsilon_k^{(\ell)}], \quad k = 1, 2, \ldots, m, \tag{4.29}$$

which completes the ℓ*th iteration.*

We see that the sequence $\{\underline{\beta}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ can be calculated by applying this algorithm to the problem (4.25), except that some of the early values of ℓ may be omitted. Therefore, if we can prove that the generalization does not damage the main conclusions of §3, then we will know that the constraint residuals $\{\check{a}_k^T \underline{\beta}^{(\ell)} - \check{b}_k : \ell = 1, 2, 3, \ldots\}$ converge to a nonnegative limit for some $k \in \check{\mathcal{K}}$. Thus we will deduce that $\mathcal{K} \cup \{k\}$ is an "established" set, the details of the argument being given after we have found suitable extensions of our previous theory to the "generalized algorithm." Therefore the problem (4.25) will not be mentioned again until the analogue of Theorem 3.7 has been stated for the generalization.

The analogue of expression (1.9) for the generalized algorithm is the notation

$$\check{\phi}^{(\ell)} = \check{\Phi}^{(\ell)}(\underline{x}^{(\ell)}), \quad \ell = 1, 2, 3, \ldots . \tag{4.30}$$

It is shown below that each new $\underline{x}^{(\ell)}$ is well defined and that $\check{\phi}^{(\ell)}$ satisfies a bound that is a little weaker than the first statement of Lemma 2.2.

LEMMA 4.2. *The function* (4.26) *is strictly convex and has a unique minimizer* $\underline{x}^{(\ell)}$*, which is an interior point of* $\mathcal{S}_1^{(\ell)}$*. Moreover, each of the numbers* (4.30) *has the property*

$$\check{\phi}^{(\ell)} \leq \sigma \underline{c}^T \underline{x}^{(*)} + w_{11} \eta^\ell \sum_{k=1}^m \lambda_k^{(\ell)}, \tag{4.31}$$

where $\underline{x}^{(*)}$ *is any solution of the given linear programming and where* w_{11} *is a positive constant.*

Proof. The set $\mathcal{S}_1^{(\ell)}$ has a nonempty interior, because the feasibility of $\underline{x}^{(*)}$ and the conditions (4.27) imply the strict inequalities

$$\sigma(\underline{a}_k^T \underline{x}^{(*)} - b_k) + 1 + \epsilon_k^{(\ell)} \geq 1 - \eta^\ell > 0, \quad k = 1, 2, \ldots, m, \tag{4.32}$$

for every positive integer ℓ. It follows that the function (4.26) is strictly convex if its second derivative matrix is positive definite at all interior points of $\mathcal{S}_1^{(\ell)}$, which can be confirmed by direct computation. Furthermore, $\underline{x}^{(\ell)}$ is well defined if $\mathcal{S}_1^{(\ell)}$ is bounded, which can be established by the argument of the first paragraph of the proof of Lemma 2.1, the set $\mathcal{S}_0$ being a subset of $\mathcal{S}_1^{(\ell)}$ for each ℓ. Therefore the first sentence of the statement of Lemma 4.2 is true.

The definition of $\check{\phi}^{(\ell)}$ and expression (4.32) provide the relation

$$\begin{aligned} \check{\phi}^{(\ell)} \leq \check{\Phi}^{(\ell)}(\underline{x}^{(*)}) &= \sigma \underline{c}^T \underline{x}^{(*)} - \sum_{k=1}^m \lambda_k^{(\ell)} \log[\sigma(\underline{a}_k^T \underline{x}^{(*)} - b_k) + 1 + \epsilon_k^{(\ell)}] \\ &\leq \sigma \underline{c}^T \underline{x}^{(*)} - \sum_{k=1}^m \lambda_k^{(\ell)} \log(1 - \eta^\ell) < \sigma \underline{c}^T \underline{x}^{(*)} + \sum_{k=1}^m \lambda_k^{(\ell)} \eta^\ell / (1 - \eta^\ell), \end{aligned} \tag{4.33}$$

where the last inequality is an elementary property of the log function. Therefore the choice $w_{11} = 1/(1-\eta)$ completes the proof. □

We require the following analogue of Lemma 2.3.

LEMMA 4.3. *Every iteration of the generalized algorithm satisfies the inequality*

$$\check{\phi}^{(\ell+1)} - \check{\phi}^{(\ell)} \geq w_{12} \sum_{k=1}^{m} \lambda_k^{(\ell+1)} [\sigma(\underline{a}_k^T \underline{x}^{(\ell)} - b_k) + \epsilon_k^{(\ell)}]^2 - w_{13} \eta^\ell \sum_{k=1}^{m} \lambda_k^{(\ell+1)}, \tag{4.34}$$

where w_{12} and w_{13} are positive constants. The algorithm also has the property

$$\sigma \underline{c}^T \hat{\underline{x}}^{(*)} - \check{\phi}^{(\ell+1)} \leq \sum_{k=1}^{m} \lambda_k^{(\ell+1)} \sigma(\underline{a}_k^T \hat{\underline{x}}^{(*)} - b_k) + w_{14} \eta^\ell \sum_{k=1}^{m} \lambda_k^{(\ell+1)}, \ell = 1, 2, 3, \ldots, \tag{4.35}$$

where $\hat{\underline{x}}^{(*)}$ is any solution of the linear programming problem and where w_{14} is another positive constant.

Proof. See the Appendix for the proof of this lemma. □

The definition of the set $\mathcal{K}^{(*)}$ and Lemma 3.1 are the same as before, because they depend on the given linear programming problem and not on the algorithm that is used to solve it. We find, however, that a small change is needed in Lemma 3.2 that takes account of the perturbations $\epsilon_k^{(\ell)}$. Specifically, we compare the sequences $\{\underline{x}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ and $\{\underline{\lambda}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ that are calculated by applying the generalized algorithm to the original linear programming problem with the sequences $\{\hat{\underline{x}}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ and $\{\hat{\underline{\lambda}}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$, say, that occur when the generalized algorithm is applied to the problem (3.10), where $\hat{h}$, $\{\hat{\underline{a}}_k : k = 1, 2, \ldots, m\}$, $\{\hat{b}_k : k = 1, 2, \ldots, m\}$ and $\hat{\mathcal{S}}_0$ are defined as before. The values of σ and η are the same in the two calculations, and again the initial Lagrange multiplier vectors are related by the equation $\hat{\underline{\lambda}}^{(1)} = (\sigma\hat{h}+1)\underline{\lambda}^{(1)}$. The perturbations $\{\epsilon_k^{(\ell)} : k = 1, 2, \ldots, m\}$ of expression (4.26) that were chosen for the original linear programming problem, however, must be replaced by the values

$$\hat{\epsilon}_k^{(\ell)} = (\sigma\hat{h}+1)^{-1} \epsilon_k^{(\ell)}, \quad k = 1, 2, \ldots, m, \tag{4.36}$$

when the new problem is solved. Thus we achieve the following extension of Lemma 3.2.

LEMMA 4.4. *Let the conditions of Lemma 3.2 be satisfied and let the sequences $\{\underline{x}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$, $\{\underline{\lambda}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$, $\{\hat{\underline{x}}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ and $\{\hat{\underline{\lambda}}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ be calculated by the generalized algorithm in the way that has just been described. Then these sequences satisfy* (3.11).

Proof. The old value of η is adequate for the perturbations (4.36), because $|\hat{\epsilon}_k^{(\ell)}|$ is bounded above by $|\epsilon_k^{(\ell)}|$ for every k and ℓ. Furthermore, the choice (4.36) and expression (3.16) imply the identities

$$\sigma(\hat{\underline{a}}_k^T \underline{x} - \hat{b}_k) + 1 + \hat{\epsilon}_k^{(\ell)} = (\sigma\hat{h}+1)^{-1} [\sigma(\underline{a}_k^T \underline{x} - b_k) + 1 + \epsilon_k^{(\ell)}], \quad k = 1, 2, \ldots, m, \tag{4.37}$$

which suggest the replacement of the terms $[\sigma(\hat{\underline{a}}_k^T \underline{y} - \hat{b}_k) + 1]$ and $[\sigma(\underline{a}_k^T \underline{y} - b_k) + 1]$ by $[\sigma(\hat{\underline{a}}_k^T \underline{y} - \hat{b}_k) + 1 + \hat{\epsilon}_k^{(\ell)}]$ and $[\sigma(\underline{a}_k^T \underline{y} - b_k) + 1 + \epsilon_k^{(\ell)}]$, respectively, wherever these terms occur in the proof of Lemma 3.2, the vector $\underline{y}$ being either $\underline{x}$ or $\underline{x}^{(\ell)}$. Thus it is straightforward to establish that Lemma 4.4 is true. □

Of course the purpose of Lemma 4.4 is to allow us to assume without loss of generality that the set $\mathcal{K}^{(*)}$ is nonempty in our analysis of the calculated sequence $\{\underline{x}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$. Next we note that there is no change to Lemma 3.3.

LEMMA 4.5. *The Lagrange multiplier estimates of the generalized algorithm have the property*

$$\lambda_k^{(\ell)} \leq w_{15}\ell, \quad k=1,2,\ldots,m, \quad \ell=1,2,3,\ldots, \tag{4.38}$$

where w_{15} is a positive constant.

Proof. See the Appendix for the proof. □

The theory of §3 employs the fact that the numbers $\{\phi^{(\ell)} : \ell=1,2,3,\ldots\}$ are increasing, but the corresponding sequence $\{\check{\phi}^{(\ell)} : \ell=1,2,3,\ldots\}$ need not be monotonic. Therefore the following lemma identifies another sequence that can replace $\{\check{\phi}^{(\ell)} : \ell=1,2,3,\ldots\}$ in the part of the analysis that requires this monotonicity property.

LEMMA 4.6. *There exist positive constants w_{16} and w_{17} such that the sequence*

$$\bar{\phi}^{(\ell)} = \check{\phi}^{(\ell)} - (w_{16}+w_{17}\ell)\,\eta^{\ell}, \quad \ell=1,2,3,\ldots, \tag{4.39}$$

increases monotonically. Furthermore, its terms are bounded by the condition

$$\bar{\phi}^{(\ell)} \leq \sigma\,\underline{c}^T\hat{\underline{x}}^{(*)}, \quad \ell=1,2,3,\ldots, \tag{4.40}$$

where for convenience we let $\hat{\underline{x}}^{(*)}$ be the solution of the given linear programming problem that occurs in Lemma 3.1. Furthermore, the sequences $\{\bar{\phi}^{(\ell)} : \ell=1,2,3,\ldots\}$ and $\{\check{\phi}^{(\ell)} : \ell=1,2,3,\ldots\}$ are convergent.

Proof. See the Appendix for the proof. □

We can now present the analogues of Lemmas 3.4–3.6 and Theorem 3.7 that complete our analysis of the generalized algorithm.

LEMMA 4.7. *Lemma 3.4 remains true when the sequence $\{\underline{x}^{(\ell)} : \ell=1,2,3,\ldots\}$ is calculated by the generalized algorithm.*

LEMMA 4.8. *There exist constants w_{18}, w_{19}, and w_{20}, satisfying $w_{18}>0$, $0\leq w_{19}<1$ and $w_{20}>0$, such that, if the conditions*

$$|\underline{a}_k^T\underline{x}^{(\ell)}-b_k| \leq w_{18}, \quad k\in\mathcal{K}^{(*)}, \tag{4.41}$$

hold for some positive integer ℓ, then the generalized algorithm gives the inequality

$$\sigma\,\underline{c}^T\hat{\underline{x}}^{(*)} - \check{\phi}^{(\ell+1)} \leq w_{19}\left(\sigma\,\underline{c}^T\hat{\underline{x}}^{(*)}-\check{\phi}^{(\ell)}\right) + w_{20}\,\eta^{\ell}\sum_{k=1}^{m}\lambda_k^{(\ell+1)}. \tag{4.42}$$

LEMMA 4.9. *The positive monotonically decreasing sequence $\{\sigma\,\underline{c}^T\hat{\underline{x}}^{(*)}-\bar{\phi}^{(\ell)} : \ell=1,2,3,\ldots\}$ has the property*

$$\sigma\,\underline{c}^T\hat{\underline{x}}^{(*)} - \bar{\phi}^{(\ell)} \leq w_{21}w_{22}^{\ell}, \quad \ell=1,2,3,\ldots, \tag{4.43}$$

where w_{21} and w_{22} are constants that satisfy $w_{21}>0$ and $0<w_{22}<1$, and where $\bar{\phi}^{(\ell)}$ has the value (4.39).

THEOREM 4.10. *Theorem 3.7 remains true when the vectors of variables $\{\underline{x}^{(\ell)} : \ell=1,2,3,\ldots\}$ are calculated by the generalized algorithm.*

Proofs. See the Appendix for the proofs. □

We now return to the linear programming problem (4.25). Because it has been noted that $\check{c}$ is zero, we apply the construction of Lemma 3.2. Specifically, we let $\hat{h}$ be as large as possible subject to the consistency of the inequalities

$$\check{\underline{a}}_k^T\underline{\beta} \geq \check{b}_k+\hat{h}, \quad k\in\check{\mathcal{K}}, \tag{4.44}$$

for some $\underline{\beta}$ in $\mathcal{R}^{\check{n}}$. Then, instead of expression (3.9), we employ the notation

$$\bar{\underline{a}}_k = (\sigma\hat{h}+1)^{-1}\check{\underline{a}}_k \quad \text{and} \quad \bar{b}_k = (\sigma\hat{h}+1)^{-1}(\check{b}_k+\hat{h}), \quad k\in\check{\mathcal{K}}. \tag{4.45}$$

It follows from Lemma 3.2 that the feasible region of the problem

$$\begin{aligned} &\text{Minimize} \quad \check{\underline{c}}^T\underline{\beta}\ (\equiv 0), \qquad \underline{\beta}\in\mathcal{R}^{\check{n}}, \\ &\text{subject to} \quad \bar{\underline{a}}_k^T\underline{\beta} \geq \bar{b}_k, \qquad k\in\check{\mathcal{K}}, \end{aligned} \tag{4.46}$$

is nonempty but has no interior. Furthermore, we deduce from Lemma 4.4 that the sequence $\{\underline{\beta}^{(\ell)} : \ell=1,2,3,\ldots\}$ can be calculated by applying the generalized algorithm to this new problem, except that we noted soon after expression (4.25) that the initial value of ℓ may have to exceed one. Therefore Theorem 4.10 gives the property

$$\lim_{\ell\to\infty} \bar{\underline{a}}_k^T\underline{\beta}^{(\ell)} = \bar{b}_k, \quad k\in\hat{\mathcal{K}}^{(*)}, \tag{4.47}$$

where k is in $\hat{\mathcal{K}}^{(*)}$ if and only if the conditions (4.44) on $\underline{\beta}$ imply the identity $\check{\underline{a}}_k^T\underline{\beta}=\check{b}_k+\hat{h}$, the set $\hat{\mathcal{K}}^{(*)}$ being nonempty because $\hat{h}$ is as large as possible. Furthermore, expression (4.47) and the definitions (4.45) of $\bar{\underline{a}}_k$ and $\bar{b}_k$ provide the limits

$$\lim_{\ell\to\infty} \check{\underline{a}}_k^T\underline{\beta}^{(\ell)} = \check{b}_k+\hat{h}, \quad k\in\hat{\mathcal{K}}^{(*)}. \tag{4.48}$$

We express this conclusion in terms of the variables of the original linear programming problem.

LEMMA 4.11. *Let $\mathcal{K}$ be an "established" set that does not include all the constraint indices, but that has the property that none of the vectors $\{\underline{a}_k : k\in\check{\mathcal{K}}\}$ is in the space spanned by the constraint gradients $\{\underline{a}_k : k\in\mathcal{K}\}$, where $\check{\mathcal{K}}$ is the set (4.12). Furthermore, let $\hat{h}$ be as large as possible subject to the consistency of the conditions*

$$\begin{aligned} &\underline{a}_k^T\underline{x}-b_k = r_k^{(*)}, \qquad k\in\mathcal{K}, \\ &\underline{a}_k^T\underline{x} \geq b_k+\hat{h}, \qquad k\in\check{\mathcal{K}}, \end{aligned} \tag{4.49}$$

for some $\underline{x}$ in $\mathcal{R}^n$. Given this choice of $\hat{h}$, we let $\mathcal{X}$ be the set of values of $\underline{x}$ that are allowed by expression (4.49). Then there exists a nonempty subset $\hat{\mathcal{K}}^{(*)}$ of $\check{\mathcal{K}}$ such that, for each k in $\hat{\mathcal{K}}^{(*)}$, the inequality $\underline{a}_k^T\underline{x}\geq b_k+\hat{h}$ holds as an equation for every $\underline{x}$ in $\mathcal{X}$. Furthermore, the set $\mathcal{K}\cup\hat{\mathcal{K}}^{(*)}$ is "established."

Proof. It is straightforward to deduce from continuity and compactness that the required choice of $\hat{h}$ can be achieved. Then we suppose that, for each k in $\check{\mathcal{K}}$, there is a feasible vector $\underline{x}_k$ such that $\underline{a}_k^T\underline{x}_k-b_k-\hat{h}$ is positive, and we let $\underline{x}$ be the average of these vectors. Thus we find that every inequality of expression (4.49) can be satisfied strictly by a single admissible value of $\underline{x}$, which implies that $\hat{h}$ can be increased. It follows from this contradiction that $\hat{\mathcal{K}}^{(*)}$ is nonempty.

Next we employ the change of variables that is described in the paragraph that includes (4.13)–(4.15). Specifically, we express the general vector $\underline{x}\in\mathcal{R}^n$ in the form

$$\underline{x} = A\,\underline{\alpha} + Z\underline{\beta}, \tag{4.50}$$

where the matrices A and Z have been defined already, and where $\underline{\alpha}$ and $\underline{\beta}$ are related to $\underline{x}$ by the formulae

$$\underline{\alpha} = (A^TA)^{-1}A^T\underline{x} \in \mathcal{R}^{\hat{n}} \quad \text{and} \quad \underline{\beta} = Z^T\underline{x} \in \mathcal{R}^{\check{n}}. \tag{4.51}$$

Thus, if $\underline{x}$ satisfies the first line of expression (4.49), then we have the identity

$$\underline{r}^{(*)} = A^T\underline{x} - \underline{b} = A^TA\,\underline{\alpha} - \underline{b}, \tag{4.52}$$

which fixes $\underline{\alpha}$, the vectors $\underline{b}$ and $\underline{r}^{(*)}$ being introduced just before (4.19). Then the second line of expression (4.49) gives the conditions

$$\underline{a}_k^T A\,\underline{\alpha} + \underline{a}_k^T Z\underline{\beta} \geq b_k + \hat{h}, \quad k \in \check{\mathcal{K}}, \tag{4.53}$$

on $\underline{\beta}$. Furthermore, because the relation (4.52) shows that $\underline{\alpha}$ can be replaced by the vector $(A^TA)^{-1}(\underline{b}+\underline{r}^{(*)})$, we can write the inequalities (4.53) in the form (4.44), where we are recalling the notation (4.18) and (4.22). It follows that the value of $\hat{h}$ in the conditions (4.49) is the same as the one that occurs in the paragraph that includes expressions (4.44)–(4.48). Furthermore, the sets $\hat{\mathcal{K}}^{(*)}$ in that paragraph and in the statement of Lemma 4.11 are the same.

Therefore the limit (4.48) is achieved when the statement of Lemma 4.11 defines $\hat{h}$ and $\hat{\mathcal{K}}^{(*)}$. Remembering also that the moduli of the components of the vector (4.19) are at most $w_8 w_9^{\ell}$, we deduce that (4.14), (4.15), (4.19), (4.18), (4.48), and (4.22) imply the limit

$$\begin{aligned}
\lim_{\ell\to\infty} \underline{a}_k^T\underline{x}^{(\ell)} &= \lim_{\ell\to\infty} \left[\underline{a}_k^T A\,(A^TA)^{-1}A^T\underline{x}^{(\ell)} + \underline{a}_k^T Z\underline{\beta}^{(\ell)}\right] \\
&= \lim_{\ell\to\infty} \left[\underline{a}_k^T A\,(A^TA)^{-1}(\underline{\delta}^{(\ell)} + \underline{b} + \underline{r}^{(*)}) + \check{\underline{a}}_k^T\underline{\beta}^{(\ell)}\right] \\
&= \underline{a}_k^T A\,(A^TA)^{-1}(\underline{b} + \underline{r}^{(*)}) + \check{b}_k + \hat{h} = b_k + \hat{h}, \quad k \in \hat{\mathcal{K}}^{(*)}.
\end{aligned} \tag{4.54}$$

Moreover, a comparison of expressions (4.2) and (4.49) shows that $\hat{h}$ is nonnegative. It follows that the first condition for an "established" set is preserved if the elements of $\hat{\mathcal{K}}^{(*)}$ are included in $\mathcal{K}$, the new values of $r_k^{(*)}$ being equal to $\hat{h}$.

Turning to the second condition, we have noted already that the components of $\underline{\delta}^{(\ell)}$ tend to zero at an R-linear rate as $\ell \to \infty$. Moreover, Theorem 4.10 implies that the convergence rates of expressions (4.47) and (4.48) are also R-linear. Therefore the limit (4.54) enjoys this rate of convergence too. Consequently, condition (4.1) allows $\mathcal{K}$ to be augmented by $\hat{\mathcal{K}}^{(*)}$, provided that the numbers w_8 and w_9 are increased if necessary.

Finally, we let $\underline{\hat{x}}^{(*)}$ in expression (4.2) be any vector $\underline{x}$ that satisfies the constraints (4.49). Remembering that the definition of $\hat{\mathcal{K}}^{(*)}$ in the statement of Lemma 4.11 provides the equations

$$\underline{a}_k^T\underline{\hat{x}}^{(*)} = b_k + \hat{h} = b_k + r_k^{(*)}, \quad k \in \hat{\mathcal{K}}^{(*)}, \tag{4.55}$$

and that $\hat{h}$ is nonnegative, it follows that the third condition for $\mathcal{K}\cup\hat{\mathcal{K}}^{(*)}$ to be "established" is also obtained, which completes the proof. □

We deduce from Lemmas 4.1 and 4.11 that the set $\{1, 2, \ldots, m\}$ of all the constraint indices can be "established." Indeed, as suggested just before the statement of Lemma 4.1, we apply a recursive procedure that begins with the "established" set $\mathcal{K}$ that is identified in the second or third paragraph of this section. Then each step of the recursion enlarges $\mathcal{K}$ by applying Lemma 4.1 or 4.11 until every constraint index is in $\mathcal{K}$. It follows that expression (4.6) is true. Furthermore, in view of the remarks that are made just after that expression, we now know that the sequence

$\{\underline{x}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ converges to a solution of the given linear programming problem. This important result will be stated as a theorem in §5 so that it is available to those readers who skip §4.

5. Conclusions. We let $\hat{\underline{x}}^{(*)}$ be the following solution of the original linear programming problem. If there is only one solution, then the problem defines $\hat{\underline{x}}^{(*)}$ uniquely. Otherwise, the definition of $\mathcal{K}^{(*)}$ in the first paragraph of §3 gives the equations

$$\underline{a}_k^T \hat{\underline{x}}^{(*)} = b_k, \quad k \in \mathcal{K}^{(*)}. \tag{5.1}$$

Furthermore, we let $\check{\mathcal{K}}_1$ be the subset of $\{1, 2, \ldots, m\}$ such that j is in $\check{\mathcal{K}}_1$ if and only if these equations do not fix the value of $\underline{a}_j^T \hat{\underline{x}}^{(*)}$. In other words, $\underline{a}_j$ is not in the space that is spanned by the vectors $\{\underline{a}_k : k \in \mathcal{K}^{(*)}\}$. Then, after satisfying expression (5.1), we take up (some of) the remaining freedom in $\hat{\underline{x}}^{(*)}$ by maximizing the least of the residuals $\{\underline{a}_j^T \hat{\underline{x}}^{(*)} - b_j : j \in \check{\mathcal{K}}_1\}$, which is the condition

$$\min\{\underline{a}_j^T \hat{\underline{x}}^{(*)} - b_j : j \in \check{\mathcal{K}}_i\} = \max_{\underline{x} \in \check{\mathcal{S}}_i} \min\{\underline{a}_j^T \underline{x} - b_j : j \in \check{\mathcal{K}}_i\}, \tag{5.2}$$

where $i = 1$ and where $\check{\mathcal{S}}_i$ contains those vectors in $\mathcal{R}^n$ that achieve the conditions on $\hat{\underline{x}}^{(*)}$ that have been imposed already. If there is still some freedom in $\hat{\underline{x}}^{(*)}$, we let $\check{\mathcal{K}}_{i+1}$ be the subset of $\{1, 2, \ldots, m\}$ such that j is in $\check{\mathcal{K}}_{i+1}$ if and only if the constraints on $\hat{\underline{x}}^{(*)}$ so far do not fix the value of $\underline{a}_j^T \hat{\underline{x}}^{(*)}$. Then we require $\hat{\underline{x}}^{(*)}$ to satisfy (5.2) after increasing i by 1. We continue this procedure recursively, increasing i at each stage, until there is no freedom in $\hat{\underline{x}}^{(*)}$. The procedure terminates, because the argument in the first paragraph of the proof of Lemma 4.11 provides the strict inequality $|\check{\mathcal{K}}_{i+1}| < |\check{\mathcal{K}}_i|$, where $|\check{\mathcal{K}}_i|$ is the number of elements in $\check{\mathcal{K}}_i$. It is important that this definition of $\hat{\underline{x}}^{(*)}$ is independent of the parameters σ and $\underline{\lambda}^{(1)}$, because $\hat{\underline{x}}^{(*)}$ has the following fundamental property.

THEOREM 5.1. *Let the algorithm of* §1 *be applied to minimize the function* (1.1) *subject to the constraints* (1.2), *where the feasible region is bounded and nonempty. Then the sequence* $\{\underline{x}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ *converges to the solution* $\hat{\underline{x}}^{(*)}$ *of the linear programming problem that has just been defined.*

Proof. The argument in the penultimate paragraph of §3 proves the theorem in the case when the linear programming problem has a unique solution. In the alternative case, we employ the analysis of §4. Specifically, we take the vector $\hat{\underline{x}}^{(*)}$ in expression (4.2) from the statement of Theorem 5.1, because this choice is allowed by the initial "established" set $\mathcal{K}$ that is specified in §4, and because there is no need to change $\hat{\underline{x}}^{(*)}$ when Lemmas 4.1 and 4.11 are applied. Therefore the required result follows from the last paragraph of §4 and from the paragraph that precedes the statement of Lemma 4.1. □

The limit of the sequence $\{\underline{\lambda}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ is also useful. Indeed, having proved Theorems 3.8 and 5.1, we know from Lemma 2.4 that the limit $\underline{\lambda}^{(*)}$ is a Lagrange multiplier vector of the KKT conditions at $\hat{\underline{x}}^{(*)}$. This vector, however, depends on the initial parameters of the algorithm, namely, σ and $\underline{\lambda}^{(1)}$, if several values of Lagrange multipliers are allowed by the KKT conditions. In particular, in the highly degenerate case when only one vector $\underline{x}$ satisfies the constraints (1.2), and when all of these constraints hold as equations, the conditions (1.7) admit several choices of $\underline{\lambda}^{(*)}$ whose components are all positive, and, if $\underline{\lambda}^{(1)}$ is set to any of these

choices, then the function (1.4) has the property $\underline{\nabla}\Phi^{(1)}(\underline{\hat{x}}^{(*)})=0$. Thus the algorithm picks the value $\underline{x}^{(1)}=\underline{\hat{x}}^{(*)}$. It follows (by induction for $\ell\geq 2$) that every iteration sets $\underline{\lambda}^{(\ell+1)}=\underline{\lambda}^{(\ell)}$, so $\underline{\lambda}^{(1)}$ is the limit of the sequence $\{\underline{\lambda}^{(\ell)}:\ell=1,2,3,\ldots\}$.

Next we consider some rates of convergence of the algorithm, keeping in mind that it would be usual to end the calculation on the ℓth iteration if an upper bound on $\underline{c}^T\underline{x}^{(\ell)}-\underline{c}^T\underline{\hat{x}}^{(*)}$ were sufficiently small, provided that any constraint violations at $\underline{x}^{(\ell)}$ were tolerable. The sequence of Lagrange multiplier estimates enjoys the following properties.

THEOREM 5.2. *Let the set $\mathcal{K}^{(*)}$ and the vector $\underline{\hat{x}}^{(*)}$ be defined by the first paragraphs of §§3 and 5, respectively, and let the conditions of Theorem* 5.1 *hold. Then, for each k in $\mathcal{K}^{(*)}$, the sequence $\{\lambda_k^{(\ell)}:\ell=1,2,3,\ldots\}$ tends to a positive limit. Alternatively, if k is a constraint index that is not in $\mathcal{K}^{(*)}$, then its Lagrange multiplier estimates converge to zero at the Q-linear rate*

$$\lim_{\ell\to\infty}\lambda_k^{(\ell+1)}/\lambda_k^{(\ell)}=[\sigma(\underline{a}_k^T\underline{\hat{x}}^{(*)}-b_k)+1]^{-1},\quad k\notin\mathcal{K}^{(*)}. \tag{5.3}$$

Proof. The first statement of Theorem 5.2 is taken from Theorem 3.8. The other statement is an immediate consequence of Theorem 5.1 and formula (1.8). □

We derive a bound on $\underline{c}^T\underline{x}^{(\ell)}-\underline{c}^T\underline{\hat{x}}^{(*)}$ from (2.11) and the feasibility of $\underline{\hat{x}}^{(*)}$. Specifically, we find the condition

$$\underline{c}^T\underline{x}^{(\ell)}-\underline{c}^T\underline{\hat{x}}^{(*)}=\sum_{k=1}^m\lambda_k^{(\ell+1)}(\underline{a}_k^T\underline{x}^{(\ell)}-\underline{a}_k^T\underline{\hat{x}}^{(*)})\leq\sum_{k=1}^m\lambda_k^{(\ell+1)}(\underline{a}_k^T\underline{x}^{(\ell)}-b_k), \tag{5.4}$$

which is useful because it is straightforward to calculate the right hand side for each ℓ. Furthermore, we write this condition in the form

$$\underline{c}^T\underline{x}^{(\ell)}-\underline{c}^T\underline{\hat{x}}^{(*)}\leq\sum_{j\in\mathcal{K}^{(*)}}\lambda_j^{(\ell+1)}(\underline{a}_j^T\underline{x}^{(\ell)}-b_j)+\sum_{k\notin\mathcal{K}^{(*)}}\lambda_k^{(\ell+1)}(\underline{a}_k^T\underline{x}^{(\ell)}-b_k), \tag{5.5}$$

in order to explain that the right-hand side tends to zero as $\ell\to\infty$. Indeed, the boundedness of the Lagrange multiplier estimates and the limits $\{\underline{a}_j^T\underline{x}^{(\ell)}\to\underline{a}_j^T\underline{\hat{x}}^{(*)}=b_j:j\in\mathcal{K}^{(*)}\}$ imply the convergence to zero of the first sum of expression (5.5), while, remembering that each $\underline{x}^{(\ell)}$ is confined to the bounded set $\mathcal{S}_1$, the other sum also has this property due to the assertion $\{\lambda_k^{(\ell+1)}\to 0:k\notin\mathcal{K}^{(*)}\}$ of Theorem 5.2.

When inequality (5.5) is used in practice, the rates of convergence to zero of the sequences $\{\underline{a}_j^T\underline{x}^{(\ell)}-b_j:\ell=1,2,3,\ldots\}$ and $\{\lambda_k^{(\ell+1)}:\ell=1,2,3,\ldots\}$ for $j\in\mathcal{K}^{(*)}$ and $k\notin\mathcal{K}^{(*)}$ are of interest. The latter case is answered by (5.3). Moreover, inequality (2.9) is relevant to the former sequence, because, with the help of Lemma 2.2 and expression (2.10), it gives the conditions

$$\begin{aligned}(\underline{a}_j^T\underline{x}^{(\ell)}-b_j)^2&\leq(w_1\lambda_j^{(\ell+1)}\sigma^2)^{-1}(\phi^{(\ell+1)}-\phi^{(\ell)})\leq(w_1\lambda_j^{(\ell+1)}\sigma^2)^{-1}(\sigma\,\underline{c}^T\underline{\hat{x}}^{(*)}-\phi^{(\ell)})\\&\leq(w_1\lambda_j^{(\ell+1)}\sigma)^{-1}\sum_{k=1}^m\lambda_k^{(\ell)}(\underline{a}_k^T\underline{\hat{x}}^{(*)}-b_k),\quad\ell=2,3,4,\ldots\ .\end{aligned} \tag{5.6}$$

Thus, remembering that all of the numbers $\{\underline{a}_k^T\underline{\hat{x}}^{(*)}-b_k:k\in\mathcal{K}^{(*)}\}$ are zero, we deduce the following bounds on the residuals of the constraints that are satisfied as equations at $\underline{\hat{x}}^{(*)}$.

THEOREM 5.3. *If the set $\mathcal{K}^{(*)}$ is nonempty, then, after the first iteration, the algorithm of* §1 *has the property*

$$|\underline{a}_j^T \underline{x}^{(\ell)} - b_j| \leq (\bar{\rho}^{(\ell)})^{1/2}, \quad j \in \mathcal{K}^{(*)}, \tag{5.7}$$

where $\bar{\rho}^{(\ell)}$ is the number

$$\bar{\rho}^{(\ell)} = \Big[w_1 \sigma \min\{\lambda_j^{(\ell+1)} : j \in \mathcal{K}^{(*)}\} \Big]^{-1} \sum_{k=1}^{m} \lambda_k^{(\ell)} (\underline{a}_k^T \hat{\underline{x}}^{(*)} - b_k), \quad \ell = 2, 3, 4, \ldots, \tag{5.8}$$

which is zero in the degenerate case $\mathcal{K}^{(*)} = \{1, 2, \ldots, m\}$. Alternatively, if some of the constraint indices are not in $\mathcal{K}^{(*)}$, then each $\bar{\rho}^{(\ell)}$ is positive and the sequence $\{\bar{\rho}^{(\ell)} : \ell = 2, 3, 4, \ldots\}$ tends to zero at the Q-linear rate

$$\lim_{\ell \to \infty} \bar{\rho}^{(\ell+1)} / \bar{\rho}^{(\ell)} = \max\{ [\sigma(\underline{a}_k^T \hat{\underline{x}}^{(*)} - b_k) + 1]^{-1} : k \notin \mathcal{K}^{(*)}\}. \tag{5.9}$$

Proof. Inequality (5.7) is an elementary consequence of the bound (5.6). Furthermore, in view of the conditions

$$\underline{a}_k^T \hat{\underline{x}}^{(*)} = b_k, \quad k \in \mathcal{K}^{(*)}, \quad \text{and} \quad \underline{a}_k^T \hat{\underline{x}}^{(*)} > b_k, \quad k \notin \mathcal{K}^{(*)}, \tag{5.10}$$

we have $\bar{\rho}^{(2)} = 0$ if all the constraint indices are in $\mathcal{K}^{(*)}$, so in this case we deduce from expression (5.7) that the algorithm sets $\underline{x}^{(2)}$ to the only feasible vector of variables. Otherwise, because each of the vectors $\{\underline{\lambda}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ has positive components, we see that the numbers (5.8) are positive. Furthermore, because Theorem 5.2 asserts that the sequence $\{\lambda_j^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ tends to a positive limit for each j in $\mathcal{K}^{(*)}$, the term inside the square brackets of (5.8) also tends to a positive limit, so we have the relation

$$\lim_{\ell \to \infty} \frac{\bar{\rho}^{(\ell+1)}}{\bar{\rho}^{(\ell)}} = \lim_{\ell \to \infty} \frac{\sum_{k=1}^{m} \lambda_k^{(\ell+1)} (\underline{a}_k^T \hat{\underline{x}}^{(*)} - b_k)}{\sum_{k=1}^{m} \lambda_k^{(\ell)} (\underline{a}_k^T \hat{\underline{x}}^{(*)} - b_k)} = \lim_{\ell \to \infty} \frac{\sum_{k \notin \mathcal{K}^{(*)}} \lambda_k^{(\ell+1)} (\underline{a}_k^T \hat{\underline{x}}^{(*)} - b_k)}{\sum_{k \notin \mathcal{K}^{(*)}} \lambda_k^{(\ell)} (\underline{a}_k^T \hat{\underline{x}}^{(*)} - b_k)}, \tag{5.11}$$

the last equation being a consequence of the first part of expression (5.10).

Let τ be the right-hand side of (5.9) and let $\bar{\mathcal{K}}$ be the set

$$\bar{\mathcal{K}} = \{k : [\sigma(\underline{a}_k^T \hat{\underline{x}}^{(*)} - b_k) + 1]^{-1} = \tau\}, \tag{5.12}$$

which is a nonempty subset of $\{1, 2, \ldots, m\} \backslash \mathcal{K}^{(*)}$. It follows from the limits (5.3) that, for $j \in \bar{\mathcal{K}}$ and $k \in \{1, 2, \ldots, m\} \backslash \{\mathcal{K}^{(*)} \cup \bar{\mathcal{K}}\}$, the ratio $\lambda_k^{(\ell)} / \lambda_j^{(\ell)}$ tends to zero as $\ell \to \infty$. Therefore (5.11) and the definition (5.12) give the identities

$$\lim_{\ell \to \infty} \frac{\bar{\rho}^{(\ell+1)}}{\bar{\rho}^{(\ell)}} = \lim_{\ell \to \infty} \frac{\sum_{k \in \bar{\mathcal{K}}} \lambda_k^{(\ell+1)} (\underline{a}_k^T \hat{\underline{x}}^{(*)} - b_k)}{\sum_{k \in \bar{\mathcal{K}}} \lambda_k^{(\ell)} (\underline{a}_k^T \hat{\underline{x}}^{(*)} - b_k)} = \lim_{\ell \to \infty} \frac{\sum_{k \in \bar{\mathcal{K}}} \lambda_k^{(\ell+1)}}{\sum_{k \in \bar{\mathcal{K}}} \lambda_k^{(\ell)}}. \tag{5.13}$$

Now, according to Theorem 5.2, each of the ratios $\{\lambda_k^{(\ell+1)} / \lambda_k^{(\ell)} : k \in \bar{\mathcal{K}}\}$ tends to τ as $\ell \to \infty$. Hence we deduce from the positivity of the Lagrange multiplier estimates that the right-hand side of expression (5.13) is equal to τ. Thus this expression establishes the property (5.9), which completes the proof. □

The limits (5.3) and (5.9) are particularly useful because $\hat{\underline{x}}^{(*)}$ is independent of σ. It follows that the asymptotic rates of convergence of the sequences $\{\lambda_k^{(\ell)} : \ell = 1,2,3,\ldots\}$ and $\{\underline{a}_j^T \underline{x}^{(\ell)} - b_j : \ell = 1,2,3,\ldots\}$ for $k \notin \mathcal{K}^{(*)}$ and $j \in \mathcal{K}^{(*)}$ can be made arbitrarily fast by choosing a sufficiently large constant value of σ. Therefore the convergence rate of the bound (5.5) on $\underline{c}^T \underline{x}^{(\ell)} - \underline{c}^T \hat{\underline{x}}^{(*)}$ as $\ell \to \infty$ can be made arbitrarily fast too. Furthermore, this property is also enjoyed by the maximum constraint violation at $\underline{x}^{(\ell)}$. Indeed, Theorem 5.3 proves this assertion for the constraint indices in $\mathcal{K}^{(*)}$. Alternatively, if $k \notin \mathcal{K}^{(*)}$, then the second part of expression (5.10) and the limit $\underline{x}^{(\ell)} \to \hat{\underline{x}}^{(*)}$ imply that the constraint $\underline{a}_k^T \underline{x}^{(\ell)} \geq b_k$ is satisfied strictly for all sufficiently large ℓ.

Several researchers, including one of the referees, have suggested that the given deductions may remain valid if we replace the assumption that the feasible region is bounded by the weaker condition that the set of solutions of the linear programming problem is compact, but I have not tried to prove this conjecture. Moreover, the same referee questioned the availability of $\underline{x}^{(\ell)}$, because in general the minimizer of the function (1.4) cannot be computed exactly. Further consideration of the generalized algorithm that includes expressions (4.26)–(4.29) could allow some freedom in each $\underline{x}^{(\ell)}$, but it would be more useful to develop a suitable termination condition for the case when the minimization of $\{\Phi^{(\ell)}(\underline{x}) : \underline{x} \in \mathcal{S}_1\}$ is done by a Newton–Raphson algorithm with a practical line search.

The numerical results of Jensen, Polyak, and Schneur (1992) show that the modified log barrier method is highly suitable for the solution of many linear programming problems, but the version of §1 has the disadvantage that occasionally no single value of σ gives good efficiency throughout the calculation. This difficulty is particularly severe in the semi-infinite programming problems that are studied by Powell (1992), because several of the constraint residuals $\{\underline{a}_k^T \underline{x} - b_k : k \notin \mathcal{K}^{(*)}\}$ are very close to zero at $\underline{x} = \hat{\underline{x}}^{(*)}$. In this case expressions (5.3) and (5.9) suggest correctly that a large value of σ is needed for a good final rate of convergence, but such values introduce the ill-conditioning problems that are mentioned in the second paragraph of §1, which can increase greatly the amount of work of the early iterations. Therefore Powell (1992) also investigates numerically an extension to the algorithm that increases σ automatically, finding that the extended version is much faster than the original one. The details of the extension are complicated by the fact that, because $\underline{x}^{(\ell)}$ is used as a starting approximation in the calculation of $\underline{x}^{(\ell+1)}$, it is necessary for the numbers $\{\sigma(\underline{a}_k^T \underline{x}^{(\ell)} - b_k) + 1 : k = 1,2,\ldots,m\}$ to remain positive if σ is increased. Thus there is a constraint on any new value of σ in the usual case when $\underline{x}^{(\ell)}$ violates some of the constraints (1.2). On the other hand, this difficulty does not occur in the unmodified log barrier method that is mentioned at the beginning of §1.

Therefore our analysis should be regarded just as a contribution to knowledge that may assist some future research. Indeed, although the algorithm of §1 has some highly interesting properties, the author does not know of any linear programming calculations that it solves more efficiently than all other methods. Perhaps some practical advantages will be derived from the fact that there is no need for the initial vector of variables to be a strictly interior point of the feasible region. This question is studied by Freund (1991), who develops a "shifted log barrier method" whose σ parameter tends to infinity. Moreover, our algorithm can be applied when the constraints (1.2) are inconsistent, provided that the set $\mathcal{S}_1$ of expression (1.4) has a nonempty interior. In this case we conjecture that the calculated sequence $\{\underline{x}^{(\ell)} : \ell = 1,2,3,\ldots\}$ will converge to a limit $\hat{\underline{x}}^{(*)}$ at which the greatest constraint violation is

minimized. Moreover, some of our theory may be relevant to new work on modified log barrier methods for nonlinear constraints.

Appendix.

Proof of Lemma 4.3. We recall that $\underline{x}^{(\ell)}$ minimizes the function (4.26). Therefore $\underline{\nabla}\check{\Phi}^{(\ell)}(\underline{x}^{(\ell)})$ is zero, which gives the equation

$$\underline{c}=\sum_{k=1}^{m}\lambda_k^{(\ell)}\,[\,\sigma(\underline{a}_k^T\underline{x}^{(\ell)}-b_k)+1+\epsilon_k^{(\ell)}\,]^{-1}\underline{a}_k=\sum_{k=1}^{m}\lambda_k^{(\ell+1)}\underline{a}_k, \tag{A.1}$$

the last assertion being due to the updating formula (4.29). Moreover, we now let $\hat{\underline{x}}^{(\ell+1)}$ minimize the strictly convex function

$$\hat{\Phi}^{(\ell+1)}(\underline{x})=\sigma\,\underline{c}^T\underline{x}-\sum_{k=1}^{m}\lambda_k^{(\ell+1)}\log[\,\sigma(\underline{a}_k^T\underline{x}-b_k)+1+\epsilon_k^{(\ell)}\,],\quad \underline{x}\in\mathcal{S}_1^{(\ell)}. \tag{A.2}$$

Thus the analogue of expression (2.12) is the inequality

$$\begin{aligned}\hat{\Phi}^{(\ell+1)}(\hat{\underline{x}}^{(\ell+1)})&\geq\sigma\,\underline{c}^T\hat{\underline{x}}^{(\ell+1)}-\sum_{k=1}^{m}\lambda_k^{(\ell+1)}\,[\,\sigma(\underline{a}_k^T\hat{\underline{x}}^{(\ell+1)}-b_k)+\epsilon_k^{(\ell)}\,]\\&=\sum_{k=1}^{m}\lambda_k^{(\ell+1)}(\sigma\,b_k-\epsilon_k^{(\ell)}).\end{aligned} \tag{A.3}$$

Furthermore, remembering the definitions (4.26) and (4.30) of $\check{\Phi}^{(\ell)}$ and $\check{\phi}^{(\ell)}$, and using (A.1) again, we find that the analogue of condition (2.13) is the bound

$$\begin{aligned}&\hat{\Phi}^{(\ell+1)}(\hat{\underline{x}}^{(\ell+1)})-\check{\phi}^{(\ell)}\\&\quad=\hat{\Phi}^{(\ell+1)}(\hat{\underline{x}}^{(\ell+1)})-\sigma\,\underline{c}^T\underline{x}^{(\ell)}+\sum_{k=1}^{m}\lambda_k^{(\ell)}\log[\,\sigma(\underline{a}_k^T\underline{x}^{(\ell)}-b_k)+1+\epsilon_k^{(\ell)}\,]\\&\quad\geq\sum_{k=1}^{m}\lambda_k^{(\ell)}\log[\,\sigma(\underline{a}_k^T\underline{x}^{(\ell)}-b_k)+1+\epsilon_k^{(\ell)}\,]-\sum_{k=1}^{m}\lambda_k^{(\ell+1)}\,[\,\sigma(\underline{a}_k^T\underline{x}^{(\ell)}-b_k)+\epsilon_k^{(\ell)}\,].\end{aligned} \tag{A.4}$$

It follows from the updating formula (4.29) that, instead of expression (2.14), we now have the relation

$$\hat{\Phi}^{(\ell+1)}(\hat{\underline{x}}^{(\ell+1)})-\check{\phi}^{(\ell)}\geq\sum_{k=1}^{m}\lambda_k^{(\ell+1)}\psi\Big(\sigma(\underline{a}_k^T\underline{x}^{(\ell)}-b_k)+\epsilon_k^{(\ell)}\Big), \tag{A.5}$$

where ψ is still the function (2.15), except that we increase the constant $\bar{\theta}$ in order to accommodate the perturbations $\epsilon_k^{(\ell)}$. Thus the derivation of inequality (2.18) gives the condition

$$\hat{\Phi}^{(\ell+1)}(\hat{\underline{x}}^{(\ell+1)})-\check{\phi}^{(\ell)}\geq w_{12}\sum_{k=1}^{m}\lambda_k^{(\ell+1)}\,[\,\sigma(\underline{a}_k^T\underline{x}^{(\ell)}-b_k)+\epsilon_k^{(\ell)}\,]^2, \tag{A.6}$$

where w_{12} has the value $\frac{1}{2}(\bar{\theta}+1)^{-1}$. Therefore the first assertion of Lemma 4.3 is true if $\{\hat{\Phi}^{(\ell+1)}(\hat{\underline{x}}^{(\ell+1)})-\check{\phi}^{(\ell+1)}\}$ is at most a constant multiple of $\eta^\ell\sum_{k=1}^{m}\lambda_k^{(\ell+1)}$.

We obtain a suitable bound on $\{\hat{\Phi}^{(\ell+1)}(\underline{\hat{x}}^{(\ell+1)})-\check{\phi}^{(\ell+1)}\}$ from the value of $\hat{\Phi}^{(\ell+1)}$ at a point that is near to $\underline{x}^{(\ell+1)}$, allowing for the possibility that some of the logarithms in (A.2) may have negative arguments if we set $\underline{x}=\underline{x}^{(\ell+1)}$. Therefore we pick the point

$$\underline{\xi}=[\,2\,\eta^{\ell}\underline{\hat{x}}^{(*)}+(1-\eta^{\ell})\,\underline{x}^{(\ell+1)}\,]\,/\,(1+\eta^{\ell}), \tag{A.7}$$

where $\underline{\hat{x}}^{(*)}$ is any solution of the linear programming problem. Because $\underline{\hat{x}}^{(*)}$ is feasible and because $\underline{x}^{(\ell+1)}$ satisfies the constraints

$$\sigma(\underline{a}_k^T\underline{x}^{(\ell+1)}-b_k)+1+\epsilon_k^{(\ell+1)}>0,\quad k=1,2,\ldots,m, \tag{A.8}$$

the conditions

$$|\epsilon_k^{(\ell)}|\leq\eta^{\ell},\quad k=1,2,\ldots,m,\quad \ell=1,2,3,\ldots, \tag{A.9}$$

given in expression (4.27), imply the inequalities

$$\begin{aligned}\sigma(\underline{a}_k^T\underline{\xi}-b_k)+1+\epsilon_k^{(\ell)}&=\frac{2\,\eta^{\ell}\sigma(\underline{a}_k^T\underline{\hat{x}}^{(*)}-b_k)+(1-\eta^{\ell})\,\sigma(\underline{a}_k^T\underline{x}^{(\ell+1)}-b_k)}{1+\eta^{\ell}}+1+\epsilon_k^{(\ell)}\\&>(1-\eta^{\ell})\,(-1-\epsilon_k^{(\ell+1)})\,(1+\eta^{\ell})^{-1}+1+\epsilon_k^{(\ell)}\\&\geq(1-\eta^{\ell})\,(-1-\eta^{\ell+1})\,(1+\eta^{\ell})^{-1}+1-\eta^{\ell}>0,\quad k=1,2,\ldots,m,\end{aligned} \tag{A.10}$$

where the last assertion is obtained by increasing $\eta^{\ell+1}$ to η^{ℓ}. Therefore the function (A.2) is well defined at $\underline{x}=\underline{\xi}$. Furthermore, the relation

$$\hat{\Phi}^{(\ell+1)}(\underline{\hat{x}}^{(\ell+1)})\leq\hat{\Phi}^{(\ell+1)}(\underline{\xi}) \tag{A.11}$$

is a consequence of the definition of $\underline{\hat{x}}^{(\ell+1)}$.

Next we seek an upper bound on the right-hand side of the identity

$$\begin{aligned}\hat{\Phi}^{(\ell+1)}(\underline{\xi})-\check{\phi}^{(\ell+1)}&=\hat{\Phi}^{(\ell+1)}(\underline{\xi})-\check{\Phi}^{(\ell+1)}(\underline{x}^{(\ell+1)})=\sigma\,\underline{c}^T(\underline{\xi}-\underline{x}^{(\ell+1)})\\&+\sum_{k=1}^{m}\lambda_k^{(\ell+1)}\Big\{\log[\sigma(\underline{a}_k^T\underline{x}^{(\ell+1)}-b_k)+1+\epsilon_k^{(\ell+1)}]-\log[\sigma(\underline{a}_k^T\underline{\xi}-b_k)+1+\epsilon_k^{(\ell)}]\Big\}.\end{aligned} \tag{A.12}$$

Because the definition (A.7) gives the formula

$$\underline{\xi}-\underline{x}^{(\ell+1)}=2\,\eta^{\ell}(\underline{\hat{x}}^{(*)}-\underline{x}^{(\ell+1)})\,/\,(1+\eta^{\ell}), \tag{A.13}$$

the boundedness of the sequence $\{\underline{x}^{(\ell)}:\ell=1,2,3,\ldots\}$ and (A.1) provide the inequality

$$\sigma\,\underline{c}^T(\underline{\xi}-\underline{x}^{(\ell+1)})\leq 2\,\sigma\,\eta^{\ell}\,\|\underline{\hat{x}}^{(*)}-\underline{x}^{(\ell+1)}\|\sum_{k=1}^{m}\lambda_k^{(\ell+1)}\|\underline{a}_k\|\leq w_{23}\,\eta^{\ell}\sum_{k=1}^{m}\lambda_k^{(\ell+1)}, \tag{A.14}$$

where w_{23} is a positive constant. Our treatment of the second line of expression (A.12) depends on the remark that we can restrict attention to the values of k that make positive contributions to the sum. Therefore we assume the condition

$$\begin{aligned}&\sigma(\underline{a}_k^T\underline{\xi}-b_k)+\epsilon_k^{(\ell)}<\sigma(\underline{a}_k^T\underline{x}^{(\ell+1)}-b_k)+\epsilon_k^{(\ell+1)}\\&=(1+\eta^{\ell})\,(1-\eta^{\ell})^{-1}\sigma(\underline{a}_k^T\underline{\xi}-b_k)-2\,\eta^{\ell}\,(1-\eta^{\ell})^{-1}\sigma(\underline{a}_k^T\underline{\hat{x}}^{(*)}-b_k)+\epsilon_k^{(\ell+1)},\end{aligned} \tag{A.15}$$

where the last line is derived from the definition (A.7). Thus, noting that a reformulation of this inequality gives a lower bound on $\sigma(\underline{a}_k^T\underline{\xi}-b_k)$, and recalling the feasibility of $\hat{\underline{x}}^{(*)}$ and the assumption (A.9), we deduce the relation

$$\begin{aligned}\sigma(\underline{a}_k^T\underline{\xi}-b_k)+1+\epsilon_k^{(\ell)} &> \sigma(\underline{a}_k^T\hat{\underline{x}}^{(*)}-b_k)+\tfrac{1}{2}(1-\eta^\ell)\,(\epsilon_k^{(\ell)}-\epsilon_k^{(\ell+1)})\,/\,\eta^\ell+1+\epsilon_k^{(\ell)}\\ &\geq \tfrac{1}{2}(1+\eta^\ell)\,(\epsilon_k^{(\ell)}/\eta^\ell)-\tfrac{1}{2}(1-\eta^\ell)\,(\epsilon_k^{(\ell+1)}/\eta^\ell)+1\\ &\geq -\tfrac{1}{2}(1+\eta^\ell)-\tfrac{1}{2}\eta\,(1-\eta^\ell)+1\geq\tfrac{1}{2}(1-\eta)^2. \qquad \text{(A.16)}\end{aligned}$$

It follows from the elementary property

$$0<\alpha<\beta \quad\Rightarrow\quad \log\beta-\log\alpha<(\beta-\alpha)/\alpha \qquad \text{(A.17)}$$

that, when the term inside the braces of (A.12) is positive, its value is less than the number

$$2\,[\,\sigma\,\underline{a}_k^T(\underline{x}^{(\ell+1)}-\underline{\xi})+\epsilon_k^{(\ell+1)}-\epsilon_k^{(\ell)}\,]\,/\,(1-\eta)^2. \qquad \text{(A.18)}$$

Furthermore, expressions (A.13) and (A.9) show that this number is at most a constant multiple of η^ℓ. Therefore the remarks of this paragraph provide the bound

$$\hat{\Phi}^{(\ell+1)}(\underline{\xi})-\check{\phi}^{(\ell+1)}\leq w_{13}\,\eta^\ell\sum_{k=1}^m\lambda_k^{(\ell+1)}, \qquad \text{(A.19)}$$

where w_{13} is a positive constant. Thus, in view of inequality (A.11) we have the property

$$\hat{\Phi}^{(\ell+1)}(\hat{\underline{x}}^{(\ell+1)})-\check{\phi}^{(\ell+1)}\leq w_{13}\,\eta^\ell\sum_{k=1}^m\lambda_k^{(\ell+1)}. \qquad \text{(A.20)}$$

It follows that the first statement of Lemma 4.3 is a consequence of expression (A.6).

In order to prove the other half of the lemma, we note that expressions (A.1), (A.3), and (A.9) give the condition

$$\begin{aligned}\sigma\,\underline{c}^T\hat{\underline{x}}^{(*)}-\hat{\Phi}^{(\ell+1)}(\hat{\underline{x}}^{(\ell+1)}) &\leq \sigma\sum_{k=1}^m\lambda_k^{(\ell+1)}\,\underline{a}_k^T\hat{\underline{x}}^{(*)}-\sum_{k=1}^m\lambda_k^{(\ell+1)}(\sigma\,b_k-\epsilon_k^{(\ell)})\\ &\leq\sum_{k=1}^m\lambda_k^{(\ell+1)}\,[\,\sigma(\underline{a}_k^T\hat{\underline{x}}^{(*)}-b_k)+\eta^\ell\,]. \qquad \text{(A.21)}\end{aligned}$$

Eliminating $\hat{\Phi}^{(\ell+1)}(\hat{\underline{x}}^{(\ell+1)})$ from the relations (A.20) and (A.21), we find the required inequality (4.35) where w_{14} is the constant $(w_{13}+1)$, which completes the proof. □

Proof of Lemma 4.5. The first statement of Lemma 4.3 and the updating formula (4.29) imply that the analogue of condition (3.22) for the generalized algorithm is the inequality

$$\sum_{k=1}^m\frac{(\lambda_k^{(\ell+1)}-\lambda_k^{(\ell)})^2}{\lambda_k^{(\ell+1)}}\leq w_{12}^{-1}(\check{\phi}^{(\ell+1)}-\check{\phi}^{(\ell)})+w_{12}^{-1}w_{13}\,\eta^\ell\sum_{k=1}^m\lambda_k^{(\ell+1)}. \qquad \text{(A.22)}$$

We continue to employ the notation (3.23), so expression (3.24) remains valid. Furthermore, the last sum of inequality (A.22) is at most $m\,\rho^{(\ell+1)}$. Thus the bound (3.26) can be replaced by the relation

$$(\rho^{(\ell+1)}-\rho^{(\ell)})^2/\,\rho^{(\ell+1)} \leq w_{12}^{-1}(\check{\phi}^{(\ell+1)}-\check{\phi}^{(\ell)}) + w_{12}^{-1}w_{13}\,\eta^{\ell}m\,\rho^{(\ell+1)}, \tag{A.23}$$

which gives the condition

$$\sum_{j=L}^{\ell-1}\frac{(\rho^{(j+1)}-\rho^{(j)})^2}{\rho^{(j+1)}} \leq w_{12}^{-1}(\check{\phi}^{(\ell)}-\check{\phi}^{(L)}) + w_{12}^{-1}w_{13}\,m\sum_{j=L}^{\ell-1}\eta^j\rho^{(j+1)}, \tag{A.24}$$

where L and ℓ are any integers that satisfy $0<L<\ell$.

Next we make use of the bound (4.31) on $\check{\phi}^{(\ell)}$. Thus, in view of the relation

$$\eta^{\ell}\sum_{k=1}^{m}\lambda_k^{(\ell)} \leq \eta^{\ell}m\,\rho^{(\ell)} \leq m\,\eta\sum_{j=L}^{\ell-1}\eta^j\rho^{(j+1)}, \tag{A.25}$$

expression (A.24) implies the inequality

$$\sum_{j=L}^{\ell-1}\frac{(\rho^{(j+1)}-\rho^{(j)})^2}{\rho^{(j+1)}} \leq w_{12}^{-1}(\sigma\,\underline{c}^T\underline{x}^{(*)}-\check{\phi}^{(L)}) + w_{24}\sum_{j=L}^{\ell-1}\eta^j\rho^{(j+1)}, \tag{A.26}$$

where w_{24} is the constant $w_{12}^{-1}m\,(w_{11}\eta+w_{13})$. We now define L to be the least positive integer that satisfies the condition

$$\eta^L \leq (1-\eta)\,(1-\sqrt{\eta})^2/\,w_{24}, \tag{A.27}$$

because this choice allows us to deduce that the right-hand side of expression (A.26) remains finite as $\ell\to\infty$.

Specifically, remembering that $\{\rho^{(j)} : j=1,2,3,\ldots\}$ is a monotonically increasing sequence, the choice (A.27) gives the bound

$$w_{24}\sum_{j=L}^{\ell-1}\eta^j\rho^{(j+1)} < w_{24}\,\eta^L(1+\eta+\eta^2+\cdots)\,\rho^{(\ell)} \leq (1-\sqrt{\eta})^2\rho^{(\ell)}. \tag{A.28}$$

Therefore, by retaining only the last term of the sum on the left-hand side of expression (A.26), we find the condition

$$(\rho^{(\ell)}-\rho^{(\ell-1)})^2 \leq w_{12}^{-1}(\sigma\,\underline{c}^T\underline{x}^{(*)}-\check{\phi}^{(L)})\,\rho^{(\ell)} + [\,(1-\sqrt{\eta})\,\rho^{(\ell)}\,]^2, \quad \ell>L. \tag{A.29}$$

Then we complete the square on the right-hand side by adding a term that is independent of $\rho^{(\ell)}$, which implies the inequality

$$|\,\rho^{(\ell)}-\rho^{(\ell-1)}| \leq (1-\sqrt{\eta})\,(\rho^{(\ell)}+w_{25}), \quad \ell>L, \tag{A.30}$$

where w_{25} is the number

$$w_{25} = \max[\,0,\ \tfrac{1}{2}w_{12}^{-1}(\sigma\,\underline{c}^T\underline{x}^{(*)}-\check{\phi}^{(L)})\,/\,(1-\sqrt{\eta})^2\,]\,. \tag{A.31}$$

Furthermore, since the left-hand side of expression (A.30) is bounded below by the difference $(\rho^{(\ell)}+w_{25})-(\rho^{(\ell-1)}+w_{25})$, we obtain the relation

$$\rho^{(\ell)}+w_{25} \leq (\rho^{(\ell-1)}+w_{25})\,/\,\sqrt{\eta}, \quad \ell>L. \tag{A.32}$$

It follows by induction that we have the property

$$\rho^{(j)} \leq \rho^{(j)}+w_{25} \leq (\rho^{(L)}+w_{25})\,\eta^{-(j-L)/2}, \quad j>L. \tag{A.33}$$

Thus the sum on the right-hand side of expression (A.26) satisfies the bound

$$\sum_{j=L}^{\ell-1} \eta^j \rho^{(j+1)} \leq (\rho^{(L)}+w_{25}) \sum_{j=L}^{\ell-1} \eta^j \eta^{-(j+1-L)/2} < (\rho^{(L)}+w_{25})\,\eta^{L-1/2}/(1-\sqrt{\eta}). \tag{A.34}$$

Therefore, because L is a constant integer, inequality (A.26) provides the condition

$$\sum_{j=1}^{\ell} (\rho^{(j+1)}-\rho^{(j)})^2/\rho^{(j+1)} < w_{26}, \quad \ell=1,2,3,\ldots, \tag{A.35}$$

where w_{26} is another positive constant. Hence, remembering the first line of expression (3.28), we deduce the bound

$$\rho^{(\ell+1)} < 2\,\rho^{(1)} + w_{26}\,\ell < (\rho^{(1)}+w_{26})\,(\ell+1), \quad \ell=1,2,3,\ldots\,. \tag{A.36}$$

Furthermore, this bound is trivial when $\ell=0$. It follows from the definition (3.23) of $\rho^{(\ell)}$ that the required inequality (4.38) is true if w_{15} has the value $(\rho^{(1)}+w_{26})$. □

Proof of Lemma 4.6. Lemmas 4.3 and 4.5 imply the bounds

$$\check{\phi}^{(\ell+1)} \geq \check{\phi}^{(\ell)} - w_{13}\,\eta^{\ell} \sum_{k=1}^{m} \lambda_k^{(\ell+1)} \geq \check{\phi}^{(\ell)} - w_{13}w_{15}\,m\,(\ell+1)\,\eta^{\ell}, \quad \ell=1,2,3,\ldots. \tag{A.37}$$

It follows from the elementary identity

$$(\ell+1)\,\eta^{\ell} = \{[1+(1-\eta)\,\ell]\,\eta^{\ell} - [1+(1-\eta)\,(\ell+1)]\,\eta^{\ell+1}\}\,/(1-\eta)^2 \tag{A.38}$$

that the inequalities

$$\check{\phi}^{(\ell+1)} - [w_{16}+w_{17}(\ell+1)]\,\eta^{\ell+1} \geq \check{\phi}^{(\ell)} - (w_{16}+w_{17}\,\ell)\,\eta^{\ell}, \quad \ell=1,2,3,\ldots, \tag{A.39}$$

are satisfied, where w_{16} and w_{17} have the values

$$w_{16} = w_{13}w_{15}\,m/(1-\eta)^2 \quad \text{and} \quad w_{17} = (1-\eta)\,w_{16}, \tag{A.40}$$

which establishes the first statement of Lemma 4.6. Furthermore, expressions (4.39), (4.31), and (4.38) provide the property

$$\limsup_{\ell\to\infty} \bar{\phi}^{(\ell)} = \limsup_{\ell\to\infty} \check{\phi}^{(\ell)} \leq \limsup_{\ell\to\infty} \left[\sigma\,\underline{c}^T\hat{\underline{x}}^{(*)} + w_{11}\,\eta^{\ell} \sum_{k=1}^{m} \lambda_k^{(\ell)}\right] = \sigma\,\underline{c}^T\hat{\underline{x}}^{(*)}. \tag{A.41}$$

Therefore condition (4.40) is a consequence of the monotonicity of the sequence $\{\bar{\phi}^{(\ell)} : \ell=1,2,3,\ldots\}$. Thus it is elementary that this sequence is convergent. Furthermore, the convergence of $\{\check{\phi}^{(\ell)} : \ell=1,2,3,\ldots\}$ now follows from the remark that, due to the definition (4.39) of $\bar{\phi}^{(\ell)}$, the differences $\{\check{\phi}^{(\ell)}-\bar{\phi}^{(\ell)} : \ell=1,2,3,\ldots\}$ tend to zero as $\ell\to\infty$. The proof is complete. □

Proof of Lemma 4.7. The analogue of expression (3.32) for the generalized algorithm is the inequality

$$\begin{aligned}\log(\Pi^{(\ell+1)}/\Pi^{(\ell)}) &= -\sum_{k\in\mathcal{K}^{(*)}} \mu_k \log[\,\sigma(\underline{a}_k^T\underline{x}^{(\ell)}-b_k)+1+\epsilon_k^{(\ell)}\,] \\ &\geq -\sum_{k\in\mathcal{K}^{(*)}} \mu_k\,[\,\sigma(\underline{a}_k^T\underline{x}^{(\ell)}-b_k)+\epsilon_k^{(\ell)}\,] \\ &\quad + \tfrac{1}{2}(\bar{\theta}+1)^{-2}\sum_{k\in\mathcal{K}^{(*)}} \mu_k\,[\,\sigma(\underline{a}_k^T\underline{x}^{(\ell)}-b_k)+\epsilon_k^{(\ell)}\,]^2, \qquad \text{(A.42)}\end{aligned}$$

where $\Pi^{(\ell)}$ is still the product (3.31) and where we must use the new value of $\bar{\theta}$ that is mentioned just after condition (A.5). Therefore, because the relations (3.33) and (3.34) are still valid, it is suitable to replace expression (3.35) by the bound

$$\begin{aligned}&\sum_{k\in\mathcal{K}^{(*)}} \mu_k\,[\,\sigma(\underline{a}_k^T\underline{x}^{(\ell)}-b_k)+\epsilon_k^{(\ell)}\,]^2 \\ &\leq 2\,(\bar{\theta}+1)^2\left\{\log(\Pi^{(\ell+1)}/\Pi^{(\ell)})+\sum_{k=1}^{m}\lambda_k^{(\ell+1)}[\,\sigma(\underline{a}_k^T\underline{x}^{(\ell)}-b_k)\,]+\sum_{k\in\mathcal{K}^{(*)}}\mu_k\epsilon_k^{(\ell)}\right\}. \qquad \text{(A.43)}\end{aligned}$$

Furthermore, in view of Lemmas 4.3 and 4.5, the analogue of expression (3.36) is the condition

$$\begin{aligned}&\sum_{k=1}^{m}\lambda_k^{(\ell+1)}\,[\,\sigma(\underline{a}_k^T\underline{x}^{(\ell)}-b_k)+\epsilon_k^{(\ell)}\,] \leq \left\{\sum_{k=1}^{m}\lambda_k^{(\ell+1)}\,[\,\sigma(\underline{a}_k^T\underline{x}^{(\ell)}-b_k)+\epsilon_k^{(\ell)}\,]^2\sum_{k=1}^{m}\lambda_k^{(\ell+1)}\right\}^{\frac{1}{2}} \\ &\leq \left\{m\,(w_{15}/w_{12})\,(\ell+1)\left[\check{\phi}^{(\ell+1)}-\check{\phi}^{(\ell)}+w_{13}\,\eta^\ell\sum_{k=1}^{m}\lambda_k^{(\ell+1)}\right]\right\}^{\frac{1}{2}}, \qquad \text{(A.44)}\end{aligned}$$

so it is helpful to replace inequality (3.37) by the relation

$$\begin{aligned}&\sum_{\ell=p}^{q-1}(\ell+1)^{1/2}\left[\check{\phi}^{(\ell+1)}-\check{\phi}^{(\ell)}+w_{13}\,\eta^\ell\sum_{k=1}^{m}\lambda_k^{(\ell+1)}\right]^{\frac{1}{2}} \\ &\leq q\left[\check{\phi}^{(q)}-\check{\phi}^{(p)}+w_{13}\sum_{\ell=p}^{q-1}\left\{\eta^\ell\sum_{k=1}^{m}\lambda_k^{(\ell+1)}\right\}\right]^{\frac{1}{2}} \\ &\leq q\left[\check{\phi}^{(q)}-\check{\phi}^{(p)}+w_{13}w_{15}\,m\,(p+1)\,\eta^p/\,(1-\eta)^2\right]^{\frac{1}{2}}, \qquad \text{(A.45)}\end{aligned}$$

where the last line is derived from the bound $\lambda_k^{(\ell+1)}\leq w_{15}(\ell+1)$, given in Lemma 4.5, and from the identity

$$\sum_{\ell=p}^{\infty}(\ell+1)\,\eta^\ell = \frac{d}{d\eta}\sum_{\ell=p+1}^{\infty}\eta^\ell = \frac{(p+1-\eta p)\,\eta^p}{(1-\eta)^2}. \qquad \text{(A.46)}$$

Let $\tau^{(p)}$ be the positive number

$$\tau^{(p)} = \sup_q\{\check{\phi}^{(q)}-\check{\phi}^{(p)} : q\geq p\} + w_{13}w_{15}\,m\,(p+1)\,\eta^p/\,(1-\eta^2),\quad p=1,2,3,\ldots, \qquad \text{(A.47)}$$

which is well defined because we found in Lemma 4.6 that the sequence $\{\check{\phi}^{(\ell)} : \ell = 1,2,3,\ldots\}$ is convergent. Furthermore, this convergence and $0<\eta<1$ show that the numbers $\{\tau^{(p)} : p=1,2,3,\ldots\}$ tend to zero as $p\to\infty$. Therefore, for any $\epsilon>0$, there exists a fixed integer L that has the property

$$2\,(\bar{\theta}+1)^2\,[\,m\,(w_{15}/w_{12})\,\tau^{(L)}\,]^{1/2} \leq \epsilon, \tag{A.48}$$

which is analogous to inequality (3.39). Moreover, expressions (A.44), (A.45), and (A.47) imply the bound

$$\sum_{\ell=p}^{q-1}\left\{\sum_{k=1}^{m}\lambda_k^{(\ell+1)}\,[\,\sigma(\underline{a}_k^T\underline{x}^{(\ell)}-b_k)+\epsilon_k^{(\ell)}\,]\right\} < q\,[\,m\,(w_{15}/w_{12})\,\tau^{(p)}\,]^{\frac{1}{2}}, \quad p\geq 1. \tag{A.49}$$

It follows from inequality (A.43) that, when ℓ exceeds L, it is suitable to replace expression (3.40) by the condition

$$\begin{aligned}
&\sum_{j=1}^{\ell}\left\{\sum_{k\in\mathcal{K}^{(*)}}\mu_k\,[\,\sigma(\underline{a}_k^T\underline{x}^{(j)}-b_k)+\epsilon_k^{(j)}\,]^2\right\}\\
&\quad\leq 2\,(\bar{\theta}+1)^2\left\{\log(\Pi^{(\ell+1)}/\,\Pi^{(1)})+\sum_{j=1}^{L-1}\left[\sum_{k=1}^{m}\lambda_k^{(j+1)}\,[\,\sigma(\underline{a}_k^T\underline{x}^{(j)}-b_k)\,]\right]\right.\\
&\qquad\left.+\sum_{j=L}^{\ell}\left[\sum_{k=1}^{m}\lambda_k^{(j+1)}\,[\,\sigma(\underline{a}_k^T\underline{x}^{(j)}-b_k)\,]\right]+\sum_{j=1}^{\ell}\sum_{k\in\mathcal{K}^{(*)}}\mu_k\,\epsilon_k^{(j)}\right\}\\
&\quad\leq 2\,(\bar{\theta}+1)^2\left\{\log(\Pi^{(\ell+1)}/\,\Pi^{(1)})+L\,[\,m\,(w_{15}/w_{12})\,\tau^{(1)}\,]^{\frac{1}{2}}\right.\\
&\qquad\left.-\sum_{j=1}^{\ell}\sum_{k=1}^{m}\lambda_k^{(j+1)}\epsilon_k^{(j)}+\sum_{j=1}^{\ell}\sum_{k\in\mathcal{K}^{(*)}}\mu_k\epsilon_k^{(j)}\right\}+(\ell+1)\,\epsilon.
\end{aligned} \tag{A.50}$$

Guided by the proof of Lemma 3.4, we divide both sides of inequality (A.50) by ℓ and then we consider the limit as $\ell\to\infty$. Because the bound (3.41) is valid if we replace w_2 by the constant w_{15} of Lemma 4.5, we argue as before that there is a zero contribution to this limit from the first two terms inside the braces on the right-hand side of expression (A.50). Furthermore, the other two terms inside the braces also enjoy this property, because the right-hand side of the condition

$$-\sum_{j=1}^{\ell}\sum_{k=1}^{m}\lambda_k^{(j+1)}\epsilon_k^{(j)}+\sum_{j=1}^{\ell}\sum_{k\in\mathcal{K}^{(*)}}\mu_k\,\epsilon_k^{(j)} < w_{15}\,m\sum_{j=1}^{\infty}(j+1)\,\eta^j+\sum_{k\in\mathcal{K}^{(*)}}|\mu_k|\sum_{j=1}^{\infty}\eta^j \tag{A.51}$$

is finite. Thus we deduce the bound

$$\lim_{\ell\to\infty}\frac{1}{\ell}\sum_{j=1}^{\ell}\left\{\sum_{k\in\mathcal{K}^{(*)}}\mu_k\,[\,\sigma(\underline{a}_k^T\underline{x}^{(j)}-b_k)+\epsilon_k^{(j)}\,]^2\right\}\leq\epsilon, \tag{A.52}$$

which implies the limit

$$\lim_{\ell\to\infty}\frac{1}{\ell}\sum_{j=1}^{\ell}\left\{\sum_{k\in\mathcal{K}^{(*)}}[\,\underline{a}_k^T\underline{x}^{(j)}-b_k+\sigma^{-1}\epsilon_k^{(j)}\,]^2\right\}=0, \tag{A.53}$$

due to the positivity of the constants σ and $\{\mu_k : k\in\mathcal{K}^{(*)}\}$ and the freedom to make ϵ arbitrarily small. The required result (3.30) now follows from the elementary relation

$$(\underline{a}_k^T\underline{x}^{(j)}-b_k)^2\le 2\,(\underline{a}_k^T\underline{x}^{(j)}-b_k+\sigma^{-1}\epsilon_k^{(j)})^2+2\,(\sigma^{-1}\epsilon_k^{(j)})^2 \tag{A.54}$$

and from the finiteness of the double sum $\sum_{j=1}^{\infty}\sum_{k\in\mathcal{K}^{(*)}}(\epsilon_k^{(j)})^2$. Therefore Lemma 4.7 is true. □

Proof of Lemma 4.8. In the highly degenerate case when all of the constraint indices are in $\mathcal{K}^{(*)}$, the first sum on the right-hand side of inequality (4.35) is zero. It follows that Lemma 4.8 is valid if we let w_{18} be any positive constant and if we pick the values $w_{19}=0$ and $w_{20}=w_{14}$. Therefore for the remainder of the proof we assume that condition (3.5) is available, which provides the number $\hat{h}$.

We take the definitions of $\mathcal{S}^{(*)}$ and $\underline{s}(\underline{x})$ from the paragraph that includes expressions (3.45)–(3.48), and we recall that inequality (3.46) is satisfied, where w_5 is a constant. Therefore the analogue of expression (3.50) is that the conditions

$$|\underline{a}_k^T\underline{x}^{(\ell)}-b_k|\le w_{18},\quad k\in\mathcal{K}^{(*)}, \tag{A.55}$$

given in the statement of Lemma 4.8, imply the bound

$$\left|\sum_{k\notin\mathcal{K}^{(*)}}\lambda_k^{(\ell+1)}\underline{a}_k^T\,[\,\underline{x}^{(\ell)}-\underline{s}(\underline{x}^{(\ell)})\,]\right|\le w_5w_{18}\sum_{k\notin\mathcal{K}^{(*)}}\lambda_k^{(\ell+1)}\|\,\underline{a}_k\|_2. \tag{A.56}$$

Thus, by letting w_{18} have the value (3.51), we preserve the relation (3.49). Consequently, the assertions (3.52) and (3.53) remain valid.

These assertions and condition (A.9) give the property

$$(\tfrac{1}{2}\sigma\,\hat{h}-\eta^{\ell})\sum_{k\notin\mathcal{K}^{(*)}}\lambda_k^{(\ell+1)}\le\sum_{k\notin\mathcal{K}^{(*)}}\lambda_k^{(\ell+1)}\,[\,\sigma(\underline{a}_k^T\underline{x}^{(\ell)}-b_k)+\epsilon_k^{(\ell)}\,]. \tag{A.57}$$

Hence the argument that provided expression (3.55) yields the inequality

$$(\tfrac{1}{2}\sigma\,\hat{h}-\eta^{\ell})^2\sum_{k\notin\mathcal{K}^{(*)}}\lambda_k^{(\ell+1)}\le\sum_{k\notin\mathcal{K}^{(*)}}\lambda_k^{(\ell+1)}\,[\,\sigma(\underline{a}_k^T\underline{x}^{(\ell)}-b_k)+\epsilon_k^{(\ell)}\,]^2,\quad \ell\ge L, \tag{A.58}$$

where L is any fixed positive integer that satisfies $\eta^L<\frac{1}{2}\sigma\,\hat{h}$ in order that $(\frac{1}{2}\sigma\,\hat{h}-\eta^{\ell})$ is positive.

We now invoke the two parts of Lemma 4.3, noting that the relation (4.35) implies the bound

$$\sigma\,w_{27}\sum_{k\notin\mathcal{K}^{(*)}}\lambda_k^{(\ell+1)}\ge\sigma\,\underline{c}^T\underline{\hat{x}}^{(*)}-\check{\phi}^{(\ell+1)}-w_{14}\,\eta^{\ell}\sum_{k=1}^{m}\lambda_k^{(\ell+1)}, \tag{A.59}$$

where w_{27} is the largest of the numbers $\{\underline{a}_k^T\hat{\underline{x}}^{(*)}-b_k : k\notin\mathcal{K}^{(*)}\}$. Thus inequalities (A.58) and (4.34) give the condition

$$\frac{(\frac{1}{2}\sigma\hat{h}-\eta^\ell)^2}{\sigma\, w_{27}}\left[\sigma\underline{c}^T\hat{\underline{x}}^{(*)}-\check{\phi}^{(\ell+1)}-w_{14}\,\eta^\ell\sum_{k=1}^m\lambda_k^{(\ell+1)}\right]\le(\tfrac{1}{2}\sigma\hat{h}-\eta^\ell)^2\sum_{k\notin\mathcal{K}^{(*)}}\lambda_k^{(\ell+1)}$$

$$\text{(A.60)}\quad\le w_{12}^{-1}\left[(\sigma\underline{c}^T\hat{\underline{x}}^{(*)}-\check{\phi}^{(\ell)})-(\sigma\underline{c}^T\hat{\underline{x}}^{(*)}-\check{\phi}^{(\ell+1)})+w_{13}\,\eta^\ell\sum_{k=1}^m\lambda_k^{(\ell+1)}\right],\quad\ell\ge L.$$

After replacing $\frac{1}{2}\sigma\hat{h}-\eta^\ell$ by $\frac{1}{2}\sigma\hat{h}-\eta^L$, we see that this expression can be written in the form

$$\text{(A.61)}\quad\sigma\underline{c}^T\hat{\underline{x}}^{(*)}-\check{\phi}^{(\ell+1)}\le w_{19}\left(\sigma\underline{c}^T\hat{\underline{x}}^{(*)}-\check{\phi}^{(\ell)}\right)+w_{20}\,\eta^\ell\sum_{k=1}^m\lambda_k^{(\ell+1)},\quad\ell\ge L,$$

where w_{19} and w_{20} are the constants

$$\text{(A.62)}\qquad\begin{aligned}w_{19}&=\left[1+w_{12}\,(\tfrac{1}{2}\sigma\hat{h}-\eta^L)^2/(\sigma\,w_{27})\right]^{-1},\\ w_{20}&=\left[w_{13}+w_{12}w_{14}\,(\tfrac{1}{2}\sigma\hat{h}-\eta^L)^2/(\sigma\,w_{27})\right]w_{19},\end{aligned}$$

so we have the property $w_{19}<1$. Thus the required bound (4.42) is satisfied for $\ell\ge L$. Furthermore, we can increase w_{20} if necessary so that this bound is also valid for all smaller values of ℓ, which completes the proof. □

Proof of Lemma 4.9. We pick the integer L in the way that is suggested by the beginning of the proof of Lemma 3.6. Specifically, we require the condition

$$\text{(A.63)}\qquad\sum_{j=1}^{\hat{\ell}}\left\{\sum_{k\in\mathcal{K}^{(*)}}(\underline{a}_k^T\underline{x}^{(j)}-b_k)^2\right\}\le\tfrac{1}{2}\hat{\ell}\,w_{18}^2,\quad\hat{\ell}\ge L,$$

which can be achieved due to Lemma 4.7. Thus inequality (A.55) is satisfied by at least half of the values of ℓ in the interval $[1,\hat{\ell}]$. For these values of ℓ, Lemmas 4.8, 4.6, and 4.5 provide the bound

$$\text{(A.64)}\qquad\sigma\underline{c}^T\hat{\underline{x}}^{(*)}-\bar{\phi}^{(\ell+1)}-[\,w_{16}+w_{17}(\ell+1)\,]\,\eta^\ell\le w_{19}\,(\sigma\underline{c}^T\hat{\underline{x}}^{(*)}-\bar{\phi}^{(\ell)})+w_{15}w_{20}m\,(\ell+1)\eta^\ell,$$

which we write in the form

$$\text{(A.65)}\qquad\sigma\underline{c}^T\hat{\underline{x}}^{(*)}-\bar{\phi}^{(\ell+1)}\le w_{28}\,(\sigma\underline{c}^T\hat{\underline{x}}^{(*)}-\bar{\phi}^{(\ell)})+(w_{29}+w_{30}\,\ell)\,\eta^\ell,$$

where w_{28}, w_{29}, and w_{30} are positive constants, and where w_{28} can have any value that satisfies $w_{28}\ge w_{19}$, which allows us to impose the strict inequalities

$$\text{(A.66)}\qquad\eta<w_{28}<1.$$

Next we make use of the elementary identity

$$\text{(A.67)}\qquad\begin{aligned}&(w_{29}+w_{30}\,\ell)\,\eta^\ell\\ &=w_{28}\,\eta^\ell\left\{\frac{w_{29}+w_{30}\ell}{w_{28}-\eta}+\frac{w_{30}\eta}{(w_{28}-\eta)^2}\right\}-\eta^{\ell+1}\left\{\frac{w_{29}+w_{30}(\ell+1)}{w_{28}-\eta}+\frac{w_{30}\eta}{(w_{28}-\eta)^2}\right\}.\end{aligned}$$

Thus, on the iterations that satisfy the conditions of Lemma 4.8, expression (A.65) gives the inequality

$$\sigma\, \underline{c}^T \hat{\underline{x}}^{(*)} - \dot{\phi}^{(\ell+1)} \leq w_{28} \left(\sigma\, \underline{c}^T \hat{\underline{x}}^{(*)} - \dot{\phi}^{(\ell)} \right), \tag{A.68}$$

where we are introducing the notation

$$\dot{\phi}^{(\ell)} = \bar{\phi}^{(\ell)} - \eta^\ell \left\{ \frac{w_{29} + w_{30}\ell}{w_{28} - \eta} + \frac{w_{30}\eta}{(w_{28}-\eta)^2} \right\}, \quad \ell = 1, 2, 3, \ldots . \tag{A.69}$$

Moreover, because this notation implies the relation

$$\begin{aligned} \dot{\phi}^{(\ell+1)} - \dot{\phi}^{(\ell)} &= \bar{\phi}^{(\ell+1)} - \bar{\phi}^{(\ell)} + \eta^\ell \left\{ \frac{(w_{29} + w_{30}\ell)\,(1-\eta)}{w_{28}-\eta} + \frac{w_{30}\,(1-w_{28})\,\eta}{(w_{28}-\eta)^2} \right\} \\ &> \bar{\phi}^{(\ell+1)} - \bar{\phi}^{(\ell)} \geq 0, \end{aligned} \tag{A.70}$$

where the last assertion is given in Lemma 4.6, the iterations of the generalized algorithm enjoy the monotonicity property

$$\sigma\, \underline{c}^T \hat{\underline{x}}^{(*)} - \dot{\phi}^{(\ell+1)} < \sigma\, \underline{c}^T \hat{\underline{x}}^{(*)} - \dot{\phi}^{(\ell)} \tag{A.71}$$

for all values of ℓ.

The work of the previous paragraph shows that, instead of the bound (3.63), our definition of L provides the inequality

$$\sigma\, \underline{c}^T \hat{\underline{x}}^{(*)} - \dot{\phi}^{(\ell+1)} \leq w_{28}^{\ell/2} \left(\sigma\, \underline{c}^T \hat{\underline{x}}^{(*)} - \dot{\phi}^{(1)} \right), \quad \ell \geq L. \tag{A.72}$$

Therefore, corresponding to the relation (3.64), expressions (A.66) and (A.71) give the conditions

$$\sigma\, \underline{c}^T \hat{\underline{x}}^{(*)} - \dot{\phi}^{(\ell)} \leq w_{28}^{(\ell-L)/2} \left(\sigma\, \underline{c}^T \hat{\underline{x}}^{(*)} - \dot{\phi}^{(1)} \right), \quad \ell = 1, 2, 3, \ldots . \tag{A.73}$$

Now the definition (A.69) is such that $\sigma\, \underline{c}^T \hat{\underline{x}}^{(*)} - \dot{\phi}^{(\ell)}$ is an upper bound on $\sigma\, \underline{c}^T \hat{\underline{x}}^{(*)} - \bar{\phi}^{(\ell)}$. Hence Lemma 4.9 is true when w_{21} and w_{22} have the values

$$w_{21} = w_{28}^{-L/2} \left(\sigma\, \underline{c}^T \hat{\underline{x}}^{(*)} - \dot{\phi}^{(1)} \right) \quad \text{and} \quad w_{22} = w_{28}^{1/2}. \tag{A.74}$$

□

Proof of Theorem 4.10. We recall that the proof of Lemma 4.7 establishes the limit (A.53). Therefore, instead of expression (3.68), we can pick a fixed integer L that provides the condition

$$\sum_{j=1}^{\ell} [\, \sigma(\underline{a}_k^T \underline{x}^{(j)} - b_k) + \epsilon_k^{(j)} \,]^2 \leq [\, \tfrac{1}{2} \log(1/w_{22}) \,]^2 \ell, \quad \ell \geq L, \tag{A.75}$$

where k is any integer from $\mathcal{K}^{(*)}$ and where w_{22} is defined by (A.74). Thus the analogue of expressions (3.69) and (3.70) is the bound

$$\begin{aligned} \log(\lambda_k^{(\ell+1)} / \lambda_k^{(1)}) &= -\sum_{j=1}^{\ell} \log[\, \sigma(\underline{a}_k^T \underline{x}^{(j)} - b_j) + 1 + \epsilon_k^{(j)} \,] \geq -\sum_{j=1}^{\ell} [\, \sigma(\underline{a}_k^T \underline{x}^{(j)} - b_k) + \epsilon_k^{(j)} \,] \\ &\geq -\left[\ell \sum_{j=1}^{\ell} [\, \sigma(\underline{a}_k^T \underline{x}^{(j)} - b_k) + \epsilon_j^{(j)} \,]^2 \right]^{\frac{1}{2}} \geq -\tfrac{1}{2} \ell \log(1/w_{22}), \quad \ell \geq L, \end{aligned} \tag{A.76}$$

which implies the inequality

$$\lambda_k^{(\ell+1)} \geq w_{22}^{\ell/2} \lambda_k^{(1)}, \quad \ell \geq L, \quad k \in \mathcal{K}^{(*)}. \tag{A.77}$$

Next we give further attention to the sequence $\{\bar{\phi}^{(\ell)} : \ell = 1, 2, 3, \ldots\}$ that is introduced in Lemma 4.6. The constants (A.40) provide the equation

$$\begin{aligned} (\bar{\phi}^{(\ell+1)} - \bar{\phi}^{(\ell)}) - (\check{\phi}^{(\ell+1)} - \check{\phi}^{(\ell)}) &= [\, w_{16}\,(1-\eta) + w_{17}\,(\ell - \ell\eta - \eta)\,]\,\eta^{\ell} \\ &= w_{16}\,(1-\eta)^2\,(\ell+1)\,\eta^{\ell} = w_{13} w_{15}\, m\,(\ell+1)\,\eta^{\ell}. \end{aligned} \tag{A.78}$$

Therefore, remembering Lemmas 4.5 and 4.3, we have the relation

$$\begin{aligned} \bar{\phi}^{(\ell+1)} - \bar{\phi}^{(\ell)} &\geq \check{\phi}^{(\ell+1)} - \check{\phi}^{(\ell)} + w_{13}\,\eta^{\ell} \sum_{k=1}^{m} \lambda_k^{(\ell+1)} \\ &\geq w_{12} \sum_{k=1}^{m} \lambda_k^{(\ell+1)}\, [\, \sigma(\underline{a}_k^T \underline{x}^{(\ell)} - b_k) + \epsilon_k^{(\ell)}\,]^2. \end{aligned} \tag{A.79}$$

It follows from conditions (A.77), (4.40), and (4.43) that, for every constraint index k in $\mathcal{K}^{(*)}$, the analogue of expressions (3.72) and (3.73) is the inequality

$$\begin{aligned} [\, \sigma(\underline{a}_k^T \underline{x}^{(\ell)} - b_k) + \epsilon_k^{(\ell)}\,]^2 &\leq (w_{12}\,\lambda_k^{(\ell+1)})^{-1} (\bar{\phi}^{(\ell+1)} - \bar{\phi}^{(\ell)}) \\ &\leq (w_{12} w_{22}^{\ell/2} \lambda_k^{(1)})^{-1} (\sigma \underline{c}^T \hat{\underline{x}}^{(*)} - \bar{\phi}^{(\ell)}) \\ &\leq w_{21}\,(w_{12}\,\lambda_k^{(1)})^{-1} w_{22}^{\ell/2}, \quad \ell \geq L, \end{aligned} \tag{A.80}$$

which gives the bound

$$\sigma\, |\, \underline{a}_k^T \underline{x}^{(\ell)} - b_k| \leq w_{21}^{1/2} (w_{12}\,\lambda_k^{(1)})^{-1/2} w_{22}^{\ell/4} + \eta^{\ell}, \quad \ell \geq L. \tag{A.81}$$

Moreover, the conditions (A.66) and (A.74) imply the relation

$$\eta^{\ell} < w_{28}^{\ell} < w_{28}^{\ell/8} = w_{22}^{\ell/4}. \tag{A.82}$$

Therefore the required property $|\underline{a}_k^T \underline{x}^{(\ell)} - b_k| \leq w_8 w_9^{\ell}$ holds for $\ell \geq L$ if we set $w_9 = w_{22}^{1/4}$ and if w_8 satisfies the constraint

$$w_8 \geq \sigma^{-1}\, [\, w_{21}^{1/2} (w_{12}\,\lambda_k^{(1)})^{-1/2} + 1\,]. \tag{A.83}$$

Again we increase w_8 if necessary to accommodate the values of ℓ that are less than L, so inequality (3.67) is true. Because it provides the limit (3.66), our analysis is complete. □

Acknowledgments. I am very grateful to Roman Polyak for much advice and encouragement and to Dirk Siegel for valuable comments on a draft of this work, which included a way to shorten the proof of the second assertion of Lemma 2.3. I also offer my thanks to an unknown referee for several constructive criticisms and suggestions.

REFERENCES

R. FLETCHER (1987), *Practical Methods of Optimization*, John Wiley and Sons, Chichester.

A. V. FIACCO AND G. P. MCCORMICK (1968), *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley and Sons, New York.

R. M. FREUND (1991), *Theoretical efficiency of a shifted barrier function algorithm for linear programming*, Linear Algebra Appl., 152, pp. 19–41.

P. E. GILL, W. MURRAY, AND M. H. WRIGHT (1981), *Practical Optimization*, Academic Press, London.

A. J. GOLDMAN AND A. W. TUCKER (1956), *Theory of linear programming*, in Linear Inequalities and Related Systems, H.W. Kuhn and A.W. Tucker, eds., Princeton University Press, Princeton, pp. 53–97.

C. C. GONZAGA (1991), *Large step path-following methods for linear programming, Part* 1: *Barrier function method*, SIAM J. Optim., 1, pp. 268–279.

D. L. JENSEN, R. POLYAK, AND R. SCHNEUR (1992), *Numerical experience with modified barrier functions method for linear programming*, Tech. Report, IBM T.J. Watson Research Center, New York.

I. J. LUSTIG, R. E. MARSTEN, AND D. F. SHANNO (1991), *Computational experience with a primal-dual interior point method for linear programming*, Linear Algebra Appl., 152, pp. 191–222.

R. POLYAK (1992), *Modified barrier functions (theory and methods)*, Math. Programming, 54, pp. 177–222.

M. J. D. POWELL (1992), *Log barrier methods for semi-infinite programming calculations*, in Hellenic Research in Mathematics and Informatics '92, E.A. Lipitakis, ed., Hellenic Mathematical Society, Athens, pp. 23–43.

SIAM J. OPTIMIZATION
Vol. 5, No. 4, pp. 740–769, November 1995

002

FAST INTERIOR POINT METHODS FOR BIPARTITE MATCHING*

LOV K. GROVER†

Abstract. In this paper we show that by using path following interior point methods with nonlogarithmic potential functions that vary inversely with the βth power of the distances from the hyperplane (with $\beta = O(\log v)$), it is possible to obtain an *approximate* bipartite matching with the number of edges within a factor of $\left(1 - \frac{1}{\rho}\right)$ of that in the optimal matching for arbitrarily specified ρ in $O^*(\rho)$ matrix inversions ($O^*(X) = O(X \log^k n)$, i.e., we ignore logarithmic factors of n in stating most bounds in this paper). At present the best-known logarithmic time parallel algorithm for finding an approximate matching is that for finding a maximal matching that contains at least half of the edges in the optimal matching by Karp and Wigderson [*J. ACM*, 32 (1985), pp. 762–773].

By combining the approximate matching algorithm discussed in this paper with an augmenting path algorithm it is possible to derive the optimal matching in $O^*(v^{1/2})$ time. The previous fastest parallel algorithms for general bipartite graphs are those by Vaidya [*Proc. 22nd Ann. ACM Symp. Theory Computing*, 1990, pp. 583–589], which runs in $O^*((ve)^{1/4})$ time and that by Goldberg, Plotkin, and Vaidya [*Proc 29th IEEE Symp. Foundations of Computer Science*, 1990, pp. 175–185], which obtains solutions in $O^*(v^{2/3})$ time.

Key words. interior point methods, parallel algorithms, path following algorithms, bipartite matching, network flow algorithms, linear programming

AMS subject classifications. 90C05, 90C27, 90B10

1. Introduction.

The significance of the bipartite matching problem is well recognized in the contexts of both sequential as well as parallel computation. Historically, matching has been one of the most well-studied problems in both combinatorics and graph theory and several efficient sequential algorithms are known for it. Recently the importance of the problem has been recognized for parallel computation as well. The efficiency of several parallel algorithms depends on the parallel complexity of bipartite matching. For example Aggarwal and Anderson [1] and Aggarwal, Anderson, and Kao [2] show, respectively, that an NC algorithm for bipartite matching implies an NC algorithm for the problems of constructing depth-first search trees in undirected and directed graphs. (An NC algorithm is a logarithmic time algorithm that requires a polynomial number of processors.) Also, an NC algorithm for bipartite matching will lead to NC algorithms for the unit capacity network flow problem according to Karp, Upfal, and Wigderson [10] and hence for networks with capacities polynomially bound in n.

Bipartite matching is known to be in RNC (i.e., randomized algorithms are known to converge to the solution in logarithmic time with a polynomial number of processors with extremely high probability) (see Karp, Upfal, and Wigderson [10] and Mulmuley, Vazirani, and Vazirani [13]); however, despite extensive research there is no known deterministic algorithm that always converges to the optimum matching in logarithmic time. Special cases of the problem are known to be in NC. Lev, Pippenger, and Valiant [11] gave an algorithm to find a perfect matching in a regular bipartite graph; Miller

*Received by the editors September 24, 1992; accepted for publication (in revised form) June 15, 1994. This research was supported in part by National Science Foundation grant CCR-89180780.

†This work was carried out when the author was in the School of Electrical Engineering, Cornell University, Ithaca, New York. Current address: AT&T Bell Labs, Murray Hill, New Jersey 07974 (lkg@mhcnet.att.com).

and Naor [12] gave an algorithm to find a perfect matching in a planar graph if one exists; and Grigoriev and Karpinski [6] related the number of steps to the permanent and proved that the number of steps is related to the logarithm of the permanent.

In this paper we show how to obtain an approximate matching much faster than an exact matching. An *approximate* matching is defined as a set of edges that do not share any vertex and the cardinality of which is within a factor of $(1-\frac{1}{\rho})$ of that in the optimal matching for an arbitrarily specified ρ. This can be obtained in $O^*(\rho)$ matrix inversions using the algorithm described in this paper. (As mentioned in the abstract, $O^*(X) = O(X \log^k n)$, i.e., we ignore logarithmic factors of n in stating most bounds in this paper.) It is possible to do a matrix inversion in polylogarithmic time using a polynomial number of processors; the number of processors required is that required to invert an $[v \times v]$ matrix in polylogarithmic time. In case the optimal matching is desired, ρ is chosen to be $\sqrt{v}$ and an augmenting path algorithm is used to find the remaining (at most $O^*(v^{1/2})$) edges which leads to an overall time bound of $O^*(v^{1/2})$.

The fastest parallel algorithms for general graphs are those developed by Vaidya [16], that use interior point methods that run in $O^*((ve)^{1/4})$ matrix inversions and that, due to Goldberg, Plotkin, and Vaidya [4], use combinatorial properties to obtain solutions in $O^*(v^{2/3})$ time. Goldberg et al. [5] first pioneered the use of interior point methods in parallel computation for bipartite matching algorithms and gave an $O^*(e^{1/2})$ time parallel algorithm—in this paper we make use of the rounding algorithm due to this paper. Vaidya [16] used another interior point method using volumetric considerations to reduce the number of matrix inversions required by the interior point algorithm. The heart of the algorithm described in this paper is a new interior point method. We show that by using this interior point method along with the rounding algorithm of Goldberg et al. [5] and a standard augmenting path algorithm, it is possible to derive the optimal matching in $O^*(v^{1/2})$ time in parallel.

The linear programming (LP) problem can be visualized as the problem of finding the extreme point in a polytope in a given direction. This may be mathematically represented as

$$\sum_{j=1}^{n} A_{ij}x_j + C_i \geq 0, \quad i = 1, 2, \ldots, m, \tag{1.1}$$
$$\text{Max} \sum_{j=1}^{n} B_j x_j.$$

An interior point algorithm starts with a point in the interior of the feasible region (the polytope: $(\sum_j A_{ij}x_j + C_i \geq 0)$) and successively makes steps in the interior to reduce the objective function. Karmarkar [8] used interior point methods to develop an LP algorithm that converges in $O(mL)$ steps where m is the number of constraints and L the bit length of the input. Since then this has been improved to $O(\sqrt{m}L)$ steps by Renegar [15]. There has been extensive research in interior point methods and a host of algorithms has been developed. All these are based on the logarithmic barrier function in one way or another. The advantages of the logarithmic barrier function are well known; the dual has good properties that enable the steps to be well bounded. The disadvantage is that there are no known ways to obtain a fractional improvement better than $O(\frac{1}{\sqrt{m}})$ in each step with the logarithmic barrier function and hence it has not been possible to improve the bound to better than $O(\sqrt{m}L)$ iterations. This paper introduces a new potential function that varies as $\frac{1}{d_p^{\beta-1}}$, where d_p is the distance of the

center (i.e., the point at which the potential function is minimized in the polytope) from the hyperplane and β is $O(\log v)$. The advantage of this potential function is that due to the high degree of nonlinearity, the minimum stays well away from all the constraint hyperplanes; also the effect of all hyperplanes further away than twice the distance of the closest hyperplane can be neglected, thereby reducing the effective number of hyperplanes.

The algorithm described in this paper runs on the dual of the bipartite matching problem. The initial location of the center is obtained by superposing a v-dimensional cube, the hyperplanes of which dominate the gradient exerted at the center considering a point close to the center of the cube. Gradually, the *weight* of the hyperplanes of the cube is reduced and a Newton–Raphson (N-R) step is used to compute the new location of the center. Using standard interior point methods, it would take $O^*(\sqrt{m})$ steps to find the center for a constant reduction of the weight of the hyperplanes. However, in this paper we show that the N-R method for the above problem can be made to converge in $O^*(1)$ steps. This uses a particular case of a recently proved result, presented in this paper, according to which a convex function of the type $\sum_{ij} g_{ij}(u_i+u_j)+\sum_i g_i(u_i)$, with g_{ij} and g_i having positive definite and slowly varying Hessians, can be minimized in a cube $|u_i - c_i| \leq \kappa$ in $O^*(1)$ matrix inversions. Note that the minimization problem would be trivial without the g_{ij} since there would be no coupling between the various directions and the problem could be separated. The interior point method of this paper finds the minimum with coupling between the axes provided it is of the form indicated above. The dual of the matching problem has its constraints of the form indicated above. In case an approximate matching is desired with the number of edges within $\frac{v}{\rho}$ of the optimum, it is shown that the corresponding center of the interior point method will be at a distance of at least $O(\rho)$ from each hyperplane of the matching polytope. The center can then be obtained in $O^*(\rho)$ time. Once the center is obtained, an equivalent point in the primal polytope can be immediately obtained at which the value of the objective function is within $O^*(\frac{v}{\rho})$ of that of the optimal vertex. The algorithm due to Goldberg, et al. [5] is used to round this to a better vertex in $O^*(1)$ time. In case the optimal matching is desired, ρ is chosen to be $v^{1/2}$ and an augmenting path algorithm is used to find the remaining (at most $O^*(v^{1/2})$) edges which leads to an overall time bound of $O^*(v^{1/2})$.

The paper is organized as follows: §2 presents the broad framework and some of the terminology; §3 contains a simplified description of the interior point method; §4 describes the steps in the algorithm (without proofs); §5 proves the theorems mentioned in the previous section.

2. Terminology and framework. As mentioned briefly in the Introduction, LP is represented as

$$\sum_{j=1}^{n} A_{pj}x_j + C_p \geq 0, \quad p = 1, 2, \ldots, m \tag{2.1}$$

$$\text{Max} \sum_{j=1}^{n} B_j x_j.$$

The outward normal vector to the hyperplane p is defined as $\hat{e}_p$ and equals $\sum_j A_{pj}\hat{e}_j$ (here $\hat{e}_j$ is the usual unit vector in the jth coordinate direction). The bipartite match-

ing problem can be formulated as LP, e.g., Papadimitriou and Steiglitz [14]

$$
\begin{aligned}
& f_{ij} \geq 0, \quad (i,j) \in E, \\
& \sum_{\substack{j \in v \\ (i,j) \in E}} f_{ij} \leq 1, \quad i = 1,2,\ldots,v, \\
& \text{Max} \quad \sum_{\substack{i,j=1 \\ (i,j) \in E}}^{v} f_{ij}.
\end{aligned} \tag{2.2}
$$

The value of f_{ij} in the final solution denotes whether or not the edge between the i and j vertices is included in the matching. By unimodularity of the constraint matrix on the left-hand side (LHS) it may be shown that all vertices of the polytope have f_{ij} either 0 or 1, e.g., Papadimitriou and Steiglitz [14]. Note that in this case n, the number of variables, is v; and m, the number of additional constraints, is e. $v \leq e \leq v^2$ and therefore $O(\log e) = O(\log v)$.

Taking the dual of (2.2) gives (2.3), which we shall be concerned with for most of this paper:

$$
\begin{aligned}
& u_i \geq 0, \quad i = 1,2,\ldots,v, \\
& \frac{u_i + u_j}{2} \geq \frac{1}{2}, \quad (i,j) \in E, \\
& \text{Min} \quad \sum_{i=1}^{v} u_i.
\end{aligned} \tag{2.3}
$$

The factor of 2 in the second equation of (2.3) is inserted to simplify the definitions of distances. The distances are defined as $d_p = u_i$ and $(\frac{u_i+u_j-1}{2})$ for the two kinds of hyperplanes present in (2.3). Some of the symbols used in this paper with their brief meanings are as follows.

1. $\bar{\delta}$ = step made by center.
2. $\delta_p = \bar{\delta} \cdot (-\hat{e}_p)$, i.e., component *towards* hyperplane. As mentioned after (2.1), $\hat{e_p}$ is the unit outward vector to the hyperplane p.
3. Δ is the standard symbol used to denote the change in a particular quantity.
4. $\bar{d_0}$ is the vector connecting the center and the point at which the objective function is at its optimized value. d_0 is the component of $\bar{d_0}$ in the direction of the gradient of the objective function.
5. K and κ are universal symbols for constants and used in different places with different meanings, their use is local to particular theorems or lemmae. K generally denotes constants of magnitude greater than 1 and κ denotes constants less than 1.
6. k is a scaling factor whose usage is reserved for a particular term discussed in (2.8) below. This varies the relative weight of the constraining cube—initially this is a large amount $(\frac{1}{8\beta})^\beta$ and is gradually reduced (to $(\frac{1}{128\beta})^\beta$).
7. ϵ, Γ, and γ are used to denote small but finite constants. ε is used to denote an infinitesimal.
8. w_p, ν_p, ω_p denote the weights of the pth hyperplanes (elaborated on in (2.6) below). w_p denotes the weight of the hyperplanes of the matching polytope, ν_p the weights of the hyperplanes of the constraining cube, and ω_p is a generic term denoting weights of arbitrary hyperplanes. The changes of weights of the hyperplanes of the matching polytope are always negative, those of the cube may be positive or negative.

However the changes of weights are finite and the total weights do not change by more than a constant factor in the course of the algorithm (this is proved in Theorem 3).

9. Hyperplanes indicated by $p \in M$ are the constraint hyperplanes of the matching polytope (2.3) and hyperplanes indicated by $p \in C$ are those of the surrounding cube.

10. Ω_{ij} is used to denote the weight of the (i,j) edge in a weighted matching. Our interest lies in the case when all Ω_{ij} are equal to 1; however, in intermediate steps we consider small deviations from 1 in Theorems 9 and 11.

11. ρ, as mentioned in the abstract, denotes the degree of approximation of the matching. The algorithm finds a bipartite matching with the number of edges within a factor of $(1 - \frac{1}{\rho})$ of that in the optimal matching. In case the optimal matching is desired, ρ is chosen to be $\sqrt{v}$ and an augmenting path algorithm is used to find the remaining (at most $O^*(v^{1/2})$) edges, which leads to an overall time bound of $O^*(v^{1/2})$.

12. β is chosen to be $8\lceil \log_2 v \rceil + 1$. Note that β is an odd integer of magnitude $O^*(1)$.

13. λ denotes the *force* that pushes the center toward the optimum. This is large enough to ensure that the center gets pushed close enough to the optimum. It is chosen to be $(2\rho)^\beta$. λ is held constant during the course of the algorithm, while the effect of the constraining cube is reduced by reducing k.

14. $f(\bar{u})$, $f_1(\bar{u})$, $f_2(\bar{u})$ and $\psi(\bar{u})$, $\psi_1(\bar{u})$, $\psi_2(\bar{u})$ denote potential functions discussed below in (2.4)–(2.9).

15. The equations in this paper are expressed in *vector* notation, i.e., component by component; this is in contrast to papers in most interior point methods where *matrix* notation is used. Especially in the case of the bipartite matching problem, where there are at most two variables per constraint, this notation is more insightful, e.g., see the discussion after (2.5).

Consider the potential function $f_1(\bar{u})$ for the system of hyperplanes corresponding to the dual of the bipartite matching problem (2.3) as defined below:

$$f_1(\bar{u}) = \sum_{p \in M} \frac{1}{(\beta - 1) d_p^{\beta - 1}} + \sum_{i=1}^{v} \lambda u_i. \tag{2.4}$$

The algorithm of this paper finds the minimum of $f_1(\bar{u})$. At the minimum, the gradient of $f_1(\bar{u})$ is zero, i.e.,

$$-\nabla f_1(\bar{u}) = \sum_{p \in M} \frac{1}{d_p^\beta} \hat{e}_p - \sum_{i=1}^{v} \lambda \hat{e}_i = 0. \tag{2.5}$$

Here $\frac{1}{d_p^\beta} \hat{e}_p$ represents the effect of the potential due to the hyperplanes of the matching polytope, as mentioned after (2.1), $\hat{e_p}$ is the unit outward vector to the hyperplane p; $\lambda \hat{e}_i$ a component in the direction of the gradient of the objective function. (2.5) may be visualized as an external force $-\sum_{i=1}^{v} \lambda \hat{e}_i$ balanced by outward forces exerted by the hyperplanes that vary inversely as the βth power of distance. The center (minimum of the potential function) is the point at which they balance. The problem is to find the center. It is shown that if $\lambda = (2\rho)^\beta$ and $\beta = 8\lceil \log_2 v \rceil + 1$, (note that β by the above definition is an odd number) it is possible to obtain the center in $O^*(\rho)$ matrix inversions and the optimal matching, from this location of the center, in $O^*(v/\rho)$ steps. Therefore if ρ is chosen to be $\sqrt{v}$, then it is possible to obtain the optimal matching in $O^*(v^{1/2})$ time.

In order to track the minimum of $f_1(\bar{u})$ using the Newton-Raphson (N-R) method, a variable weight w_p is defined for each hyperplane which is varied to take second order effects into account. The symbol $f(\bar{u})$ is used to denote the modified potential function

$$f(\bar{u}) = \sum_{p \in M} \frac{w_p}{(\beta - 1) d_p^{\beta - 1}} + \sum_{i=1}^{v} \lambda u_i. \tag{2.6}$$

With this modification (2.5) becomes

$$-\nabla f(\bar{u}) = \sum_{p \in M} \frac{w_p}{d_p^{\beta}} \hat{e}_p - \sum_{i=1}^{v} \lambda \hat{e}_i = 0. \tag{2.7}$$

$f_2(\bar{u})$ represents the difference between $f(\bar{u})$ and $f_1(\bar{u})$; i.e., $f(\bar{u}) = f_1(\bar{u}) + f_2(\bar{u})$.

In this paper we show that by using an interior point method, it is possible to find the minimum of the function $f_1(\bar{u})$, with d_p of the form indicated in (2.3); furthermore, the center is constrained to a cube such that none of the hyperplanes of the matching polytope intersect the cube. For this, examine minima of the function $\psi_1(\bar{u})$ defined by

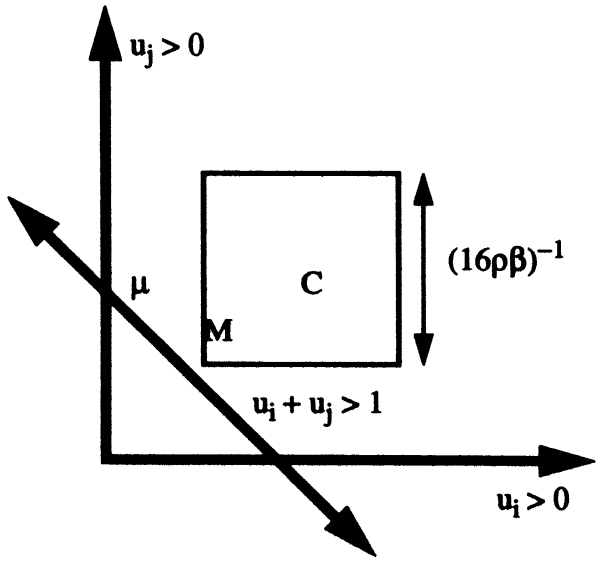
$$\psi_1(\bar{u}) = \left(\sum_{p \in M} \frac{1}{(\beta - 1) d_p^{\beta - 1}} + \sum_{i=1}^{v} \lambda u_i \right) + k \sum_{p \in C} \frac{1}{(\beta - 1) d_p^{\beta - 1}}. \tag{2.8}$$

Note that the cube will be defined in such a way that none of the faces of the cube ever intersect any of the hyperplanes of the matching polytope. k is a scaling factor: when it is small the potential function is close to the desired potential function and when it is large the potential function is dominated by the hyperplanes of the cube. As mentioned before, the weights of the hyperplanes are perturbed to take second order effects into account; a new function $\psi(\bar{u})$ is defined.

$$\psi(\bar{u}) = \left(\sum_{p \in M} \frac{w_p}{(\beta - 1) d_p^{\beta - 1}} + \sum_{i=1}^{v} \lambda u_i \right) + k \sum_{p \in C} \frac{\nu_p}{(\beta - 1) d_p^{\beta - 1}}. \tag{2.9}$$

There are two symbols, ν_p and w_p, used for the weights since the equations for the changes in weights of the cube and the matching hyperplanes are slightly different. We show in §§3–5 that we can indeed follow the trajectory of the minimum of $\psi(\bar{u})$ while reducing k by a factor of $O\left(\beta \left|\frac{\delta_p}{d_p}\right|_{\max}\right)$ in each step. In order to keep nonlinear effects bounded $\left|\frac{\delta_p}{d_p}\right|_{\max}$ will be kept less than $O(\frac{1}{\beta^2})$ in each step. Therefore in $O(\beta^2)$ iterations the algorithm will be able to decrease k by a factor of $2^{K\beta}$, at which point the second term on the right-hand side (RHS) of (2.9) becomes negligible, and the point at which $\psi(\bar{u})$ is minimum approaches the point inside the cube at which $f(\bar{u})$ is minimum.

In order to find the minimum of $f(\bar{u})$ in the entire matching polytope, the process described above can be iterated. It is possible to show that the minimum of $f(\bar{u})$ will lie at least $\frac{1}{4\rho}$ from each hyperplane of the matching polytope (Corollary 4A) *and* within the cube $\{3 > u_i > 0\}$. Thus if the side of the cube is smaller than $\frac{1}{4\rho}$, it will not intersect any of the hyperplanes of the matching polytope (it is actually chosen



(a) The dual of the matching polytope is defined by the hyperplanes $u_i > 0$; $u_i + u_j > 1$; where $(i,j) \in E$ as mentioned in (2.3).

(b) A cube of side $\frac{1}{16\rho\beta}$ where $\beta = 8\lceil \log_2 v \rceil + 1$, is inscribed within the polytope obtained by introducing the additional constraints $u_i < 3$ to the dual of the matching polytope (2.3). C is the center of this cube and M the point in the cube at which the potential function $f_1(\bar{u}) = \sum_{p \in M} \frac{1}{(\beta-1)d_p^{\beta-1}} + \sum_{i=1}^{v} \lambda u_i$ is minimum; $\lambda = (2\rho)^\beta$.

(c) The interior point method repeatedly finds the minimum (M) of $f_1(\bar{u})$ in the cube and then shifts the center of the cube to M.

(d) The center C will always stay at least at a distance of $\frac{1}{4\rho}$ from each of the hyperplanes of the matching polytope and within the large cube defined by $\{3 > u_i > 0\}$. Therefore by convexity of $f_1(\bar{u})$, $(f_1(C) - f_1(M)) \geq \frac{1}{48\rho\beta}(f_1(C) - f_1(\mu))$, where $f_1(\bar{u})$ is minimum at μ in the entire matching polytope.

(e) Within $O^*(\beta\rho)$ repetitions of (b) and (c), $f_1(M)$ approaches within $O(v^{-K})$ of $f_1(\mu)$.

FIG. 1. *Simplified outline of algorithm to find an approximate matching with the number of edges within a factor of* $(1 - O(\frac{1}{\rho}))$ *of the optimal matching.*

to be $\frac{1}{16\rho\beta}$ to accommodate second order effects). The algorithm is summarized in Fig. 1.

The algorithm proceeds by surrounding the current position of the center with a cube of side $\frac{1}{16\beta\rho}$. The center of the cube is denoted by C whose coordinates are represented by $(c_1, c_2, \ldots, c_i, \ldots)$; the minimum of $f(\bar{u})$ in the cube is assumed to be at the point M and the minimum of $f(\bar{u})$ in the entire matching polytope is assumed to lie at the point μ. The final point obtained by the interior point method in the cube is denoted by F. The values of $f(\bar{u})$ at these points are represented as $f(C)$, $f(M)$, $f(\mu)$, and $f(F)$, respectively. Since, when k is large, the minimum of $\psi(\bar{u})$ approaches the minimum of $f(\bar{u})$ inside the cube, it is clear that $f(F) \approx f(M)$. By the argument of the previous paragraph, the cube will never intersect any of the hyperplanes of the matching polytope, and thus the interior point method of §3 (mentioned briefly in the Introduction) can be used for finding the minimum of $f(\bar{u})$. By convexity of the

function $f(\bar{u})$, $\{f(C) - f(F)\} \geq \frac{1}{48\beta\rho}\{f(C) - f(\mu)\}$. The actual constant is smaller than $\frac{1}{48\beta\rho}$ due to higher order effects and is derived in Theorem 6 as $\frac{1}{768\beta\rho}$. Therefore, in $O^*(\beta\rho)$ iterations, $f(\bar{u})$ becomes very close to $f(\mu)$. This point is transferred to the primal to obtain an interior point of the matching polytope. A matching is derived from this by using the rounding algorithm of Goldberg et al. [5].

As mentioned previously in this section, in order to derive the exact matching, ρ is chosen to be $v^{1/2}$ and an approximate matching which has at most $O^*(\frac{v}{\rho})$, i.e., $O^*(v^{1/2})$, fewer edges than the optimal matching can be determined in $O^*(\rho)$, i.e., $O^*(v^{1/2})$ steps. It is possible to increase the number of edges in the set by unity in $O^*(1)$ steps using a standard augmenting path algorithm (see Even [3]). Thus all the edges in the optimum matching can be obtained in $O^*(v^{\frac{1}{2}})$ steps of the augmenting path algorithm giving an overall time bound of $O^*(v^{1/2})$.

3. Interior point method. This section develops the interior point method that the later algorithm uses. We show how to minimize the function

$$f_1(\bar{u}) = \sum_{p \in M} \frac{1}{(\beta - 1)d_p^{\beta-1}} + \sum_{i=1}^{v} \lambda u_i$$

in a cube $|u_i - c_i| \leq \kappa$, where none of the hyperplanes in the matching polytope intersects the cube $|u_i - c_i| \leq \kappa$. This section describes a simplified version of the algorithm without second and higher order effects. The detailed algorithm, including these effects, is described in §4 and proved in Theorems 2 and 3 in §5. As mentioned after (2.3), here $d_p = u_i$ or $\left(\frac{u_i + u_j - 1}{2}\right)$ (these are the only two kinds of hyperplanes present in the dual of the matching polytope (2.3) and the superposed cube). Consider the function $\psi_1(\bar{u})$ as mentioned in (2.8):

$$\psi_1(\bar{u}) = \left(\sum_{p \in M} \frac{1}{(\beta - 1)d_p^{\beta-1}} + \sum_{i=1}^{v} \lambda u_i\right) + k \sum_{p \in C} \frac{1}{(\beta - 1)d_p^{\beta-1}}$$

with $\beta = 8\lceil \log_2 v \rceil + 1$ and $\lambda = (2\rho)^\beta$. The d_p denote the distances from the hyperplanes of the matching polytope and the constraining cube as defined in (2.3). k is a scaling parameter that starts from a large quantity and gradually decreases. The first component is the part to be minimized and the last component constrains the center to the indicated cube. As k decreases the first component dominates.

An N-R type step is used to find the location of the minimum after every change of k. We show that k can change by a constant factor in each step while guaranteeing that $\bar{\delta}$ makes a small step relative to each of the hyperplanes. Contrast this with the logarithmic potential where k could only improve by a factor of $(1 + O(\frac{1}{\sqrt{m}}))$ in each step. In order to derive the rate of change of k, consider the conditions for a minimum before and after the step $\bar{\delta}$. The condition $\nabla\psi_1(\bar{u}) = 0$ yields

$$\left(\sum_{p \in M} \frac{1}{d_p^\beta}\, \hat{e}_p - \sum_{i=1}^{v} \lambda \hat{e}_i\right) + k \sum_{p \in C} \frac{1}{d_p^\beta}\, \hat{e}_p = 0. \tag{3.1}$$

Changing k by Δk and making the step $\bar{\delta}$ so as to stay at the minimum of $\psi_1(\bar{u})$, we obtain the condition

$$\left(\sum_{p \in M} \frac{1}{(d_p - \delta_p)^\beta}\, \hat{e}_p - \sum_{i=1}^{v} \lambda \hat{e}_i\right) + (k + \Delta k) \sum_{p \in C} \frac{1}{(d_p - \delta_p)^\beta}\, \hat{e}_p = 0. \tag{3.2}$$

Subtracting (3.1) from (3.2) and retaining first order terms gives (a detailed analysis that incorporates higher order effects is presented in Theorems 2 and 3 in §5);

$$\sum_{p\in M} \frac{\beta\delta_p}{d_p^{\beta+1}}\,\hat{e}_p + k\sum_{p\in C} \frac{\beta\delta_p}{d_p^{\beta+1}}\,\hat{e}_p + \Delta k \sum_{p\in C} \frac{1}{d_p^{\beta}}\,\hat{e}_p = 0. \tag{3.3}$$

Next construct a vector $\bar{\delta}_{(\beta)}$ as indicated below with which we will take the dot products of (3.1) and (3.3) and take their ratio to obtain $\frac{\Delta k}{k}$. Normally in logarithmic potential based interior point methods this vector is $\bar{\delta}$; however, using $\bar{\delta}$ here will give $\frac{\Delta k}{k}$ as approximately

$$\frac{\sum_p \frac{\delta_p^2}{d_p^{\beta+1}}}{\sum_p \frac{\delta_p}{d_p^{\beta+1}}}$$

which is difficult to lower bound. Instead we use the vector $\bar{\delta}_{(\beta)}$ which has its component in $\hat{e}_i$ as δ_i^{β} where δ_i is the component of $\bar{\delta}$ in the direction $\hat{e}_i$ ($\hat{e}_i$ represents the coordinate direction i).

Taking the dot products of (3.1) and (3.3) with $\bar{\delta}_{(\beta)}$ and then their ratio gives

$$\frac{\Delta k}{k} = \frac{\beta \sum_{p\in C} \left(\frac{\delta_p}{d_p}\right)^{\beta+1} + \frac{\beta}{k}\sum_{p\in M} \frac{\delta_p\cdot\delta_{(\beta)p}}{d_p^{\beta+1}}}{\sum_{p\in C}\left(\frac{\delta_p}{d_p}\right)^{\beta}}. \tag{3.4}$$

The first term in the numerator is clearly positive since β is an odd number. The last term in the numerator is either of the form $\frac{\beta}{k}\frac{\delta_i^{\beta+1}}{d_p^{\beta+1}}$ in case the matching hyperplane is $(u_i \geq 0)$, or else of the form

$$\frac{\beta}{k}\frac{(\delta_i+\delta_j)(\delta_i^{\beta}+\delta_j^{\beta})}{4d_p^{\beta+1}}$$

in case the matching hyperplane is $(u_i + u_j \geq 1)$. In either case this term is positive. This is the one step in which we require the potential function to be of the form $\psi(u) = \sum_{ij}\{g_i(u_i) + g_{ij}(u_i+u_j)\}$ since without this restriction it is not possible to bound the step in terms of a finite constant times $\frac{\Delta k}{k}$. Since the numerator is positive, the sign of the denominator adjusts itself according to the sign of $\frac{\Delta k}{k}$. The first term gives the required gain as follows from Lemma 3 of §5. This is because the magnitude of the term

$$\frac{\sum_{p\in C}\left(\frac{\delta_p}{d_p}\right)^{\beta+1}}{\sum_{p\in C}\left(\frac{\delta_p}{d_p}\right)^{\beta}}$$

is at least $\frac{1}{2}\left|\frac{\delta_p}{d_p}\right|_{\max}$ if $\beta = O(\log v)$, i.e., at least half of the magnitude of the $\frac{\delta_p}{d_p}$ with the maximum absolute value. An intuitive way to see this is to look upon

$$\frac{\sum_{p\in C}\left(\frac{\delta_p}{d_p}\right)^{\beta+1}}{\sum_{p\in C}\left(\frac{\delta_p}{d_p}\right)^{\beta}}$$

as a weighted average of $\left|\frac{\delta_p}{d_p}\right|$ with weights $\left|\frac{\delta_p}{d_p}\right|^\beta$. The weights of any $\left|\frac{\delta_p}{d_p}\right|$ that are less than $\frac{1}{2}\left|\frac{\delta_p}{d_p}\right|_{\max}$ become very small as compared to the weight of $\left|\frac{\delta_p}{d_p}\right|_{\max}$.

In case the log barrier function had been used, an equation similar to (3.4) with $\beta = 1$ would be obtained, the RHS of this could only be bounded to $\frac{1}{\sqrt{v}}\left|\frac{\delta_p}{d_p}\right|_{\max}$, not $\frac{1}{2}\left|\frac{\delta_p}{d_p}\right|_{\max}$.

4. The algorithm. The following algorithm is described for general ρ. Steps 0–8 will yield an approximate matching with the number of edges within a factor of $(1 - O(\frac{1}{\rho}))$ of that in the optimal matching. In order to obtain the optimal matching set $\rho = \sqrt{v}$ and execute Steps 0–9.

Step 0. Initially set the location of the starting point to $c_i = 1$ for all i.

Step 1. Define the cube of interest by the hyperplanes $|u_i - c_i| \leq \frac{1}{32\beta\rho}$. Initially set k to a relatively large quantity ($k = \frac{1}{(8\beta)^\beta}$) and all $w_p = 1$ and $\nu_p = 1$. The center of the cube is denoted by C.

Step 2. Perturb the positions of one of the hyperplanes of the cube slightly in each direction i by s_i so that at the center of the cube C, $\nabla\psi_1(\bar{u}) = 0$, $\psi_1(\bar{u})$ is defined in (2.8). In case after Step 1, $\frac{\partial\psi_1}{\partial u_i} > 0$, move the hyperplane $(u_i - c_i) \leq \frac{1}{32\beta\rho}$, closer to the center so that with the new positions of the hyperplane $\frac{\partial\psi_1}{\partial u_i} = 0$. In case after Step 1, $\frac{\partial\psi_1}{\partial u_i} < 0$, move the hyperplane $-(u_i - c_i) \leq \frac{1}{32\beta\rho}$, closer to the center so that with the new positions of the hyperplane $\frac{\partial\psi_1}{\partial u_i} = 0$. Repeat for all i.

Step 3. Reduce k by Δk so that $\frac{\Delta k}{k} = -\frac{1}{12\beta}$ and make the step $\bar{\delta}$ indicated in (4.1). It is proved in Theorem 2 in §5 that $\left|\frac{\delta_p}{d_p}\right|_{\max} \leq \frac{1}{6\beta^2}\cdot\bar{\delta}$, $\Delta\nu_p$, and Δw_p are determined by the following equation similar to (3.3) (observe that in (4.1b) and (4.1c) all $\Delta\nu_p$, Δw_p are second order terms in δ_p):

$$\text{(4.1a)}\qquad \sum_{p\in M}\frac{\beta w_p\delta_p}{d_p^{\beta+1}}\,\hat{e}_p + k\sum_{p\in C}\frac{\beta\nu_p\delta_p}{d_p^{\beta+1}}\,\hat{e}_p + \Delta k\sum_{p\in C}\frac{\nu_p}{d_p^\beta}\,\hat{e}_p = 0,$$

$$\text{(4.1b)}\qquad \frac{\Delta w_p}{(d_p-\delta_p)^\beta} + \left\{\frac{w_p}{(d_p-\delta_p)^\beta} - \frac{w_p}{d_p^\beta} - \frac{\beta w_p\delta_p}{d_p^{\beta+1}}\right\} = 0,$$

$$\text{(4.1c)}\qquad (k+\Delta k)\frac{\Delta\nu_p}{(d_p-\delta_p)^\beta} + k\left\{\frac{\nu_p}{(d_p-\delta_p)^\beta} - \frac{\nu_p}{d_p^\beta} - \frac{\beta\nu_p\delta_p}{d_p^{\beta+1}}\right\} + \Delta k\left\{\frac{\nu_p}{(d_p-\delta_p)^\beta} - \frac{\nu_p}{d_p^\beta}\right\} = 0.$$

Step 4. If $k \geq \frac{1}{(128\beta)^\beta}$, go to Step 3; else if $k < \frac{1}{(128\beta)^\beta}$, denote the position of the center by F and continue on to Step 5.

Step 5. In case, $\left|\frac{\partial f}{\partial u_i}\right|$ in any direction u_i is more than $\frac{(2\rho)^\beta}{2v^{1.5}}$, go to Step 1 with C (the new center of the cube) set to F; else continue on to Step 6.

Step 6. Obtain the corresponding point in the primal by the following transformation. Let the distance of the nearest hyperplane of the matching polytope (2.3) to F

be $d_{p,\min}$. Set f_{ij} on the (i,j) edge in the graph to 0 in case the hyperplane $(\frac{u_i+u_j}{2} \geq \frac{1}{2})$ is at a distance greater than $2d_{p,\min}$ from F. Otherwise, set f_{ij} on the (i,j) edge in the graph to $\frac{(2\rho)^{-\beta}}{d_p^\beta}$, where d_p are the distances of the $(\frac{u_i+u_j}{2} \geq \frac{1}{2})$ hyperplane from the center F.

Step 7. Evaluate $S_i = \sum_j f_{ij}$ for all vertices i. For each edge (i,j) let S_{ij} be the larger of S_i and S_j; in case S_{ij} be greater than unity, reduce f_{ij} by a factor S_{ij}.

Step 8. Use the rounding algorithm of Goldberg et al. [5] to derive a better vertex than this interior point. This yields an approximate matching in which the number of edges is within a factor of $\left(1 - \frac{9}{\rho}\right)$ of that in the optimal matching.

Step 9. In case the optimal matching is desired, use an augmenting path algorithm to obtain the remaining edges one by one. It is possible to increase the number of edges in the set by unity in $O^*(1)$ time using a standard augmenting path algorithm (see Even [3]).

5. Proofs of theorems. We start by proving that the solution to (4.1a) for $\bar{\delta}$ is unique and well defined. This follows from the convexity of the potential function $\psi(\bar{u})$—a formal proof from first principles is carried out in Theorem 1.

THEOREM 1. *The solution $\bar{\delta}$ to the following system of linear equations* (5.1.1) *is uniquely determined for given $\omega_p > 0$, X_p and d_p.*

$$\sum_{p=1}^{m} \left(\omega_p \frac{\delta_p}{d_p^2} \hat{e}_p + \frac{X_p}{d_p} \hat{e}_p \right) = 0, \quad \omega_p > 0. \tag{5.1.1}$$

The summation in (5.1.1) *is over all constraint hyperplanes. It is assumed that the* LHS *of* (5.1.1) *includes the set of terms $\sum_i \left(\omega_i \frac{\delta_i}{d_i^2} \hat{e}_i + \frac{X_i}{d_i} \hat{e}_i \right)$ that correspond to the equations $\{x_i = 0, \forall i = 1, \ldots, n\}$, i represent orthogonal coordinate directions.*

Proof. This follows from the fact that (5.1.1) is of the form $\sum_j H_{ij}\delta_j = v_i$, where H_{ij} is the Hessian of the convex function $-\sum_p (\omega_p \log d_p)$. For completeness we carry out a proof from first principles. We show that $\bar{\delta}$ exists and is uniquely determined by the vector equation (5.1.1). In order to do this we first write the equation in terms of orthogonal coordinates and unit vectors, i.e. $(x_1, x_2, \ldots, x_i, \ldots, x_n)$ and $(\hat{e}_{x1}, \hat{e}_{x2}, \ldots, \hat{e}_{xi}, \ldots, \hat{e}_{xn})$ where $\hat{e}_{xi}$ is the unit vector in the x_i direction:

$$d_p = \sum_{j=1}^{n} A_{pj} x_j + C_p, \tag{5.1.2a}$$

$$\delta_p = -\sum_{j=1}^{n} A_{pj} \delta x_j. \tag{5.1.2b}$$

Note that since by definition of δ_p in §2, it is the component of $\bar{\delta}$ *towards* the hyperplane p, there is a negative sign in (5.1.2b).

The vector $X_p \hat{e}_p$ may be written as $\sum_{i=1}^{n} A_{pi} X_p \hat{e}_i$. Using this, (5.1.1) may be written as

$$-\sum_{p=1}^{m} \sum_{j=1}^{n} [A_{pi}] \left[\frac{\omega_p A_{pj} \delta x_j}{d_p^2} \right] \hat{e}_i = -\sum_{p=1}^{m} A_{pi} \frac{X_p}{d_p} \hat{e}_i. \tag{5.1.3}$$

This is a set of n simultaneous linear equations in δx_j. To check that δx_j exist and are uniquely determined, we need to prove that the determinant of the matrix $X_{ij} = \sum_{p=1}^{m} A_{pi} \frac{\omega_p}{d_p^2} A_{pj}$ is nonzero. Assume that it is zero. Then there exists some nonzero vector Y_j such that $\sum_{j=1}^{n} X_{ij} Y_j = [0]$ (i.e., the zero vector). Therefore, $\sum_{i,j=1}^{n} Y_i X_{ij} Y_j = 0$. This implies that $\rightarrow \sum_{p=1}^{m} \sum_{i,j=1}^{n} Y_i A_{pi} \frac{\omega_p}{d_p^2} A_{pj} Y_j = 0$. Denote $\sum_{j=1}^{n} A_{pj} Y_j = b_p$. Then $\sum_{p=1}^{m} \frac{\omega_p}{d_p^2} b_p^2 = 0$.

Since $\omega_p > 0$, the only way this is possible is if $b_p = 0$ for all p. Since the system of equations includes the set of terms $\sum_{i=1}^{n} \omega_i \frac{\delta_i}{d_i^2} \hat{e}_i$, it follows that all components of Y_i are zero, a contradiction, since by assumption $\bar{Y}$ was a nonzero vector. It follows that our assumption is false and δx_j and hence $\bar{\delta}$ are uniquely determined. □

COROLLARY 1.1. *The solution $\bar{\delta}$ to* (4.1a) *is uniquely determined.*

Proof. The proof follows by substituting for ω_p and X_p and using Theorem 1.

For $p \in M$, substitute $\omega_p = \frac{\beta w_p}{d_p^{\beta-1}}$ and $X_p = 0$; for $p \in C$, substitute $\omega_p = \frac{k\beta\nu_p}{d_p^{\beta-1}}$ and $X_p = \Delta k \frac{\nu_p}{d_p^{\beta-1}}$. □

We show that we can follow the *central trajectory* as k in (2.8) is reduced by a factor of $O(\frac{1}{\beta})$ in each step. For the particular case of the potential function considered in this paper of the form $\phi(\bar{u}) = \sum_{i,j=1}^{v} g_{ij}(u_i + u_j) + \sum_{i=1}^{v} g_i(u_i)$ with the functions g_{ij} and g_i slowly varying, while following the central trajectory, it is possible to take higher order terms in the perturbation expansion of $\phi(\bar{u})$ into account provided $\frac{\delta_p}{d_p} = O(\frac{1}{\beta^2})$, by perturbing the weights of the hyperplanes as shown in Grover [7] for the logarithmic potential function. The convergence result with this modification is presented in Theorem 2. To prove this we first need two algebraic results proved in Lemmas 2.1 and 2.2.

LEMMA 2.1. *In case α is even, and $\sum_{i=1}^{m} w_i x_i^{\alpha} = \kappa$ and $\kappa, w_i > 0$, then*

$$\left| \sum_{i=1}^{m} w_i x_i^{\alpha-1} \right| \leq \kappa^{1-\frac{1}{\alpha}} \left(\sum_i w_i \right)^{\frac{1}{\alpha}}.$$

Proof. This is proved by using Lagrange multipliers as follows. Consider the function $\Lambda(\bar{x})$

$$\Lambda(\bar{x}) = \sum_i w_i x_i^{\alpha-1} - \lambda \left(\sum_i w_i x_i^{\alpha} - \kappa \right). \tag{5.2.0}$$

The condition $\frac{\partial \Lambda(\bar{x})}{\partial x_i} = 0$ along with (5.2.0) implies that at an extremum all nonzero x_i are equal. Assume that P x_i's are nonzero. Then at the maximum of the objective function all these are equal (to say x_0) and $(\sum_{i|x_i \neq 0} w_i) x_0^{\alpha} = \kappa$. Therefore

$$x_0 = \pm \left(\frac{\kappa}{\sum_{i|x_i \neq 0} w_i} \right)^{\frac{1}{\alpha}}$$

and the objective function becomes $(\sum_{i|x_i \neq 0} w_i)^{1/\alpha} \kappa^{1-1/\alpha}$. This is clearly highest for P as high as possible, i.e., m, from which the lemma follows. □

LEMMA 2.2. *In case* $\frac{1}{4} \leq w_i \leq 4$; α *is even; then*

$$\left| \frac{\sum_{i=1}^{m} w_i x_i^{\alpha}}{\sum_{i=1}^{m} w_i x_i^{\alpha-1}} \right| \geq \frac{|x_i|_{\max}}{(16m)^{\frac{1}{\alpha}}},$$

where $|x_i|_{\max}$ *denotes the maximum of all* $|x_i|$.

Proof. Using Lemma 2.1 and substituting, $\sum_{i=1}^{m} w_i x_i^{\alpha} = \kappa$, it follows that

$$\left| \frac{\sum_{i=1}^{m} w_i x_i^{\alpha}}{\sum_{i=1}^{m} w_i x_i^{\alpha-1}} \right| \geq \left(\frac{\kappa}{\kappa^{1-\frac{1}{\alpha}} (4m)^{\frac{1}{\alpha}}} \right), \text{ which equals } \frac{\kappa^{\frac{1}{\alpha}}}{(4m)^{\frac{1}{\alpha}}}.$$

Using bounds on w_i and definition of κ, it follows that the above is greater than $|x_i|_{\max}/(16m)^{\frac{1}{\alpha}}$. □

In Step 3 of the algorithm of §4, it is easy to accommodate higher order terms in the perturbation expansion of the potential function provided $\left|\frac{\delta_p}{d_p}\right|_{\max} = O(\frac{1}{\beta^2})$, by perturbing the weights of the hyperplanes as shown in Grover [7] for the logarithmic potential function. The convergence result with this modification is presented below in Lemma 2.3 and Theorem 2.

LEMMA 2.3. *In case before Step* 3 *of* §4 *the starting point is at the minimum of* $\psi(\bar{u})$ *(as defined in* (2.9)*), then after making steps in accordance with* (4.1a)–(4.1c)*, the new point obtained will also be at the minimum of* $\psi(\bar{u})$.

Proof. Since the initial point is at the minimum of $\psi(\bar{u})$ (as defined in (2.9)), the condition $\nabla\psi(\bar{u}) = 0$ gives

$$\left(\sum_{p \in M} \frac{w_p}{d_p^{\beta}} \hat{e}_p - \sum_{i=1}^{v} \lambda \hat{e}_i \right) + k \sum_{p \in C} \frac{\nu_p}{d_p^{\beta}} \hat{e}_p = 0. \tag{5.2.1}$$

After changing k by Δk and making the step $\bar{\delta}$ in order to satisfy the condition $\nabla\psi(\bar{u}) = 0$, $\bar{\delta}, \Delta k, \Delta w_p, \Delta \nu_p$ must satisfy

$$\left(\sum_{p \in M} \frac{w_p + \Delta w_p}{(d_p - \delta_p)^{\beta}} \hat{e}_p - \sum_{i=1}^{v} \lambda \hat{e}_i \right) + (k + \Delta k) \left(\sum_{p \in C} \frac{\nu_p + \Delta \nu_p}{(d_p - \delta_p)^{\beta}} \right) \hat{e}_p = 0. \tag{5.2.2}$$

Subtracting (5.2.1) from (5.2.2) and rearranging terms gives

$$\begin{aligned} &\left\{ \sum_{p \in M} \frac{\beta w_p \delta_p}{d_p^{\beta+1}} \hat{e}_p + k \sum_{p \in C} \frac{\beta \nu_p \delta_p}{d_p^{\beta+1}} \hat{e}_p + \Delta k \sum_{p \in C} \frac{\nu_p}{d_p^{\beta}} \hat{e}_p \right\} \\ &+ \sum_{p \in M} \left\{ \frac{\Delta w_p}{(d_p - \delta_p)^{\beta}} + \left\{ \frac{w_p}{(d_p - \delta_p)^{\beta}} - \frac{w_p}{d_p^{\beta}} - \frac{\beta w_p \delta_p}{d_p^{\beta+1}} \right\} \right\} \hat{e}_p \\ &+ \sum_{p \in C} \left\{ (k + \Delta k) \frac{\Delta \nu_p}{(d_p - \delta_p)^{\beta}} + k \left\{ \frac{\nu_p}{(d_p - \delta_p)^{\beta}} - \frac{\nu_p}{d_p^{\beta}} - \frac{\beta \nu_p \delta_p}{d_p^{\beta+1}} \right\} \right. \\ &\qquad \left. + \Delta k \left\{ \frac{\nu_p}{(d_p - \delta_p)^{\beta}} - \frac{\nu_p}{d_p^{\beta}} \right\} \right\} \hat{e}_p = 0. \end{aligned} \tag{5.2.3}$$

In Step 3 of the algorithm of §4, $\bar{\delta}$ is chosen so as to make the term in the first line in the LHS of (5.2.3) zero; $\Delta\nu_p$ and Δw_p are chosen so as to make the two terms in the last two lines each zero (note that the terms in the last two lines are both second order terms in $\frac{\delta_p}{d_p}$). Thus with this choice of $\bar{\delta}$, $\Delta\nu_p$, and Δw_p, (5.2.2) is satisfied after the step $\bar{\delta}$ since (5.2.1) was initially satisfied. □

THEOREM 2. *In case* $\frac{1}{4} \leq w_p, \nu_p \leq 4$, *then in Step* 3 *of the algorithm of* §4, $\left|\frac{\delta_p}{d_p}\right|_{\max} \leq \frac{2}{\beta}\left|\frac{\Delta k}{k}\right|$. *The perturbation of the weights of the hyperplanes of the matching polytope* (w_p) *are bounded by* $\left\{-\frac{\beta^2}{4}\left(\frac{\delta_p}{d_p}\right)^2 \geq \frac{\Delta w_p}{w_p} \geq -\beta^2\left(\frac{\delta_p}{d_p}\right)^2\right\}$. *Also, the perturbation in the weights of the hyperplanes of both the cube and the matching polytope are bounded by* $\left\{\left|\frac{\Delta\nu_p}{\nu_p}\right|, \left|\frac{\Delta w_p}{w_p}\right| \leq 5\left(\frac{\Delta k}{k}\right)^2\right\}$. *Note that the change in weight of the hyperplanes of the matching polytope is always negative.*

Proof. We show that $\left|\frac{\delta_p}{d_p}\right|$ is bounded in terms of $\frac{\Delta k}{k}$ by an equality of the form (3.4) (the only difference is that the hyperplanes instead of having unit weights, now have perturbed weights). In order to bound $\left|\frac{\delta_p}{d_p}\right|$, construct a vector $\bar{\delta}_{(\beta)}$ as indicated below with which we will take the dot products of (5.2.1) and (5.2.3) and take their ratio to obtain $\frac{\Delta k}{k}$. Normally in logarithmic potential based interior point methods this vector is $\bar{\delta}$; however, using $\bar{\delta}$ here will give $\left|\frac{\Delta k}{k}\right|$ as approximately

$$\frac{\sum_p \frac{\delta_p^2}{d_p^{\beta+1}}}{\sum_p \frac{\delta_p}{d_p^{\beta+1}}},$$

which is difficult to lower bound. Instead we use the vector $\bar{\delta}_{(\beta)}$ which has its component in $\hat{e}_i$ as δ_i^β, where δ_i is the component of $\bar{\delta}$ in the direction $\hat{e}_i$. ($\hat{e}_i$ represents the unit vector in the coordinate direction i.) Taking the dot products of (5.2.3) and (5.2.2) with $\bar{\delta}_{(\beta)}$, making use of the fact that Δw_p, $\Delta\nu_p$ are chosen so as to make the last two terms in (5.2.3) zero, and then taking the ratio of the two dot products, we obtain

$$\frac{\Delta k}{k} = \frac{\beta\sum_{p\in C}\nu_p\left(\frac{\delta_p}{d_p}\right)^{\beta+1} + \frac{1}{k}\beta\sum_{p\in M}\frac{w_p\delta_p\cdot\delta_{(\beta)p}}{d_p^{\beta+1}}}{\sum_{p\in C}\nu_p\left(\frac{\delta_p}{d_p}\right)^{\beta}}. \tag{5.2.4a}$$

The last term in the numerator is either of the form $\frac{1}{k}\beta w_p\frac{\delta_i^{\beta+1}}{d_p^{\beta+1}}$ in case the matching hyperplane is $(u_i \geq 0)$; or else of the form $\frac{1}{k}\beta w_p\frac{(\delta_i+\delta_j)(\delta_i^\beta+\delta_j^\beta)}{4d_p^{\beta+1}}$ in case the matching hyperplane is $(u_i + u_j \geq 1)$. In either case this term is positive since β is odd. This is the one step in which we require the function to be of the form $\psi(u) = \sum_{ij}\{g_i(u_i)+g_{ij}(u_i+u_j)\}$ since without this restriction it is not possible to bound the step. Note that the denominator of the RHS of (5.2.4a) is of the same sign as the LHS since the numerator of the RHS is positive by the above argument. The first term in the RHS gives the required gain as follows from Lemma 2.2 since $\beta = 8\lceil\log_2 v\rceil + 1$ and $\left\{4 \geq w_p, \nu_p \geq \frac{1}{4}\right\}$. Therefore

$$\left|\frac{\Delta k}{k}\right| \geq \frac{\beta}{2}\left|\frac{\delta_p}{d_p}\right|_{\max}. \tag{5.2.4b}$$

Substituting the value of $\frac{\Delta k}{k}$ from Step 3 of the algorithm of §4, as $-\frac{1}{12\beta}$, into (5.2.4b):

$$\left|\frac{\delta_p}{d_p}\right|_{\max} \leq \frac{1}{6\beta^2}. \tag{5.2.4c}$$

Next we show that $\frac{\Delta\nu_p}{\nu_p}$ and $\frac{\Delta w_p}{w_p}$ are small in each step. According to (4.1b) and (4.1c), $\Delta\nu_p$ and Δw_p are given by

$$\frac{\Delta w_p}{(d_p-\delta_p)^\beta} + \left\{\frac{w_p}{(d_p-\delta_p)^\beta} - \frac{w_p}{d_p^\beta} - \frac{\beta w_p \delta_p}{d_p^{\beta+1}}\right\} = 0, \tag{5.2.5a}$$

(5.2.5b)

$$(k+\Delta k)\frac{\Delta\nu_p}{(d_p-\delta_p)^\beta} + k\left\{\frac{\nu_p}{(d_p-\delta_p)^\beta} - \frac{\nu_p}{d_p^\beta} - \frac{\beta\nu_p\delta_p}{d_p^{\beta+1}}\right\} + \Delta k\left\{\frac{\nu_p}{(d_p-\delta_p)^\beta} - \frac{\nu_p}{d_p^\beta}\right\} = 0.$$

Using the mean value theorem on the Taylor series expansion of the term in parentheses gives

$$\frac{\Delta w_p}{(d_p-\delta_p)^\beta} + \frac{\beta(\beta+1)\delta_p^2 w_p}{2(d_p-\kappa\delta_p)^{\beta+2}} = 0,\ 0\leq\kappa\leq 1, \tag{5.2.6a}$$

(5.2.6b)

$$(k+\Delta k)\frac{\Delta\nu_p}{(d_p-\delta_p)^\beta} + k\frac{\beta(\beta+1)\delta_p^2\nu_p}{2(d_p-\kappa_1\delta_p)^{\beta+2}} + \Delta k\frac{\beta\delta_p\nu_p}{(d_p-\kappa_2\delta_p)^{\beta+1}} = 0,\ 0\leq\kappa_1,\kappa_2\leq 1.$$

Using (5.2.4c) to bound $\frac{\delta_p}{d_p}$,

$$-\frac{\beta^2}{4}\frac{\delta_p^2}{d_p^2} \geq \frac{\Delta w_p}{w_p} \geq -\beta^2\frac{\delta_p^2}{d_p^2}. \tag{5.2.7a}$$

Applying (5.2.4c) again, the above may be written as

$$0 \geq \frac{\Delta w_p}{w_p} \geq -4\left(\frac{\Delta k}{k}\right)^2. \tag{5.2.7b}$$

Similarly it follows from (5.2.6b) that

$$5\left(\frac{\Delta k}{k}\right)^2 \geq \frac{\Delta\nu_p}{\nu_p} \geq -5\left(\frac{\Delta k}{k}\right)^2. \tag{5.2.7c}$$

□

Finally using Theorem 2 and summing the changes in k, w_p, and ν_p, we prove that the total changes in weights are actually bounded.

THEOREM 3. *After making* $36\beta^2$ *steps as specified in* (4.1a), *the scaling factor* k *falls by a factor of at least* $2^{4\beta}$ *and the weights* w_p *and* ν_p *satisfy*

$$\frac{1}{4} \leq \frac{\nu_{p,\text{final}}}{\nu_{p,\text{initial}}} \leq 4; \qquad \frac{1}{4} \leq \frac{w_{p,\text{final}}}{w_{p,\text{initial}}} \leq 1. \tag{5.3.0}$$

Proof. $\frac{\Delta k}{k}$ in each repetition of Step 3 of the algorithm is $-\frac{1}{12\beta}$ and therefore in $36\beta^2$ steps, k will fall by a factor of $(1-\frac{1}{12\beta})^{36\beta^2}$. This is less than $2^{-4\beta}$.

The proof for the variation of weights is given for the w_p's. Exactly the same sequence of steps leads to a similar bound for the variation in ν_p's. Denote the weight w_p after the jth step by $w_{p,j+1}$ and assume the condition $1 \leq w_{p,i} \leq 4$ is satisfied for $i = 1, 2, \ldots, I$. The ratio of the initial and final weights can be written as

$$\frac{w_{p,I+1}}{w_{p,1}} = \frac{w_{p,I+1}}{w_{p,I}} \cdot \frac{w_{p,I}}{w_{p,I-1}} \cdots \frac{w_{p,2}}{w_{p,1}}. \tag{5.3.1}$$

Using Theorem 2, each term on the RHS may be bounded to lie between 1 and $\left(1 - 5\left(\frac{\Delta k}{k}\right)^2\right)$. Therefore

$$1 \geq \frac{w_{p,I+1}}{w_{p,1}} \geq \left(1 - 5\left(\frac{\Delta k}{k}\right)^2\right)^I. \tag{5.3.2}$$

For $\frac{\Delta k}{k} = -\frac{1}{12\beta}$ and $I \leq 36\beta^2$, it is seen that $1 \geq w_{p,I+1} \geq \frac{1}{4}$. The condition required by Theorem 2 is hence satisfied and the desired bound in Theorem 3 follows. □

Using the fact that the weights of each hyperplane stay in a finite range, it is possible to prove that the center stays at least $\frac{1}{4\rho}$ from each hyperplane (Corollary 3.3A) and it stays in the box $\{3 > u_i > 0,\ \forall i\}$ (Corollary 3.3B).

COROLLARY 3.1. *Assuming the minimum distance to each hyperplane of the matching polytope from the center is more than* $\frac{1}{4\rho}$, *the initial movement of the hyperplanes of the cube in Step* 2 *of the algorithm in* §4 *is less than* $\frac{1}{128\rho\beta}$.

Proof. Assume a hyperplane with its normal in the i direction moves towards C by more than $\frac{1}{128\rho\beta}$. It is easily seen that the gradient due to this dominates and is greater than the sum of the components of all other gradients in the i direction. The net gradient in the i direction due to the hyperplanes of the cube is $\sum_{p \in C} \frac{k\nu_p}{d_p^\beta}$; with $k = \frac{1}{(8\beta)^\beta}$, the magnitude of this becomes at least $\frac{1}{2}(\frac{16\rho}{3})^\beta$. Since all the hyperplanes of the matching polytope are assumed to be at least at a distance of $\frac{1}{4\rho}$, the total gradient in the i direction due to the hyperplanes of the matching polytope and $\lambda \hat{e}_i$ is limited by $v\,(4\rho)^\beta + (2\rho)^\beta$ which could not possibly balance the gradient due to the hyperplanes of the cube. Thus the component of gradient of $\psi(\bar{u})$ in the i direction could not possibly be zero as required by Step 2 of the algorithm of §4. Therefore the initial assumption was false and the initial movement of a hyperplane of the cube in Step 2 of the algorithm will be less than $\frac{1}{128\rho\beta}$. □

COROLLARY 3.2A. *Assume that at the beginning of the* i*th repetition of the loop in Steps* 1–5 *of the algorithm of* §4, *the initial movement of the hyperplanes of the cube in Step* 2 *are all smaller than* $\frac{1}{128\beta\rho}$ *and the distance of the center from a hyperplane (say* η*) of the matching polytope becomes less than* $\frac{1}{4\rho}(1 + \frac{1}{\beta})$. *Then after the execution of the loop in Steps* 1–5 *the center moves away from the hyperplane* η *by a distance of at least* $\frac{1}{128\rho\beta}$.

Proof. In order to see that in the next repetition of the loop in Steps 1–5, the center will move away from this hyperplane by at least $\frac{1}{128\rho\beta}$, assume that $d_p < \frac{1}{4\rho}(1 + \frac{1}{\beta})$ for a particular hyperplane η. Plane η is either of the form $u_i > 0$ or $\frac{u_i + u_j}{2} > \frac{1}{2}$. In either case, consider the gradient in the direction i. According to (5.2.1), the following condition is always satisfied by the center $\left(\sum_{p \in M} \frac{w_p}{d_p^\beta}\, \hat{e}_p - \sum_{i=1}^{v} \lambda \hat{e}_i\right) + k \sum_{p \in C} \frac{\nu_p}{d_p^\beta}\, \hat{e}_p = 0$. Examining the components of the above vector equation in the i direction, it is observed that any components due to the hyperplanes $p \in M$ directed

in the i direction are in the $+i$ direction. Since according to the assumption, at least one of these planes has $d_p < \frac{1}{4\rho}(1+\frac{1}{\beta})$, it follows that the gradient in the $+i$ direction is at least $\frac{(4\rho)^\beta}{6}$. The component in the $-i$ direction is due to $-\lambda\hat{e}_i$ (which is $-(2\rho)^\beta$) and possibly due to the hyperplanes of the cube. Thus the net gradient due to the hyperplanes of the cube is at least $\{\frac{(4\rho)^\beta}{6} - (2\rho)^\beta\}$ in the $-i$ direction. Since $k \leq (\frac{1}{128\beta})^\beta$ by Step 4 of the algorithm in §4, and by Theorem 3 $\{\frac{1}{4} \leq \nu_p \leq 4\}$, it follows by Theorems 2 and 3 that after an execution of the loop in Steps 1–5 that the center is within a distance of $\frac{1}{256\beta\rho}$ of the hyperplane $(u_i - c_i) < \frac{1}{32\beta\rho} - s_i$, where s_i is the shift of the hyperplane $(u_i - c_i) < \frac{1}{32\beta\rho}$ in Step 2 of the algorithm. Assuming a maximum initial shift in Step 2 of $\frac{1}{128\beta\rho}$, it follows that the center gets pushed away by at least $\frac{1}{128\beta\rho}$ if it approaches any of the hyperplanes of the matching polytope by closer than $\frac{1}{4\rho}(1+\frac{1}{\beta})$. □

COROLLARY 3.2B. *Assume that at the beginning of the ith repetition of the loop in Steps* 1–5, *the initial movement of the hyperplanes of the cube in Step* 2 *are all smaller than* $\frac{1}{128\beta\rho}$ and *some coordinate* u_i *of the center is greater than* 2.5. *Then after the execution of the loop in Steps* 1–5 *the* u_i *coordinate of the center is reduced by at least* $\frac{1}{128\rho\beta}$.

Proof. The argument is the same as the argument of Corollary 3.2A. □

COROLLARY 3.3A. *The initial movement of the hyperplanes of the cube in Step* 1 *are all smaller than* $\frac{1}{128\beta\rho}$ and *the distance of the center from all hyperplanes of the matching polytope are always all greater than* $\frac{1}{4\rho}$.

Proof. Since the distance of the center from all hyperplanes of the matching polytope is greater than $\frac{1}{4\rho}$ at the start of the first iteration of the loop in Steps 1–5 (as specified in Step 0 of the algorithm), it follows by successively applying Corollary 3.1A and Corollary 3.2A that the distances are always greater than $\frac{1}{4\rho}$ and hence by Corollary 3.1A the initial movement of the hyperplanes are smaller than $\frac{1}{128\beta\rho}$. □

COROLLARY 3.3B. *The coordinates* $u_i, i = 1, 2, \ldots, v$ *of the center are always smaller than* 3.

Proof. The same argument as in Corollary 3.3A when used along with Corollary 3.2B shows that whenever any coordinate u_i of the center becomes greater than $3\left(1-\frac{1}{\beta}\right)$, it gets pushed back and hence cannot exceed 3. □

COROLLARY 3.4. *The final solution to the dual of the matching problem* (2.3) *lies in the box* $\{3 > u_i > 0,\ \forall i\}$.

Proof. By unimodularity of the constraint matrix it follows that all vertices of the polytope have each coordinate u_i either 0 or 1. □

As mentioned in the algorithm in §4, the final point in the cube, obtained just before Step 5, is denoted by F. In order to prove that $f(F)$ is close to the minimum of $f(\bar{u})$ in the cube, we first carry out the following transformation. First, consider any faces of the cube that are within a distance of $\frac{1}{128\beta\rho}$ of F. Shift these inward toward the center of the cube so that these pass through F.

THEOREM 4. *At the point* F *obtained just before Step* 5, *the function* $f(\bar{u})$ *is within* $(\frac{v\rho^{\beta-1}}{4\beta})$ *of its minimum in the modified cube obtained after the transformation described just before this theorem.*

Proof. Just before Step 5 at the point F, according to Lemma 2.3, we have

$$\nabla f(\bar{u}) + \sum_{p \in C} \frac{1}{k} \frac{\nu_p}{d_p^\beta} \hat{e}_p = 0. \tag{5.4.0}$$

The gradient of $f(\bar{u})$ at F can hence be estimated by the gradient due to $\sum_{p \in C} \frac{1}{k} \frac{\nu_p}{d_p^\beta}$ by using (5.4.0). It follows by the convexity of $f(\bar{u})$ that the reduction obtained in $f(\bar{u})$ by going from F to the minimum of $f(\bar{u})$ in the cube (the minimum point is denoted by M) is upper bounded by $\nabla f(\bar{u}).\bar{d_0}$, where $\bar{d_0}$ is the vector connecting F to M and $\nabla f(\bar{u})$ is the gradient of $f(\bar{u})$ at F. Therefore $\Delta f < \sum_{i=1}^{v} \frac{\partial f(\bar{u})}{\partial u_i} d_i$.

Consider hyperplanes of the cube with normals in the direction i. In case $f(\bar{u})$ decreases as we approach a hyperplane of the cube that was originally within a distance $d_i \leq \frac{1}{128\beta\rho}$, then in the transformed cube, no reduction in $f(\bar{u})$ is possible since this has been shifted to coincide with F. In case $f(\bar{u})$ decreases as we approach a hyperplane of the cube that was originally at a distance $d_i > \frac{1}{128\beta\rho}$, the reduction possible in $f(\bar{u})$ is bounded by $\left|\frac{\partial f(\bar{u})}{\partial u_i}\right| d_i$. Using (5.4.0) to evaluate $\frac{\partial f(\bar{u})}{\partial u_i}$ with k as $(128\beta)^\beta$ and ν_p upper bounded by 4, it follows that $\left|\frac{\partial f(\bar{u})}{\partial u_i}\right|$ is upper bounded by $4\rho^\beta$. d_i is upper bounded by the width of the cube which is $\frac{1}{16\beta\rho}$, it follows that the function $f(\bar{u})$ can fall by at most $\frac{\rho^{\beta-1}}{4\beta}$ in the direction i in the transformed cube and thus by $\frac{v\rho^{\beta-1}}{4\beta}$ in the v directions in the transformed cube. □

Our real interest lies not in the function $f(\bar{u})$, but in $f_1(\bar{u})$ as defined in (2.4)–(2.6). In order to estimate the change in $f_1(\bar{u})$, we show that the changes in $f_2(\bar{u})$ are much smaller than the changes in $f_1(\bar{u})$. Preliminary lemmas are first proved that are used in Corollary 5.1 to derive the desired result. We first show that if $\Phi(\bar{u}) = \Phi_1(\bar{u}) + \Phi_2(\bar{u})$ and the variations of Φ_2 in the region under consideration are small as compared to Φ_1, then if we have the approximate minimum of Φ, we also have the approximate minimum of Φ_1.

LEMMA 5.1. *Consider a function $\Phi(\bar{u}) = \Phi_1(\bar{u}) + \Phi_2(\bar{u})$ defined in a closed region R. Let $\Phi(\bar{u})$ be within Γ of its minimum in R at the point F ($\Gamma > 0$), i.e., $\Phi(F) \leq \Phi(P) + \Gamma$ for all points P in R. Also assume for some given point C in R, that $|\Phi_2(C) - \Phi_2(P)| \leq \kappa\,(\Phi_1(C) - \Phi_1(F))$ for all points P in R. Let the minimum of Φ_1 in R be at the point M. Then*

$$(\Phi_1(C) - \Phi_1(F)) \geq \left(\frac{1}{1+2\kappa}\right)(\Phi_1(C) - \Phi_1(M)) - \frac{\Gamma}{1+2\kappa}. \tag{5.5.1}$$

Proof. Since $\Phi(F)$ is within Γ of the minimum of $\Phi(\bar{u})$ in R, it follows from the definition of Φ

$$(\Phi_1(C) - \Phi_1(F) + \Phi_2(C) - \Phi_2(F)) \geq (\Phi_1(C) - \Phi_1(M) + \Phi_2(C) - \Phi_2(M)) - \Gamma. \tag{5.5.2}$$

Therefore

$$(\Phi_1(C) - \Phi_1(F)) \geq ([\Phi_1(C) - \Phi_1(M)] + [\Phi_2(C) - \Phi_2(M)] - [\Phi_2(C) - \Phi_2(F)]) - \Gamma. \tag{5.5.3}$$

Using the bounds on $[\Phi_2(C) - \Phi_2(M)]$ and $[\Phi_2(C) - \Phi_2(F)]$ from the assumption of the lemma, we obtain the following relation which immediately leads to the required bound

$$(1+2\kappa)\,(\Phi_1(C) - \Phi_1(F)) \geq (\Phi_1(C) - \Phi_1(M)) - \Gamma. \tag{5.5.4}$$

□

LEMMA 5.2. *In Step* 3 *of the algorithm of* §4 *the maximum displacement* $\left|\frac{\delta_p}{d_p}\right|$ *relative to any hyperplane of the matching polytope will be at most* $\frac{1}{48\beta^3}$.

Proof. For each direction i consider one of the two hyperplanes

$$|u_i - c_i| < \frac{1}{32\beta\rho} - s_i$$

that is closer to the center. This has $d_p < \frac{1}{32\beta\rho}$. According to Theorem 2, $\left|\frac{\delta_p}{d_p}\right|_{\max} < \frac{1}{6\beta^2}$. Therefore $|\delta_i| < \frac{1}{192\beta^3\rho}$. According to Corollary 3.3, the nearest hyperplane of the matching polytope is at least at a distance of $\frac{1}{4\rho}$ from the center. $\left|\frac{\delta_p}{d_p}\right|_{\max}$ for the matching hyperplanes are bounded by $\frac{|\delta_i|}{d_p}$ and $\frac{|\delta_i|+|\delta_j|}{2d_p}$, both of which are at most $\frac{1}{48\beta^3}$. □

LEMMA 5.3. *In each repetition of the loop in Steps* 1–5, *the maximum total displacement relative to any hyperplane of the matching polytope* (*denoted by* $\left|\frac{\gamma_p}{d_p}\right|_{\max}$) *will be at most* $\frac{1}{8\beta}$.

Proof. The total displacement allowed in any coordinate direction i from the center of the cube, within the cube (denoted by γ_i), is equal to half of the side of the cube, which gives $|\gamma_i| < \frac{1}{32\rho\beta}$. According to Corollary 3.3, the nearest hyperplane of the matching polytope is at least at a distance of $\frac{1}{4\rho}$ from the center. $\left|\frac{\gamma_p}{d_p}\right|_{\max}$ for the matching hyperplanes are bounded by $\frac{|\gamma_i|}{d_p}$ and $\frac{|\gamma_i|+|\gamma_j|}{2d_p}$, both of which are at most $\frac{1}{8\beta}$. □

Next we prove that the maximum variation in $f_2(\bar{u})$ in the cube defined in Step 1 of the algorithm is less than the reduction in $f_1(\bar{u})$. This will be used along with Lemma 5.1 to show that $f_1(\bar{u})$ is close to its minimum in the cube.

THEOREM 5. $|f_2(C) - f_2(P)| \leq \frac{1}{2}\{f_1(C) - f_1(F)\}$, *where* P *is any point in the cube and* C *the center of the cube defined in Step* 1 *of the algorithm and* F *the point obtained in Step* 5.

Proof. According to (5.2.1), before making the step $\bar{\delta}$

$$\left(\sum_{p\in M} \frac{w_p}{d_p^\beta}\,\hat{e}_p - \sum_{i=1}^{v} \lambda \hat{e}_i\right) + k\sum_{p\in C} \frac{\nu_p}{d_p^\beta}\,\hat{e}_p = 0, \tag{5.5.5}$$

$\bar{\delta}$ is determined by (4.1)

$$\sum_{p\in M} \frac{\beta w_p \delta_p}{d_p^{\beta+1}}\,\hat{e}_p + k\sum_{p\in C} \frac{\beta \nu_p \delta_p}{d_p^{\beta+1}}\,\hat{e}_p + \Delta k \sum_{p\in C} \frac{\nu_p}{d_p^\beta}\,\hat{e}_p = 0. \tag{5.5.6}$$

Δw_{lp} is used to denote the change of weight (w_p) of the hyperplane p of the matching polytope in the lth step of the center as obtained in (4.1b) and (4.1c); the total change in weight Δw_p after I steps is given by $\sum_{l=1}^{I} \Delta w_{lp}$, where I is the total number of steps executed in the loop in Steps 1–5. Similarly Δf_{1l} is used to denote the change in $f_1(\bar{u})$ in the lth step and Δf_1 to denote the total change in $f_1(\bar{u})$ in I steps. Note that by Theorem 2, Δw_{lp} is always negative. Also by Theorem 2, $\frac{\Delta w_{lp}}{w_p}$ is bounded by

$$\frac{\beta^2}{4}\frac{\delta_p^2}{d_p^2} \leq -\frac{\Delta w_{lp}}{w_p} \leq \beta^2 \frac{\delta_p^2}{d_p^2}. \tag{5.5.7}$$

$-\Delta w_{lp}$ is bounded in terms of Δf_{1l} by the following procedure. Take the dot products of (5.5.5), (5.5.6) with $\bar{\delta}$ and use (5.5.7). First taking the dot product of (5.5.5) with $\bar{\delta}$:

$$\sum_{p\in M} \frac{w_p\delta_p}{d_p^\beta} + \sum_{i=1}^{v} \lambda\delta_i = -k \sum_{p\in C} \frac{\nu_p\delta_p}{d_p^\beta}. \tag{5.5.8a}$$

Taking the dot product of (5.5.6) with $\bar{\delta}$:

$$\sum_{p\in C} \frac{k\beta\nu_p\delta_p^2}{d_p^{\beta+1}} + \sum_{p\in M} \frac{\beta w_p\delta_p^2}{d_p^{\beta+1}} = -\Delta k \sum_{p\in C} \frac{\nu_p\delta_p}{d_p^\beta}. \tag{5.5.8b}$$

Combining (5.5.8a) and (5.5.8b) yields

$$\frac{\Delta k}{k}\left(\sum_{p\in M} \frac{w_p\delta_p}{d_p^\beta} + \sum_{i=1}^{v} \lambda\delta_i\right) = \sum_{p\in C} \frac{k\beta\nu_p\delta_p^2}{d_p^{\beta+1}} + \sum_{p\in M} \frac{\beta w_p\delta_p^2}{d_p^{\beta+1}}. \tag{5.5.8c}$$

Using (5.5.7) on the RHS of (5.5.8c) gives

$$\frac{\Delta k}{k}\left(\sum_{p\in M} \frac{w_p\delta_p}{d_p^\beta} + \sum_{i=1}^{v} \lambda\delta_i\right) \geq -\sum_{p\in M} \frac{\Delta w_{lp}}{\beta d_p^{\beta-1}}. \tag{5.5.8d}$$

The LHS of (5.5.8d) may be written in terms of Δf_{1l}, by using the definition of $f_1(\bar{u})$ from (2.4) and taking its Taylor's series expansion with the quadratic remainder term, since from Lemma 5.2 it follows that for the hyperplane p of the matching polytope $\left|\frac{\delta_p}{d_p}\right|_{\max} \leq \frac{1}{48\beta^3}$. From the definition of $f_1(\bar{u})$ in (2.4), it follows that

$$-\Delta f_{1l} \geq -\left(\sum_{p\in M} \frac{\delta_p}{d_p^\beta} + \sum_{i=1}^{v} \lambda\delta_i\right) - \sum_{p\in M} \beta\frac{\delta_p^2}{d_p^{\beta+1}}. \tag{5.5.9a}$$

Rewriting the first term on the RHS in (5.5.9a) it follows that

$$-\Delta f_{1l} \geq -\left(\sum_{p\in M} \frac{w_p\delta_p}{d_p^\beta} + \sum_{i=1}^{v} \lambda\delta_i\right) + \sum_{p\in M}\left(\frac{\sum_{j=1}^{l-1} \Delta w_{jp}\delta_p}{d_p^\beta}\right) - \sum_{p\in M} \beta\frac{\delta_p^2}{d_p^{\beta+1}}. \tag{5.5.9b}$$

Equation (5.5.9b) can be written as

$$-\Delta f_{1l} - \sum_{p\in M}\left(\frac{\sum_{j=1}^{l-1} \Delta w_{jp}\delta_p}{d_p^\beta}\right) + \sum_{p\in M} \beta\frac{\delta_p^2}{d_p^{\beta+1}} \geq -\left(\sum_{p\in M} \frac{w_p\delta_p}{d_p^\beta} + \sum_{i=1}^{v} \lambda\delta_i\right). \tag{5.5.10a}$$

Combining (5.5.10a) and (5.5.8d), we obtain

$$\frac{\Delta k}{k}\left(\Delta f_{1l}+\sum_{p\in M}\left(\frac{\sum_{j=1}^{l-1}\Delta w_{jp}\delta_p}{d_p^{\beta}}\right)-\sum_{p\in M}\beta\frac{\delta_p^2}{d_p^{\beta+1}}\right)\geq-\sum_{p\in M}\frac{\Delta w_{lp}}{\beta d_p^{\beta-1}}. \tag{5.5.10b}$$

Using (5.5.7) to combine the last term on the LHS and the first term on the RHS of (5.5.10b), it follows that

$$\frac{\Delta k}{k}\Delta f_{1l}+\frac{\Delta k}{k}\sum_{p\in M}\left(\frac{\sum_{j=1}^{l-1}\Delta w_{jp}\delta_p}{d_p^{\beta}}\right)\geq-\sum_{p\in M}\frac{\Delta w_{lp}}{2\beta d_p^{\beta-1}}. \tag{5.5.11}$$

Next the RHS is combined with the second term on the LHS to yield a bound on the perturbation of weights in terms of Δf_1. Making use of the facts that $\Delta w_{lp}<0$ (Theorem 2), $\frac{\Delta k}{k}<0$ (Step 3 of the algorithm), Lemma 5.3 and denoting the distance of the center of the cube to a hyperplane p of the matching polytope as d_{p0}, (5.5.11) may be written as

$$-\sum_{p\in M}\frac{\Delta w_{lp}}{4\beta d_{p0}^{\beta-1}}\leq\frac{\Delta k}{k}\Delta f_{1l}+\frac{2\Delta k}{k}\sum_{p\in M}\left(\frac{\sum_{j=1}^{l-1}\Delta w_{jp}}{d_{p0}^{\beta-1}}\right)\left|\frac{\delta_p}{d_p}\right|_{\max}. \tag{5.5.12}$$

Sum (5.5.12) over all steps from $l=1$ to I:

$$-\sum_{l=1}^{I}\sum_{p\in M}\frac{\Delta w_{lp}}{4\beta d_{p0}^{\beta-1}}\leq\frac{\Delta k}{k}\sum_{i=1}^{I}\Delta f_{1l}+\frac{2\Delta k}{k}\sum_{p\in M}\left(\sum_{l=1}^{I}\frac{\sum_{j=1}^{l-1}\Delta w_{jp}}{d_{p0}^{\beta-1}}\right)\left|\frac{\delta_p}{d_p}\right|_{\max}. \tag{5.5.13}$$

Upper bounding the last term on the RHS by using the facts that all Δw_{lp} (by Theorem 2) and Δk is always negative (Step 3 of the algorithm in §4):

$$-\sum_{l=1}^{I}\sum_{p\in M}\frac{\Delta w_{lp}}{4\beta d_{p0}^{\beta-1}}\leq\frac{\Delta k}{k}\sum_{l=1}^{I}\Delta f_{1l}+\frac{2\Delta k}{k}I\sum_{p\in M}\frac{\sum_{j=1}^{I}\Delta w_{jp}}{d_{p0}^{\beta-1}}\left|\frac{\delta_p}{d_p}\right|_{\max}. \tag{5.5.14}$$

By Theorem 2, $\left(-\frac{\Delta k}{k}I\leq 3\beta\right)$. Using Lemma 5.2 to combine the last term on the RHS with the LHS

$$-\sum_{l=1}^{I}\sum_{p\in M}\frac{\Delta w_{lp}}{8\beta d_{p0}^{\beta-1}}\leq\frac{\Delta k}{k}\sum_{l=1}^{I}\Delta f_{1l}=\frac{\Delta k}{k}\Delta f_1. \tag{5.5.15}$$

$|f_2(C)-f_2(P)|$ for any point P is upper bounded by:

$$-\sum_{l=1}^{I}\sum_{p}\left(\frac{2\Delta w_{lp}}{d_{p0}^{\beta-1}}\right)\left|\frac{\gamma_p}{d_p}\right|_{\max},$$

where $\left|\frac{\gamma_p}{d_p}\right|_{\max}$ is the maximum total displacement relative to the hyperplane $p \in M$ within the cube defined in Step 1. $\left|\frac{\gamma_p}{d_p}\right|_{\max}$ is upper bounded by $\frac{1}{4\beta}$ by Lemma 5.3. Therefore by (5.5.15) it follows that $|f_2(C) - f_2(P)|$ is upper bounded by $\frac{4\Delta k}{k}\Delta f_1$, which is less than $\frac{1}{2}\{f_1(C) - f_1(F)\}$. □

COROLLARY 5.1. *The minimum of $f_1(\bar{u})$ in the transformed cube is assumed to be at the point M and F is the final point in the cube obtained after an execution of the loop in Steps* 1–5 *of the algorithm of* §4. *Then*

$$(f_1(C) - f_1(F)) \geq \frac{1}{2}(f_1(C) - f_1(M)) - \frac{v\rho^{\beta-1}}{8\beta}. \tag{5.5.16}$$

Proof. Substituting $\Gamma = \frac{v\rho^{\beta-1}}{4\beta}$ and $\kappa = \frac{1}{2}$ in Lemma 5.1 and using Theorems 4 and 5, the corollary follows. □

Using Corollary 5.1 and convexity of the function $f_1(\bar{u})$, we deduce a result similar to that indicated in Fig. 1.

THEOREM 6 . *In case $(f_1(C) - f_1(\mu) \geq 96v\rho^{\beta})$, then*

$$(f_1(C) - f_1(F)) \geq \frac{1}{768\beta\rho}(f_1(C) - f_1(\mu)).$$

Proof. The faces of the transformed cube are shifted by at most $\frac{1}{128\rho\beta}$ in Step 2 of the algorithm as proved in Corollary 3.3A, and by $\frac{1}{128\rho\beta}$ in the transformation mentioned just before Theorem 4. Since by Step 1 of the algorithm, each face of the cube was initially at a distance of $\frac{1}{32\rho\beta}$ from the center of the cube C, it follows that each face of the transformed cube is at least $\frac{1}{64\rho\beta}$ from C. Therefore if the transformed cube is magnified by a factor of $192\rho\beta$, it follows by Corollary 3.4, that it will certainly enclose the point μ (at which $f_1(\bar{u})$ is a minimum). By convexity of the function $f_1(\bar{u})$ it follows that if the minimum of $f_1(\bar{u})$ in the cube be at M, then

$$(f_1(C) - f_1(M)) \geq \frac{1}{192\beta\rho}(f_1(C) - f_1(\mu)). \tag{5.6.1}$$

Combining (5.6.1) and Corollary 5.1

$$(f_1(C) - f_1(F)) \geq \frac{1}{384\beta\rho}(f_1(C) - f_1(\mu) - 48\rho^{\beta}v). \tag{5.6.2}$$

Since by the assumption of this theorem $(f_1(C) - f_1(\mu)) \geq 96\rho^{\beta}v$, it follows that

$$(f_1(C) - f_1(F)) \geq \frac{1}{768\beta\rho}(f_1(C) - f_1(\mu)). \tag{5.6.3}$$ □

COROLLARY 6.1. *After $O^*(\rho)$ executions of the loop in Steps* 1–5, $(f_1(C) - f_1(\mu)) \leq 96v\rho^{\beta}$.

Proof. The initial value of $f_1(\bar{u})$ is dominated by the contribution due to $\sum_{i=1}^{v}\lambda u_i$, which gives $v(2\rho)^{\beta}$; the total value is therefore at most $2v(2\rho)^{\beta}$. Therefore $(f_1(C) - f_1(\mu))$ is initially at most $2v(2\rho)^{\beta}$ and according to Theorem 6, in each execution of the loop in Steps 1–5, $(f_1(C) - f_1(\mu))$ falls by a factor of at least $(1 - \frac{1}{768\beta\rho})$ as long as $(f_1(C) - f_1(\mu) \geq 96v\rho^{\beta})$. Since $\beta = O(\log v)$, the result follows. □

In order to transfer to the primal polytope, we need a bound on $\nabla f(\bar{u})$. To obtain this we show that if $f(\bar{u})$ is extremely close to its minimum, then $\nabla f(\bar{u})$ is extremely small.

THEOREM 7. *In case at the point F obtained after an execution of the loop in Steps 1–5 in §4, the function*

$$f_1(\bar{u}) = \sum_{p \in M} \frac{1}{(\beta-1)d_p^{\beta-1}} + \sum_{i=1}^{v} \lambda u_i,$$

where $\lambda = (2\rho)^\beta$ is within $\frac{(2\rho)^\beta}{128\beta^2\rho^2K^2v}$ of its minimum, then $\left|\frac{\partial f}{\partial u_i}\right|$ in any direction u_i is less than $\frac{(2\rho)^\beta}{K}$.

Proof. First observe that according to Corollary 3.3, the point P is at least at a distance of $\frac{1}{4\rho}$ from all the hyperplanes of the matching polytope (2.3). Assume that $\frac{\partial f_1}{\partial u_i}$ in some direction i is greater than $\frac{(2\rho)^\beta}{K}$. Consider the point P' displaced from P in the ith direction by $\frac{1}{64\rho^2\beta^2vK}$ in a direction opposite to $\frac{\partial f_1}{\partial u_i}$, i.e., in a direction so as to reduce f_1. The change in $f_1(\bar{u})$ (denoted by Δf_1) in going from P to P' is evaluated by Taylor's remainder theorem:

$$\Delta f_1 = \frac{\partial f_1(\bar{u})}{\partial u_i}\delta_i + \frac{1}{2}\frac{\partial^2 f_1(\bar{u}+\kappa\bar{\delta})}{\partial u_i^2}\delta_i^2; \quad 0 \le \kappa \le 1.$$

According to the assumption that $\left|\frac{\partial f_1}{\partial u_i}\right|$ in the direction i is greater than $\frac{(2\rho)^\beta}{K}$,

(5.7.1)
$$-\Delta f_1 \ge \frac{1}{K}(2\rho)^\beta|\delta_i| - \frac{1}{2}\left\{\sum_{\substack{x \\ i,x\in E}} \frac{\beta\delta_i^2}{(\frac{u_i+u_x-1-\kappa\delta_i}{2})^{\beta+1}} + \frac{\beta\delta_i^2}{(u_i-\kappa\delta_i)^{\beta+1}}\right\}; 0 \le \kappa \le 1.$$

Here $\delta_i = \frac{1}{64\rho^2\beta^2vK}$ and according to Corollary 3.3, $\left(\frac{u_i+u_x-1}{2} \ge \frac{1}{4\rho}\right)$ and also $\left(u_i \ge \frac{1}{4\rho}\right)$. Therefore the quadratic terms in δ_i are bounded as follows:

(5.7.2)
$$\begin{aligned}\left(\frac{u_i+u_x-1-\kappa\delta_i}{2}\right)^{\beta+1} &= \left(\frac{u_i+u_x-1}{2}\right)^{\beta+1}\left(1-\frac{\kappa\delta_i}{u_i+u_x-1}\right)^{\beta+1} \\ &\ge \frac{1}{2}\left(\frac{u_i+u_x-1}{2}\right)^{\beta+1}.\end{aligned}$$

Also,

(5.7.3)
$$(u_i-\kappa\delta_i)^{\beta+1} = u_i^{\beta+1}\left(1-\frac{\kappa\delta_i}{u_i}\right)^{\beta+1} \ge \frac{1}{2}u_i^{\beta+1}.$$

Using (5.7.2), (5.7.3) in (5.7.1)

(5.7.4)
$$-\Delta f_1 \ge \frac{1}{K}(2\rho)^\beta|\delta_i| - \left\{\sum_{\substack{x \\ i,x\in E}} \frac{\beta\delta_i^2}{(\frac{u_i+u_x-1}{2})^{\beta+1}} + \frac{\beta\delta_i^2}{u_i^{\beta+1}}\right\}.$$

Therefore,

$$-\Delta f_1 \geq \frac{1}{K}(2\rho)^\beta|\delta_i| - \beta\left|\frac{\delta_i}{d_p}\right|^2_{max}\left\{\sum_{\substack{x\\ i,x\in E}}\frac{1}{(\frac{u_i+u_x-1}{2})^{\beta-1}} + \frac{1}{u_i^{\beta-1}}\right\}. \tag{5.7.5}$$

Now

$$\left(\frac{1}{\beta-1}\right)\left\{\sum_{\substack{x\\ i,x\in E}}\frac{1}{(\frac{u_i+u_x-1}{2})^{\beta-1}} + \frac{1}{u_i^{\beta-1}}\right\}$$

is part of the objective function $f_1(\bar{u})$, as defined in (2.4), and it follows that it is at most equal to the initial value of $f_1(\bar{u})$ (since by Theorem 6, $f_1(\bar{u})$ is being reduced in each execution of the loop in Steps 1–5 of the algorithm of §4). Therefore, as in Corollary 6.1,

$$\sum_{\substack{x\\ i,x\in E}}\frac{1}{(\frac{u_i+u_x-1}{2})^{\beta-1}} + \frac{1}{u_i^{\beta-1}}$$

is less than $2\beta v\,(2\rho)^\beta$, which is an upper bound on the initial value of $f_1(\bar{u})$ (this follows by the same steps in Corollary 6.1). $\left|\frac{\delta_i}{d_p}\right|_{\max}$ is upper bounded, as in Lemma 5.2, by observing that $(d_p > \frac{1}{4\rho})$ and $(|\delta_i| = \frac{1}{64\rho^2\beta^2vK})$ which gives $(\left|\frac{\delta_i}{d_p}\right|_{\max} < \frac{1}{16\beta^2K\rho v})$. Therefore we obtain

$$-\Delta f_1 \geq \frac{(2\rho)^\beta}{128\beta^2\rho^2K^2v}. \tag{5.7.6}$$

This contradicts the assumption of the theorem and hence $\left|\frac{\partial f_1}{\partial u_i}\right|$ in any direction u_i could not be more than $\frac{(2\rho)^\beta}{K}$. □

COROLLARY 7.1. *After $O^*(\rho)$ executions of the loop in Steps* 1–5 *in the algorithm in* §4, $\left|\frac{\partial f_1}{\partial u_i}\right|$ *is less than* $\sqrt{\frac{12288\beta^2\rho^2v^2}{2^\beta}}\,(2\rho)^\beta$, *for all* i.

Proof. Using Corollary 6.1 and Theorem 7 with $K = \sqrt{\frac{2^\beta}{12288\beta^2\rho^2v^2}}$, we obtain

$$\left|\frac{\partial f_1}{\partial u_i}\right| \leq \sqrt{\frac{12288\beta^2\rho^2v^2}{2^\beta}}\,(2\rho)^\beta \quad \forall i. \qquad \square$$

COROLLARY 7.2. *At the point F obtained just before Step* 6,

$$-\sum_{i=1}^{v}\lambda\hat{e}_i + \sum_{i=1}^{v}\lambda_i\hat{e}_i + \sum_{p\in M}\frac{1}{d_p^\beta}\hat{e}_p = 0,$$

where λ_i is $\epsilon_i(2\rho)^\beta$, with $|\epsilon_i| < \frac{1}{2v^{1.5}}$, for all i.

Proof. By definition, (2.5), $\left(\nabla f_1 = \sum_{p\in M}\frac{1}{d_p^\beta}\hat{e}_p - \sum_{i=1}^{v}\lambda\hat{e}_i\right)$. According to Corollary 7.1,

$$\left|\frac{\partial f_1}{\partial u_i}\right| < \sqrt{\frac{12288\beta^2\rho^2v^2}{2^\beta}}\,(2\rho)^\beta.$$

Therefore since $\beta = 8\lceil\log_2 v\rceil + 1$ and $\lambda = (2\rho)^\beta$, the corollary follows. □

COROLLARY 7.3. *At the point F obtained just before Step 6 of the algorithm in §4, the distance of the closest hyperplane of the matching polytope, $d_{p,\min}$, is less than $\frac{1}{\rho}$.*

Proof. Assume the closest hyperplane to be at a distance greater than $\frac{1}{\rho}$. Then from the definition of f_1 in (2.5), $(\nabla f_1 = \sum_{p\in M} \frac{1}{d_p^\beta}\hat{e}_p - \sum_{i=1}^{v} \lambda\hat{e}_i)$ and the assumption that all $d_p > \frac{1}{\rho}$, it follows that $\left|\frac{\partial f_1}{\partial u_i}\right| > (2\rho)^\beta - \frac{v}{2}\rho^\beta$. Since $2^\beta >> \frac{v}{2}$, this is not possible according to Corollary 7.1. □

A key feature of gradients that vary as $\frac{1}{d_p^\beta}$ is that the gradient exerted by the hyperplanes is dominated by those very close to the center. In case the closest hyperplane is at a distance $d_{p,\min}$, the gradients due to all hyperplanes at a distance greater than $2d_{p,\min}$ can be neglected. In fact, it is proved that the combined gradient due to all hyperplanes further than $2d_{p,\min}$ is considerably smaller than that due to a single hyperplane at a distance $d_{p,\min}$.

THEOREM 8. *For the dual of the matching problem, (2.3), assume $-\sum_{i=1}^{v} \lambda\hat{e}_i - \sum_{i=1}^{v} \lambda_i\hat{e}_i + \sum_{p\in M} \frac{w_p}{d_p^\beta}\hat{e}_p = 0$, where $\frac{1}{4} \le w_p \le 1$, $\lambda = (2\rho)^\beta$ and $\lambda_i = \epsilon_i(2\rho)^\beta$, where $|\epsilon_i| < \frac{1}{2v^{1.5}}$. Next the weights w_p are changed to w'_p so that all of the hyperplanes that have $d_p \ge 2d_{p,\min}$ have weights $w'_p = 0$ and the weights of hyperplanes with $d_p \le 2d_{p,\min}$ stay the same, i.e., $w'_p = w_p$, then in order to satisfy the condition $-\sum_{i=1}^{v} \lambda\hat{e}_i - \sum_{i=1}^{v} \lambda'_i\hat{e}_i + \sum_{p\in M} \frac{w'_p}{d_p^\beta}\hat{e}_p = 0$, λ'_i will be $\epsilon'_i(2\rho)^\beta$, where $|\epsilon'_i| < \frac{1}{v^{1.5}}$.*

Proof. The gradients due to each hyperplane of the matching polytope $(\frac{w_p}{d_p^\beta}\hat{e}_p)$ is in the positive i direction for all coordinate directions i according to the definition of the matching polytope (2.3). Therefore from the condition that at least one hyperplane is closer than $d_{p,\min}$, it follows that $\lambda \ge \frac{1}{2}\left(\frac{1}{d_{p,\min}}\right)^\beta$. The gradients due to hyperplanes further away than $\frac{1}{2d_{p,\min}}$ contribute at most $\frac{1}{2^\beta d_{p,\min}^\beta}$ each to the gradient in the i direction. Therefore the total contribution to the gradient in the i direction due to these hyperplanes is at most $\frac{v}{2^\beta d_{p,\min}^\beta}$ which is less than $\frac{2\lambda v}{2^\beta}$. Therefore if the w_p of hyperplanes further away than $2d_{p,\min}$ is reduced to zero and if w'_p be the new weights: $-\sum_{i=1}^{v} \lambda\hat{e}_i + \sum_{i=1}^{v} \lambda'_i\hat{e}_i + \sum_{p\in M} \frac{w_p}{d_p^\beta}\hat{e}_p = 0$, and since $\beta = 8\lceil \log_2 v\rceil + 1$ it follows that $|\lambda'_i| < \frac{\lambda}{v^2}$. □

Our real interest lies in obtaining a solution to the primal problem (2.2). We next show how to transfer from the dual to the primal problem. In general it is not possible to obtain an interior point in the primal polytope from an arbitrary interior point of the dual; however, there does exist a correspondence between points on the central path for the logarithmic barrier potential as was pointed out by Renegar [15]. We first transfer from the center of the nonlinear potential function to the center of a slightly perturbed logarithmic barrier function from which we obtain a point in the primal polytope. The analysis has been reworked from first principles in our framework for the particular case of the bipartite matching problem.

THEOREM 9. *Consider the polytope defined by* (5.9.1):

$$\begin{aligned} u_i &\ge 0, \quad i = 1, 2, \ldots, v, \\ u_i + u_j - \Omega_{ij} &\ge 0, \quad \text{where } (i,j) \in E. \end{aligned} \tag{5.9.1}$$

In case at an interior point of (5.9.1), *an external gradient $\bar{F}$ given by $(-q_1, -q_2, \ldots)$*

be balanced by gradients varying inversely with the distance from hyperplanes, i.e.,

$$\bar{F} + \sum_p \frac{\omega_p \hat{e}_p}{d_p} = 0, \quad \omega_p > 0, \tag{5.9.2}$$

where ω_p *for the hyperplanes* $u_i \geq 0$ *is* ω_i *and for the hyperplanes* $u_i + u_j - \Omega_{ij} \geq 0$ *is* ω_{ij}. *Then consider the dual polytope of* (5.9.1):

$$\begin{aligned} f_{ij} &\geq 0, \quad (i,j) \in E, \\ \sum_{\substack{j \in v \\ (i,j) \in E}} f_{ij} &\leq q_i, \quad i = 1, 2, \ldots, v. \end{aligned} \tag{5.9.3}$$

At the interior point of (5.9.3), *given by* $f_{ij} = \frac{\omega_{ij}}{u_i + u_j - \Omega_{ij}}$, *the following vector equality is satisfied:*

$$\bar{F}' + \sum_p \frac{\omega_p \hat{e}_p}{d_p} = 0. \tag{5.9.4}$$

The component of F' *along the* f_{ij} *direction is given by* Ω_{ij} *and the summation is over the hyperplanes of* (5.9.3) *and* ω_p *for the hyperplanes* $f_{ij} \geq 0$ *is* ω_{ij} *and for the hyperplanes* $\sum_{\substack{j \in V \\ (i,j) \in E}} f_{ij} \leq q_i$ *is* ω_i.

Proof. Taking the component of (5.9.2) in the i direction gives

$$\frac{\omega_i}{u_i} + \sum_j \frac{\omega_{ij}}{u_i + u_j - \Omega_{ij}} - q_i = 0. \tag{5.9.5}$$

Next define new variables f_{ij} for every $(i,j) \in E$ which are assigned values as follows:

$$f_{ij} = \frac{\omega_{ij}}{u_i + u_j - \Omega_{ij}}. \tag{5.9.6}$$

Using (5.9.6), the equation (5.9.5) may be written as

$$\left(- \sum_{\substack{j \in V \\ ((i,j) \in E)}} f_{ij} + q_i \right) u_i = \omega_i. \tag{5.9.7}$$

Substituting u_i and u_j from (5.9.7) in (5.9.6), we obtain:

$$\frac{\omega_i}{\left(- \sum_{\substack{x \in V \\ ((i,x) \in E)}} f_{ix} + q_i \right)} + \frac{\omega_j}{\left(- \sum_{\substack{x \in V \\ ((x,j) \in E)}} f_{xj} + q_j \right)} - \Omega_{ij} = \frac{\omega_{ij}}{f_{ij}}. \tag{5.9.8}$$

This is the same as the ij component of the vector equation (5.9.4). □

Next we bound the distance between the center and the optimal vertex in terms of the gradient—this can be conveniently done for the logarithmic barrier potential function. It is shown in Theorem 11 how to obtain the center of the logarithmic barrier function of a slightly modified problem (a weighted matching problem), from the center of the nonlogarithmic potential function as obtained just before Step 6; Theorem 10 will then be used to prove the goodness of the result.

THEOREM 10. *Let $\bar{d_0}$ be the vector connecting the center of the logarithmic barrier function to the optimum point and d_0 be the component of $\bar{d_0}$ in the direction of the gradient of the objective function. In case $\overline{F} + \sum \frac{\omega_p \hat{e}_p}{d_p} = 0$, with $\omega_p > 0$ for all hyperplanes, then d_0 is upper bounded by*

$$d_0 \leq \frac{\sum_p \omega_p}{|F|}. \tag{5.10.0}$$

Proof. Taking the dot product of the vector equation $\overline{F} + \sum \frac{\omega_p \hat{e}_p}{d_p} = 0$ with $\overline{d_0}$, i.e., the vector connecting the center to the optimal vertex, we obtain

$$\overline{F} \cdot \overline{d_0} = \sum_p \omega_p \left(-\frac{\hat{e}_p \cdot \overline{d_0}}{d_p} \right), \tag{5.10.1}$$

since the optimal point lies within the polytope it follows that $(-\hat{e}_p \cdot \overline{d_0}) \leq d_p$. Therefore

$$\overline{F} \cdot \overline{d_0} \leq \sum_p \omega_p. \tag{5.10.2}$$

Equation (5.10.0) immediately follows. □

THEOREM 11. *The point obtained in Step 6 of the algorithm has an objective function value within $\frac{1}{2^\beta}$ of the maximum of the modified objective function defined by $\sum_{\substack{i,j=1 \\ (i,j)\in E}}^{v} \Omega_{ij} f_{ij}$ for some Ω_{ij} in the range $(1 - \frac{4}{\rho})$ to $(1 + \frac{4}{\rho})$, in the polytope (5.11.0):*

$$\begin{aligned} f_{ij} &\geq 0, && (i,j) \in E, \\ \sum_{\substack{j \in v \\ (i,j)\in E}} f_{ij} &\leq (1 - \epsilon_i'), && i = 1, 2, \ldots, v \end{aligned} \tag{5.11.0}$$

where $|\epsilon_i'| < \frac{1}{v^{1.5}}$.

Proof. Consider the LP (2.3) and the final point F in the dual polytope (2.3) obtained just before Step 6. Carry out the following transformations. Let the distance of the nearest hyperplane to F be $d_{p,\min}$. Shift all hyperplanes within $2d_{p,\min}$ of F closer to F so that they are within a distance of ε of F, where $\varepsilon = (\frac{d_{p,\min}}{2})^\beta$ and adjust w_p to a new value w_p' so that $\frac{w_p'}{\varepsilon} = \frac{1}{d_p^\beta}$. Let s_i be the shift of the hyperplane $u_i > 0$ and s_{ij} the shift of the hyperplane $\frac{u_i + u_j}{2} \geq \frac{1}{2}$. Note that since $\varepsilon = (\frac{d_{p,\min}}{2})^\beta$, $w_p' \leq \frac{1}{2^\beta}$. Set w_p' for those hyperplanes further than $2d_{p,\min}$ to 0. Therefore the weights w_p' for all hyperplanes approach zero. Now the hyperplanes are those of the modified polytope

$$\begin{aligned} u_i &\geq s_i, && i = 1, 2, \ldots, v \\ \frac{u_i + u_j}{2} &\geq \frac{1}{2} + s_{ij}, && (i,j) \in E. \end{aligned} \tag{5.11.1}$$

Since by Corollary 7.3, $(d_{p,\min} < \frac{1}{\rho})$, it follows that the shifts s_i and s_{ij} satisfy $(\frac{2}{\rho} \geq s_i, s_{ij} \geq 0)$. Also according to Theorem 8, $\overline{F'} + \sum_p \frac{w_p' \hat{e}_p}{d_p} = 0$, where $F_i' =$

$-(2\rho)^{\beta}(1-\epsilon_i')$ and $|\epsilon_i'| < \frac{1}{v^{1.5}}$. Define new variables u_i' so that $u_i' = u_i - s_i$. The polytope now becomes

$$\begin{aligned} u_i' &\geq 0, \qquad i = 1, 2, \ldots, v \\ \frac{u_i' + u_j'}{2} &\geq \frac{1}{2} + s_{ij}', \quad (i,j) \in E, \quad s_{ij}' = s_{ij} - \frac{s_i}{2} - \frac{s_j}{2}, \end{aligned} \tag{5.11.2}$$

s_{ij}' satisfy $(\frac{2}{\rho} \geq s_{ij}' \geq -\frac{2}{\rho})$. Now using Theorem 9, we obtain that in the primal framework, the following polytope corresponds to (5.11.2):

$$\begin{aligned} f_{ij} &\geq 0, \qquad (i,j) \in E, \\ \sum_{\substack{j \in v \\ (i,j) \in E}} f_{ij} &\leq (2\rho)^{\beta}(1-\epsilon_i'), \quad i = 1, 2, \ldots, v. \end{aligned} \tag{5.11.3}$$

By Theorem 9, the following vector equation is satisfied by the system of planes (5.11.3): $\overline{F'''} + \sum \frac{w_p' \hat{e}_p}{d_p} = 0$, $w_p' \leq \frac{1}{2^{\beta}}$, where $F_{ij}'' = \Omega_{ij}$ where $\Omega_{ij} = 1 + 2s_{ij}'$. Scaling the variables f_{ij} by $(2\rho)^{-\beta}$ and F_{ij}'' by $(2\rho)^{\beta}$, we obtain the polytope

$$\begin{aligned} f_{ij} &\geq 0, \qquad (i,j \in E), \\ \sum_{\substack{j \in v \\ (i,j) \in E}} f_{ij} &\leq (1-\epsilon_i'), \quad i = 1, 2, \ldots, v. \end{aligned} \tag{5.11.4 same as (5.11.0)}$$

The equation of the gradient scales to $\overline{F'''} + \sum_p \frac{w_p' \hat{e}_p}{d_p} = 0$ with $F_{ij}'' = (2\rho)^{\beta}\Omega_{ij}$. Now using the fact that $w_p' \leq \frac{1}{2^{\beta}}$, it follows from Theorem 10 that $d_0 < \frac{1}{2^{\beta}}$. Thus f_{ij} derived in Step 6 is within $\frac{1}{2^{\beta}}$ of the maximum of the modified objective function

$$\sum_{\substack{i,j=1 \\ (i,j) \in E}}^{v} \Omega_{ij} f_{ij}$$

where $(1 - \frac{4}{\rho}) < \Omega_{ij} < (1 + \frac{4}{\rho})$ in the modified polytope described by (5.11.0). □

Next we prove that at f_{ij} obtained by the algorithm in Step 6 the objective function is within a factor of $\left(1 - \frac{2}{v^{1.5}}\right)\left(1 - \frac{8}{\rho}\right)$ of its optimal value in the polytope (5.11.0).

COROLLARY 11.1. *Let $f_{ij,0}$ be the point obtained by Step 6 of the algorithm. Then at $f_{ij,0}$ the objective function $\sum_{\substack{i,j=1 \\ (i,j) \in E}}^{v} f_{ij}$ is within a factor of $(1 - \frac{2}{v^{1.5}})(1 - \frac{8}{\rho})$ of its maximum value in the polytope* (2.2).

Proof. According to Theorem 11, at $f_{ij,0}$ the objective function $\sum_{\substack{i,j=1 \\ (i,j) \in E}}^{v} \Omega_{ij} f_{ij}$ has a value within $\frac{1}{2^{\beta}}$ of its maximum in the polytope (5.11.0). Let the maximum of $\sum_{\substack{i,j=1 \\ (i,j) \in E}}^{v} f_{ij}$ in the polytope (2.2) be at the point $f_{ij,1}$. Since $(1 - |\epsilon_i'|_{\max}) f_{ij,1}$ is clearly an interior point of (5.11.0) it follows that

$$\sum_{\substack{i,j=1 \\ (i,j) \in E}}^{v} \Omega_{ij} f_{ij,0} \geq (1 - |\epsilon_i'|_{\max}) \sum_{\substack{i,j=1 \\ (i,j) \in E}}^{v} \Omega_{ij} f_{ij,1} - \frac{1}{2^{\beta}}. \tag{5.11.5}$$

Since $(f_{ij,0} \geq 0)$ and $(f_{ij,1} \geq 0)$, it follows from (5.11.5) that

$$(\Omega_{ij})_{\max} \sum_{\substack{i,j=1 \\ (i,j)\in E}}^{v} f_{ij,0} \geq (1 - |\epsilon_i'|_{\max})\,(\Omega_{ij})_{\min} \sum_{\substack{i,j=1 \\ (i,j)\in E}}^{v} f_{ij,1} - \frac{1}{2^{\beta}}. \tag{5.11.6}$$

Using Theorem 11, $\left(|\epsilon_i'|_{\max} < \frac{1}{v^{1.5}}\right)$, the fact that $(\beta = 8\lceil \log_2 v \rceil + 1)$ and substituting bounds from Theorem 11 for Ω_{ij} in (5.11.6):

$$\sum_{\substack{i,j=1 \\ (i,j)\in E}}^{v} f_{ij,0} \geq \left(1 - \frac{2}{v^{1.5}}\right)\left(1 - \frac{8}{\rho}\right) \sum_{\substack{i,j=1 \\ (i,j)\in E}}^{v} f_{ij,1}. \tag{5.11.7}$$ □

COROLLARY 11.2. *Let $f'_{ij,0}$ be the point obtained by Step 7 of the algorithm. Then at $f'_{ij,0}$ the objective function $\sum^{v}_{\substack{i,j=1 \\ (i,j)\in E}} f_{ij}$ is within a factor of $(1 - \frac{9}{\rho})$ of its maximum value in the polytope* (2.2).

Proof. Let $f_{ij,0}$ be the point obtained by Step 6 and the maximum of $\sum^{v}_{\substack{i,j=1 \\ (i,j)\in E}} f_{ij}$ in the polytope (2.2) be at the point $f_{ij,1}$. Since according to Theorem 11, the point obtained in Step 6 of the algorithm is an interior point of (5.11.0), it follows that the reduction in each f_{ij} in Step 7 is at most by a factor of $(1 - |\epsilon_i'|_{\max})$. Therefore using this observation along with Corollary 11.1, it follows that

$$\sum_{\substack{i,j=1 \\ (i,j)\in E}}^{v} f'_{ij,0} \geq (1 - |\epsilon_i'|_{\max}) \sum_{\substack{i,j=1 \\ (i,j)\in E}}^{v} f_{ij,0} \geq \left(1 - \frac{2}{v^{1.5}}\right)^2 \left(1 - \frac{8}{\rho}\right) \sum_{\substack{i,j=1 \\ (i,j)\in E}}^{v} f_{ij,1}. \tag{5.11.8}$$

Assuming $v >> \rho$, the desired result follows. □

7. Concluding remarks. This paper demonstrates the use of a nonlogarithmic potential function $f = \sum_p \frac{1}{d_p^{\beta}}$ with $\beta = O(\log n)$ for LP. It is a property of this potential function that once the center is obtained, a constant fractional improvement may be immediately obtained in the objective function in contrast to the $O\left(\frac{1}{\sqrt{m}}\right)$ improvement obtained by using the logarithmic barrier function. The difficult part of the algorithm is to obtain the center. For general LP it takes $O(\sqrt{m})$ matrix inversions to find the center of this nonlogarithmic potential function with the result that the overall complexity of the general algorithm is no better than that of standard path following algorithms using the logarithmic barrier function. When applied to bipartite matching, certain special properties of the problem can be used to obtain the center. This enables us to find an approximate matching in which the number of edges is within a factor of $(1 - \frac{1}{\rho})$ of the optimal matching in $O^*(\rho)$ time and the optimal matching in $O^*\left(v^{1/2}\right)$ time.

We conclude with two open questions.

1. Can the technique of nonlogarithmic potential functions be extended to general LP? In case an N-R scheme could be used to follow the center it would probably lead to a parallel algorithm for general LP that took $O^*(L)$ matrix inversions. However, unlike the log barrier function the dual does not have good properties and it becomes difficult to apply an N-R scheme.

2. Can the algorithm be modified to yield an $O^*(1)$ algorithm, at least for bipartite matching? The problem here is the same as in question 1 above; it is difficult to apply

an N-R type scheme, except for the special case considered in this paper when the center is surrounded by a cube.

Acknowledgments. The author is grateful to Pravin Vaidya and David Atkinson for going through the paper in detail and making several constructive suggestions.

REFERENCES

[1] A. AGGARWAL AND R. J. ANDERSON, *A* RNC *algorithm for* DFS, Proc. 19th Annual ACM Symp. on Theory of Computing, 1987, pp. 325–334.

[2] A. AGGARWAL, R.J. ANDERSON, AND M.Y. KAO, *Parallel* DFS *in general directed graphs*, Proc. 21st Annual ACM Symp. on Theory of Computing, 1989, pp. 297–308.

[3] S. EVEN, *Graph Algorithms*, Computer Science Press, New York, 1979, pp. 135–138.

[4] A. GOLDBERG, S.A. PLOTKIN, AND P. VAIDYA, *Sublinear-time parallel algorithms for matching and related problems*, Proc. 29th IEEE Symp. on Foundations of Computer Science, 1988, pp. 175–185.

[5] A. GOLDBERG, S.A. PLOTKIN, D.B. SHMOYS, AND E. TARDOS, *Interior-Point Methods in Parallel Computation*, Technical Memo., May 1989, Dept. of Computer Science, Stanford University, Stanford, CA.

[6] D.Y. GRIGORIEV AND M. KARPINSKI, *The matching problem for graphs with polynomially bounded permanents is in* NC, Proc. 28th IEEE Annual Symp. on Foundations of Computer Science, 1987, pp. 166–172.

[7] L. K. GROVER, *A force based approach to interior point methods*, Tech. Report, February 89, Electrical Engineering, Cornell University, Ithaca, NY.

[8] N. KARMARKAR, *A new polynomial time algorithm for linear programming*, Combinatorica 4 (1984), pp. 373–395.

[9] R. M. KARP AND A. WIGDERSON, *A fast parallel algorithm for the maximum independent set problem*, J. ACM, 32 (1985), pp. 762–773.

[10] R. M. KARP, E. UPFAL, AND A. WIGDERSON, *Constructing a maximum matching is in* RNC, Combinatorica 6 (1986), pp. 35–48.

[11] G.F. LEV, N. PIPPENGER, AND L. G. VALIANT, *A fast parallel algorithm for routing in permutation networks*, IEEE Trans. Comput., C-30 (1981), pp. 93–100.

[12] G.L. MILLER AND J. NAOR, *Flow in planar graphs with multiple sources and sinks*, Proc. 30th IEEE Annual Symp. on Foundations of Computer Science, 1989, pp. 166–172.

[13] K. MULMULEY, U.V. VAZIRANI, AND V.V. VAZIRANI, *Matching is as easy as matrix inversion*, Combinatorica, 7 (1987), pp. 105–131.

[14] C.H. PAPADIMITRIOU AND K. STEIGLITZ, *Combinatorial Optimization, Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, NJ, 1982.

[15] J. RENEGAR, *A polynomial–time algorithm based on Newton's method for linear programming*, Math. Programming 40 (1988), pp. 59–93.

[16] P.M. VAIDYA, *Reducing the parallel complexity of certain linear programming problems*, Proc. 22nd Ann. ACM Symp. Theory of Computing 1990, pp. 583–589.

SIAM J. OPTIMIZATION
Vol. 5, No. 4, pp. 770–791, November 1995

003

CONVERGENCE OF A FACTORIZED BROYDEN-LIKE FAMILY FOR NONLINEAR LEAST SQUARES PROBLEMS*

HIROSHI YABE† AND NAOKAZU YAMAKI‡

Abstract. This paper is concerned with factorized quasi-Newton methods for nonlinear least squares problems. A one parameter class of symmetric positive definite quasi-Newton updates is given that corresponds to the Broyden family. We call this new class of update formula a factorized Broyden-like family. This family is based on the full rank factorized form of a structured quasi-Newton update. We prove that a quasi-Newton method using the factorized Broyden-like family possesses local and q-superlinear convergence properties.

Key words. nonlinear least squares, quasi-Newton method, factorized Broyden-like family, local and q-superlinear convergence

AMS subject classifications. 65K05, 49D37, 90C30

1. Introduction.

In this paper, we consider numerical methods for minimizing a sum of squares of nonlinear functions

$$f(x) = \frac{1}{2}\|r(x)\|^2, \tag{1}$$

where $r(x) = (r_1(x), \ldots, r_m(x))^T$, $r_j : R^n \to R$ are twice continuously differentiable for $j = 1, \ldots, m$, $m \geq n$, and $\| \ \|$ denotes the l_2 norm. We will denote by x_* a local minimizer. These types of problems are among the most commonly occurring and important applications of nonlinear optimization.

For general unconstrained minimization problems where the Hessian matrix of the objective function can be calculated, Newton's method can be used. The method constructs a sequence of vectors $\{x_k\}$ such that

$$x_{k+1} = x_k + \alpha_k d_k, \tag{2}$$

where α_k is a scalar steplength and the search direction d_k satisfies the Newton equation

$$\nabla^2 f(x_k) d = -\nabla f(x_k), \tag{3}$$

where ∇f and $\nabla^2 f$ denote the gradient vector and the Hessian matrix of f, respectively.

The gradient vector and the Hessian matrix of the function (1) have special forms, given by

$$\nabla f(x) = J(x)^T r(x) \tag{4}$$

and

$$\nabla^2 f(x) = J(x)^T J(x) + \sum_{j=1}^{m} r_j(x) \nabla^2 r_j(x), \tag{5}$$

* Received by the editors April 1, 1991; accepted for publication (in revised form) June 8, 1994.
† Department of Industrial Management and Engineering, Faculty of Engineering, Science University of Tokyo, 1-3, Kagurazaka, Shinjuku-ku, Tokyo, 162, Japan.
‡ Research Institute of Systems Planning Ltd., 2-9, Sakuragaoka-cho, Shibuya-ku, Tokyo, 150, Japan.

respectively, where $J(x)$ is the $m \times n$ Jacobian matrix of $r(x)$ whose ith row is $\nabla r_i(x)^T$.

Since the complete Hessian matrix (5) is often expensive to compute, methods have been developed that use only the first derivative information. For example, the Gauss–Newton method and the Levenberg–Marquardt method exploit the special structure of the Hessian matrix and gradient vector. Since these methods neglect the second part of the Hessian matrix (5), they can be expected to perform well when the residual $f(x_*)$ is small or the functions r_i are close to linear. These cases are called the small residual problems. However, when the residual $f(x_*)$ is very large and the functions r_i are rather nonlinear, these methods may perform poorly [8]. These cases are called the large residual problems.

Quasi-Newton approximations to only the second part of the Hessian matrix (5) have been developed [9]. These strategies are called the structured quasi-Newton methods. Since the nonlinear least squares algorithms usually calculate the Jacobian matrix $J(x)$ analytically or numerically, the portion $J(x)^T J(x)$ of $\nabla^2 f(x)$ is always readily available, so we only have to approximate the second part of $\nabla^2 f(x)$. Therefore, for the nonlinear least squares problem, the search direction d_k can be computed by solving

$$(J(x_k)^T J(x_k) + A_k)d = -J(x_k)^T r(x_k), \tag{6}$$

where the matrix A_k is the kth approximation to the second part of $\nabla^2 f(x_k)$ so that

$$\nabla^2 f(x_k) \approx J(x_k)^T J(x_k) + A_k.$$

Following Dennis [5], we have

$$\nabla^2 f(x_{k+1}) s_k \approx z_k, \tag{7}$$

where

$$s_k = x_{k+1} - x_k \tag{8}$$

and

$$z_k = J(x_{k+1})^T J(x_{k+1}) s_k + (J(x_{k+1}) - J(x_k))^T r(x_{k+1}). \tag{9}$$

Thus the matrix A_k is updated such that the new matrix A_{k+1} satisfies the secant condition

$$A_{k+1} s_k = (J(x_{k+1}) - J(x_k))^T r(x_{k+1}). \tag{10}$$

Within the preceding framework, Bartholomew-Biggs [2] and Dennis, Gay, and Welsch [10] proposed robust algorithms for both the cases of large and small residual problems. The former dealt with the structured symmetric rank one update for the line search strategy, and the latter dealt with the structured Davidon–Fletcher–Powell (DFP) update for the trust region strategy. After them, Al-Baali and Fletcher [1] and Fletcher and Xu [13] proposed the hybrid method that combined the structured Broyden–Fletcher–Goldfarb–Shanno (BFGS) update and the Gauss–Newton method for the line search strategy. Dennis, Martinez, and Tapia [11] later derived the structure principle, and showed local and q-superlinear convergence of the structured BFGS update. Recently, Engels and Martinez [12] (see also Martinez [15]) extended these updates to the convex class of structured Broyden family and proved local and q-superlinear convergence of their method. Huschens [14] has proposed a new method

based on the extended structure principle and has shown quadratic convergence for zero residual problems and q-superlinear convergence for nonzero residual problems.

In the case of line search descent methods, it is desirable to maintain the positive definiteness of the matrix $J(x_k)^T J(x_k) + A_k$ in (6). This guarantees that d_k is a descent direction for $f(x)$. However it is not clear how to construct update formulae for A_k such that the matrix $J(x_k)^T J(x_k) + A_k$ is positive definite. To overcome this difficulty, factorized versions of structured quasi-Newton methods have been studied by Yabe and Takahashi [19]. They proposed computing the search direction d_k by solving the linear system of equations

$$(J(x_k) + L_k)^T (J(x_k) + L_k)d = -J(x_k)^T r(x_k)$$

instead of (6), where the matrix L_k is an $m \times n$ correction matrix to the Jacobian matrix such that $(J(x_k) + L_k)^T (J(x_k) + L_k)$ is the approximation to $\nabla^2 f(x_k)$. They obtained the BFGS-like and the DFP-like updates for L_k in [19] and showed local and q-superlinear convergence of their methods in [20].

In this paper, we propose a one parameter class of symmetric positive definite quasi-Newton updates that corresponds to the Broyden family. In §2, we derive a Broyden-like family and establish its full rank factorized form. This contains the BFGS-like and the DFP-like updates proposed by Yabe and Takahashi [19]. In §3, we show local and q-superlinear convergence of a quasi-Newton method with the Broyden-like family. In §4, we discuss a line search criterion and apply sizing techniques to our updates.

Throughout this paper, $\| \ \|$ denotes the l_2 norm for vectors or matrices, and $\| \ \|_F$ and $\| \ \|_{F,M}$ denote the Frobenius norm and the weighted Frobenius norm for some nonsingular matrix M, which are defined by

$$\|Q\|_F = \sqrt{\mathrm{Trace}(QQ^T)} \quad \text{and} \quad \|Q\|_{F,M} = \|MQM\|_F,$$

respectively.

2. Factorized Broyden-like family. In this section, we derive a full rank factorized form of structured quasi-Newton updates and obtain a significant class of such updates. Our approach is basically an application of the study by Yabe [17] and Yamaki and Yabe [21] for standard quasi-Newton methods to the question of structured quasi-Newton methods. Consider computing the search direction d_k by solving the linear system of equations

$$(J(x_k) + L_k)^T (J(x_k) + L_k)d = -J(x_k)^T r(x_k). \tag{11}$$

From the factorized form of the coefficient matrix in (11), we see that the search direction d_k will be a descent direction for f. From (7), the secant condition for L_{k+1} is:

$$(J(x_{k+1}) + L_{k+1})^T (J(x_{k+1}) + L_{k+1})s_k = z_k, \tag{12}$$

where s_k and z_k are defined in (8) and (9), respectively. In order to find a matrix L_{k+1} which satisfies the matrix equation (12), we consider the following problem.

PROBLEM A. *Given vectors $a \in R^n$ and $b \in R^n$ that satisfy $a^T b > 0$, find a matrix $N \in R^{m \times n}$ such that $N^T N a = b$.*

Since it is not so easy to solve Problem A directly, we consider the following problem instead.

PROBLEM B. *Given vectors* $a \in R^n$ *and* $b \in R^n$ *that satisfy* $a^T b > 0$, *find a matrix* $N \in R^{m\times n}$ *such that* $N^T h = b$ *and* $Na = h$ *for a vector* $h \in R^m$ *satisfying* $h^T h = a^T b$.

We then have the following lemma.

LEMMA 1. *The solution set of Problem* A *is equivalent to the solution set of Problem* B.

Proof. Denote the solution sets of Problem A and Problem B by

$$S_A = \{N \in R^{m\times n} \mid N^T Na = b\}$$

and

$$S_B = \{N \in R^{m\times n} \mid h^T N = b^T,\ Na = h,\ h^T h = a^T b\},$$

respectively. Let N be any element of the set S_A. Setting $h = Na$ yields

$$h^T N = a^T N^T N = b^T, \quad Na = h \quad \text{and} \quad h^T h = a^T N^T Na = a^T b.$$

This implies $N \in S_B$. Conversely, letting N be any element of the set S_B, we have

$$N^T Na = N^T h = b.$$

This implies $N \in S_A$. Thus the proof is complete. □

In order to obtain a general solution to Problem B, we use the following lemma [4, p. 43].

LEMMA 2. *The matrix equations*

$$CX = D, \quad XE = G$$

have a common solution X *if and only if each equation separately has a solution and*

$$CG = DE.$$

Furthermore, if X_0 *is a common solution of the matrix equations, then the general common solution is*

$$X = X_0 + (I - C^- C)\Phi(I - EE^-),$$

where Φ *is an arbitrary matrix, and* C^- *and* E^- *are any matrices such that* $CC^-C = C$ *and* $EE^-E = E$, *respectively.*

Set

$$X = N, \quad C = h^T, \quad D = b^T, \quad E = a \quad \text{and} \quad G = h.$$

Since $h \neq 0$ and $a \neq 0$, the matrix equations $h^T N = b^T$ and $Na = h$ separately have a solution. Then by Lemma 2, the condition $h^T h = a^T b$ guarantees that Problem B is solvable. Since $hb^T/a^T b$ is a common solution, it follows directly from Lemma 2 that

$$N = \frac{hb^T}{a^T b} + (I - (h^T)^- h^T)\Phi(I - aa^-).$$

Since $h = (\sqrt{a^T b}/\|u\|)u$ satisfies the condition $h^T h = a^T b$ for any nonzero vector $u \in R^m$, setting $(h^T)^- = \Phi a/a^T \Phi^T h$ yields a solution to Problem B as follows:

$$N = \frac{ub^T}{\sqrt{a^T b}\,\|u\|} + \left(I - \frac{\Phi a u^T}{a^T \Phi^T u}\right)\Phi, \tag{13}$$

where $\Phi \in R^{m\times n}$ and $u \in R^m$ are, respectively, any matrix and vector such that $a^T\Phi^T u \neq 0$.

Now we consider finding a matrix L_{k+1} that satisfies the matrix equation (12). Assume that $s_k^T z_k > 0$. It is clear that we can set in Problem A

$$N = J(x_{k+1}) + L_{k+1}, \qquad a = s_k \quad \text{and} \quad b = z_k.$$

Moreover by using the idea of the structure principle in [11, p. 165], we choose

$$\Phi = J(x_{k+1}) + L_k.$$

Then (13) yields

$$L_{k+1} = L_k + \frac{1}{\sqrt{s_k^T z_k}} \frac{u}{\|u\|} z_k^T - \frac{L_k^\sharp s_k u^T L_k^\sharp}{u^T L_k^\sharp s_k}, \tag{14}$$

where u is an arbitrary vector such that $u^T L_k^\sharp s_k \neq 0$ and

$$L_k^\sharp = J(x_{k+1}) + L_k. \tag{15}$$

Setting

$$B_{k+1} = (J(x_{k+1}) + L_{k+1})^T (J(x_{k+1}) + L_{k+1}) \tag{16}$$

and

$$B_k^\sharp = (L_k^\sharp)^T L_k^\sharp, \tag{17}$$

the expression (14) yields

$$B_{k+1} = B_k^\sharp - \frac{B_k^\sharp s_k s_k^T B_k^\sharp}{s_k^T B_k^\sharp s_k} + \frac{z_k z_k^T}{s_k^T z_k} + (s_k^T B_k^\sharp s_k) w_k w_k^T, \tag{18}$$

where

$$w_k = \frac{B_k^\sharp s_k}{s_k^T B_k^\sharp s_k} - \frac{(L_k^\sharp)^T u}{u^T L_k^\sharp s_k}.$$

By using (14) and (18), we would construct a family that corresponds to the Broyden family. For this purpose, we recall the BFGS-like update and the DFP-like update proposed by Yabe and Takahashi [19]. Yabe and Takahashi constructed them by means of the least change secant update technique in the sense of Dennis and Schnabel [7] under the assumption of $s_k^T z_k > 0$. These are as follows.

(i) BFGS-like update.

$$L_{k+1} = L_k + \left(\frac{L_k^\sharp s_k}{s_k^T B_k^\sharp s_k}\right)\left(\sqrt{\frac{s_k^T B_k^\sharp s_k}{s_k^T z_k}}\, z_k - B_k^\sharp s_k\right)^T, \tag{19}$$

$$B_{k+1} = B_k^\sharp - \frac{B_k^\sharp s_k s_k^T B_k^\sharp}{s_k^T B_k^\sharp s_k} + \frac{z_k z_k^T}{s_k^T z_k}. \tag{20}$$

(ii) DFP-like update.

$$L_{k+1} = L_k + L_k^\sharp \left(\sqrt{\frac{s_k^T z_k}{z_k^T (B_k^\sharp)^{-1} z_k}} (B_k^\sharp)^{-1} z_k - s_k \right) \left(\frac{z_k}{s_k^T z_k} \right)^T, \tag{21}$$

$$B_{k+1} = B_k^\sharp + \left(1 + \frac{s_k^T B_k^\sharp s_k}{s_k^T z_k} \right) \frac{z_k z_k^T}{s_k^T z_k} - \frac{B_k^\sharp s_k z_k^T + z_k s_k^T B_k^\sharp}{s_k^T z_k}. \tag{22}$$

We emphasize that the vector

$$u = \frac{L_k^\sharp s_k}{s_k^T B_k^\sharp s_k} \tag{23}$$

in (14) yields the BFGS-like update (19). Setting

$$u = \frac{L_k^\sharp (B_k^\sharp)^{-1} z_k}{s_k^T z_k} \tag{24}$$

in (14) yields the DFP-like update (21). Since the standard Broyden family is formed by a linear combination of the standard BFGS update and the standard DFP update, we can expect to obtain a Broyden-like family by replacing the vector u in (14) by a linear combination of (23) and (24). For $\phi_k \geq 0$, consider

$$u = (1 - \sqrt{\phi_k}) \frac{L_k^\sharp s_k}{s_k^T B_k^\sharp s_k} + \sqrt{\phi_k} \frac{L_k^\sharp (B_k^\sharp)^{-1} z_k}{s_k^T z_k}. \tag{25}$$

Then we have

$$w_k = \frac{B_k^\sharp s_k}{s_k^T B_k^\sharp s_k} - \frac{(1 - \sqrt{\phi_k}) \frac{B_k^\sharp s_k}{s_k^T B_k^\sharp s_k} + \sqrt{\phi_k} \frac{z_k}{s_k^T z_k}}{(1 - \sqrt{\phi_k}) + \sqrt{\phi_k}} = \sqrt{\phi_k} \left(\frac{B_k^\sharp s_k}{s_k^T B_k^\sharp s_k} - \frac{z_k}{s_k^T z_k} \right).$$

Therefore, the update (18) is reduced to

$$B_{k+1} = B_k^\sharp - \frac{B_k^\sharp s_k s_k^T B_k^\sharp}{s_k^T B_k^\sharp s_k} + \frac{z_k z_k^T}{s_k^T z_k} + \phi_k (s_k^T B_k^\sharp s_k) v_k v_k^T, \tag{26}$$

where

$$v_k = \frac{B_k^\sharp s_k}{s_k^T B_k^\sharp s_k} - \frac{z_k}{s_k^T z_k}. \tag{27}$$

Since the expression (26) is a Broyden family from $B_k^\sharp$ to B_{k+1}, we call (26) a Broyden-like family. In this case, we call an update for L_k corresponding to (26) a factorized Broyden-like family. Defining

$$\lambda_k = \frac{1}{(1 - \phi_k) \frac{s_k^T z_k}{s_k^T B_k^\sharp s_k} + \phi_k \frac{z_k^T (B_k^\sharp)^{-1} z_k}{s_k^T z_k}},$$

we have

$$\|u\|^2 = (1 - \phi_k) \frac{1}{s_k^T B_k^\sharp s_k} + \phi_k \frac{z_k^T (B_k^\sharp)^{-1} z_k}{(s_k^T z_k)^2} = \frac{1}{\lambda_k s_k^T z_k}.$$

Note that the Cauchy–Schwarz inequality guarantees

$$\frac{1}{\lambda_k} = \frac{s_k^T z_k}{s_k^T B_k^\sharp s_k} + \phi_k \frac{(s_k^T B_k^\sharp s_k)(z_k^T (B_k^\sharp)^{-1} z_k) - (s_k^T z_k)^2}{(s_k^T z_k)(s_k^T B_k^\sharp s_k)} > 0 \quad \text{for } \phi_k \geq 0.$$

Thus, by substituting (25) into (14), we obtain a factorized Broyden-like family:

$$L_{k+1} = L_k + (1 - \sqrt{\phi_k}) \left(\frac{L_k^\sharp s_k}{s_k^T B_k^\sharp s_k} \right) (\sqrt{\lambda_k} z_k - B_k^\sharp s_k)^T + \sqrt{\phi_k}\, \mathrm{L}_k^\sharp (\sqrt{\lambda_k} (B_k^\sharp)^{-1} z_k - s_k) \left(\frac{z_k}{s_k^T z_k} \right)^T, \tag{28}$$

where

$$\phi_k \geq 0, \qquad \lambda_k = \frac{1}{(1 - \phi_k) \dfrac{s_k^T z_k}{s_k^T B_k^\sharp s_k} + \phi_k \dfrac{z_k^T (B_k^\sharp)^{-1} z_k}{s_k^T z_k}}, \tag{29}$$

and the matrices $L_k^\sharp$ and $B_k^\sharp$ are defined in (15) and (17). It is clear that the cases $\phi_k = 0$ and $\phi_k = 1$ in (28) are equivalent to the factorized BFGS-like update (19) and the factorized DFP-like update (21), respectively.

3. Local and q-superlinear convergence. Our objective in this section is to show that the factorized Broyden-like family given in the previous section satisfies the bounded deterioration principle given by Broyden, Dennis, and Moré [3], and that the method has the local convergence property. For this purpose, we will use the same techniques as Stachurski [16]. Furthermore, q-superlinear convergence of our method follows from the Dennis–Moré characterization [6].

Let D be an open convex subset of R^n, which contains a local minimizer x_*. We assume the following "standard" conditions.

Assumption A1. There exist positive constants ξ_1, ξ_2, and p such that

$$\|\nabla^2 f(x) - \nabla^2 f(x_*)\| \leq \xi_1 \|x - x_*\|^p, \tag{30}$$

$$\|J(x) - J(\widehat{x})\| \leq \xi_2 \|x - \widehat{x}\|^p \tag{31}$$

for any x and $\widehat{x}$ in D.

Assumption A2. $\nabla^2 f$ is symmetric positive definite at x_*.

It follows easily from Assumption A1 that

$$\|\nabla f(x) - \nabla f(\widehat{x}) - \nabla^2 f(x_*)(x - \widehat{x})\| \leq \xi_1 (\max(\|x - x_*\|,\ \|\widehat{x} - x_*\|))^p \|x - \widehat{x}\| \tag{32}$$

and

$$\|J(x)\| \leq \xi_2 \|x - x_*\|^p + \|J(x_*)\|. \tag{33}$$

First we have the following result. This result corresponds to Lemma 2.4 in [16].

LEMMA 3. *Assume that $B_k^\sharp$ is invertible and B_{k+1} in* (26) *is well defined. Let*

$$H_k^\sharp = (B_k^\sharp)^{-1}. \tag{34}$$

Then B_{k+1} is invertible and its inverse matrix can be represented by

$$H_{k+1} = H_{k+1}^{\mathrm{BFGS}} + (1-\psi_k)\Delta H_k, \tag{35}$$

where

$$H_{k+1}^{\mathrm{BFGS}} = H_k^\sharp + \frac{(s_k - H_k^\sharp z_k)s_k^T + s_k(s_k - H_k^\sharp z_k)^T}{s_k^T z_k} - \frac{z_k^T(s_k - H_k^\sharp z_k)}{(s_k^T z_k)^2} s_k s_k^T, \tag{36}$$

$$\Delta H_k = -(z_k^T H_k^\sharp z_k)\left(\frac{s_k}{s_k^T z_k} - \frac{H_k^\sharp z_k}{z_k^T H_k^\sharp z_k}\right)\left(\frac{s_k}{s_k^T z_k} - \frac{H_k^\sharp z_k}{z_k^T H_k^\sharp z_k}\right)^T, \tag{37}$$

a parameter ψ_k is given by

$$\psi_k = (1-\phi_k)\lambda_k \frac{s_k^T z_k}{s_k^T B_k^\sharp s_k}, \tag{38}$$

and ϕ_k, λ_k are given in (29).

Proof. The result follows directly from $B_{k+1}H_{k+1} = I$. □

Furthermore, we have the following lemma concerning the parameter ψ_k.

LEMMA 4. *Assume that $\phi_k \geq 0, s_k^T z_k > 0$ and the matrix $B_k^\sharp$ is positive definite. Let ψ_k be defined by* (38). *Then*

$$\min(0, 1-\phi_k) \leq \psi_k \leq 1. \tag{39}$$

Proof. Setting

$$a = \frac{s_k^T z_k}{s_k^T B_k^\sharp s_k} \quad \text{and} \quad b = \frac{z_k^T (B_k^\sharp)^{-1} z_k}{s_k^T z_k},$$

and using (29), we have

$$\psi_k = (1-\phi_k)\lambda_k \frac{s_k^T z_k}{s_k^T B_k^\sharp s_k} = \frac{(1-\phi_k)a}{(1-\phi_k)a + \phi_k b}. \tag{40}$$

The Cauchy–Schwarz inequality yields

$$b - a = \frac{(s_k^T B_k^\sharp s_k)(z_k^T (B_k^\sharp)^{-1} z_k) - (s_k^T z_k)^2}{(s_k^T z_k)(s_k^T B_k^\sharp s_k)} \geq 0, \qquad \text{i.e.,} \qquad \frac{b}{a} \geq 1. \tag{41}$$

Thus if $0 \leq \phi_k \leq 1$, then

$$0 \leq \psi_k \leq \frac{(1-\phi_k)a}{(1-\phi_k)a + \phi_k a} \leq 1.$$

If $\phi_k > 1$, then

$$1 - \phi_k \leq \frac{1-\phi_k}{1 + \phi_k(\frac{b}{a} - 1)} = \psi_k < 1.$$

Therefore the proof is complete. □

The preceding lemma implies that if ϕ_k is bounded, then the parameter ψ_k is also bounded. This fact is essentially used in the proof of the convergence theorem below. In particular note that the convex class, i.e., $0 \leq \phi_k \leq 1$, implies $0 \leq \psi_k \leq 1$. Most of the lemmas in [16] can be applied to our proof, since we use the form (35) in order to show local convergence of the factorized Broyden-like family. The significant difference between our proof and that of Stachurski is that we will deal with the factorized form (28) to obtain estimates on matrices $L_k, B_k^\sharp$, and $H_k^\sharp$. In what follows, define

$$M = \nabla^2 f(x_*)^{\frac{1}{2}} \tag{42}$$

and

$$\sigma_k = \max(\|x_{k+1} - x_*\| \, , \, \|x_k - x_*\|). \tag{43}$$

Set

$$\widehat{H}_k^\sharp = M H_k^\sharp M, \quad \widehat{z}_k = M^{-1} z_k, \quad \widehat{s}_k = M s_k \quad \text{and} \quad \widehat{H}_{k+1} = M H_{k+1} M. \tag{44}$$

Note that by the equivalence of norms, for any $n \times n$ matrix C, there exist positive constants η_1 and η_2 such that

$$\frac{1}{\eta_1}\|C\|_{F,M} \leq \|C\| \leq \eta_2 \|C\|_{F,M}. \tag{45}$$

We prepare for the main theorem by establishing several lemmas.

LEMMA 5. *Suppose that Assumptions* A1 *and* A2 *hold. Assume that* $x_k, x_{k+1} \in D$, *and for* ε *small,*

$$\|x_k - x_*\| \leq \varepsilon \quad \text{and} \quad \|x_{k+1} - x_*\| \leq \varepsilon.$$

Then

$$\|z_k - \nabla^2 f(x_*) s_k\| \leq \zeta \sigma_k^p \|s_k\|,$$

and

$$(1 - \rho^\sharp)\|M s_k\|^2 \leq s_k^T z_k \leq (1 + \rho^\sharp)\|M s_k\|^2,$$

where

$$\zeta = \xi_1 + \frac{2^p(p+2)}{p+1}\xi_2(\xi_2 \varepsilon^p + \|J(x_*)\|) \qquad \textit{and} \qquad \rho^\sharp = \|M^{-1}\|^2 \zeta \varepsilon^p.$$

Proof. It follows from (32) that

$$\|\nabla f(x_{k+1}) - \nabla f(x_k) - \nabla^2 f(x_*) s_k\| \leq \xi_1 \sigma_k^p \|s_k\|.$$

Thus, using the mean value theorem, we have

$$r(x_{k+1}) - r(x_k) = \int_0^1 J(x_k + t s_k) s_k \, dt.$$

The definition of z_k yields

$$\begin{aligned}
&\|z_k - \nabla^2 f(x_*)s_k\| \\
&\quad \le \|\nabla f(x_{k+1}) - \nabla f(x_k) - \nabla^2 f(x_*)s_k\| + \|J(x_k)^T\|\,\|r(x_{k+1}) - r(x_k) - J(x_{k+1})s_k\| \\
&\qquad\quad + \|(J(x_{k+1}) - J(x_k))^T\|\,\|J(x_{k+1})\|\,\|s_k\| \\
&\quad \le \left(\xi_1 + \frac{2^p(p+2)}{p+1}\xi_2(\xi_2\varepsilon^p + \|J(x_*)\|)\right)\sigma_k^p\|s_k\| \\
&\quad = \zeta\sigma_k^p\|s_k\|.
\end{aligned}$$

This implies the first result of this lemma.

The second part follows directly from the result (a) of Lemma 4.2 in [3]. Therefore the proof is complete. □

LEMMA 6. *Assume that, for some positive constants $\delta^\sharp$ and $K^\sharp$,*

$$\|H_k^\sharp - \nabla^2 f(x_*)^{-1}\|_{F,M} \le \delta^\sharp \qquad \text{and} \qquad \|H_k^\sharp\| \le K^\sharp.$$

Suppose that the assumptions of Lemma 5 hold and $0 < \rho^\sharp < 1$ and $\varepsilon < 1$. Then

$$\|H_{k+1}^{\mathrm{BFGS}} - \nabla^2 f(x_*)^{-1}\|_{F,M} \le \|H_k^\sharp - \nabla^2 f(x_*)^{-1}\|_{F,M} - \frac{\|\widehat{s}_k - \widehat{H}_k^\sharp \widehat{s}_k\|^2}{2\delta^\sharp\|\widehat{s}_k\|^2} + W\sigma_k^p,$$

where W is some positive constant.

Proof. We use the same estimate as Lemma 3.5 in Stachurski [16]. It follows from (36) and (44) that

$$M(H_{k+1}^{\mathrm{BFGS}} - \nabla^2 f(x_*)^{-1})M = \left(I - \frac{\widehat{s}_k\widehat{s}_k^T}{\|\widehat{s}_k\|^2}\right)(\widehat{H}_k^\sharp - I)\left(I - \frac{\widehat{s}_k\widehat{s}_k^T}{\|\widehat{s}_k\|^2}\right) + T_k,$$

where

$$\begin{aligned}
T_k &= -\frac{\widehat{s}_k\widehat{z}_k^T\widehat{H}_k^\sharp + \widehat{H}_k^\sharp\widehat{z}_k\widehat{s}_k^T}{\widehat{s}_k^T\widehat{z}_k} + \frac{\widehat{s}_k\widehat{s}_k^T\widehat{H}_k^\sharp + \widehat{H}_k^\sharp\widehat{s}_k\widehat{s}_k^T}{\|\widehat{s}_k\|^2} \\
&\quad + \left(1 + \frac{\widehat{z}_k^T\widehat{H}_k^\sharp\widehat{z}_k}{\widehat{s}_k^T\widehat{z}_k}\right)\frac{\widehat{s}_k\widehat{s}_k^T}{\widehat{s}_k^T\widehat{z}_k} - \left(1 + \frac{\widehat{s}_k^T\widehat{H}_k^\sharp\widehat{s}_k}{\|\widehat{s}_k\|^2}\right)\frac{\widehat{s}_k\widehat{s}_k^T}{\|\widehat{s}_k\|^2} \\
&= T_{k1} + T_{k2} + T_{k3} + T_{k4}, \\
T_{k1} &= \frac{\widehat{s}_k\widehat{s}_k^T}{\widehat{s}_k^T\widehat{z}_k} - \frac{\widehat{s}_k\widehat{s}_k^T}{\widehat{s}_k^T\widehat{s}_k}, \\
T_{k2} &= \frac{\widehat{z}_k^T\widehat{H_k}^\sharp\widehat{z}_k}{(\widehat{s}_k^T\widehat{z}_k)^2}\widehat{s}_k\widehat{s}_k^T - \frac{\widehat{s}_k^T\widehat{H_k}^\sharp\widehat{s}_k}{(\widehat{s}_k^T\widehat{s}_k)^2}\widehat{s}_k\widehat{s}_k^T, \\
T_{k3} &= \frac{\widehat{H_k}^\sharp\widehat{s}_k\widehat{s}_k^T}{\widehat{s}_k^T\widehat{s}_k} - \frac{\widehat{H_k}^\sharp\widehat{z}_k\widehat{s}_k^T}{\widehat{s}_k^T\widehat{z}_k}, \\
T_{k4} &= T_{k3}^T.
\end{aligned}$$

Lemma 5 implies that there exist positive constants t_1, t_2, and t_3 such that

$$\|T_{k1}\|_F \le \frac{\|\widehat{s}_k\|\|\widehat{z}_k - \widehat{s}_k\|}{\widehat{s}_k^T\widehat{z}_k} \le \frac{\|M^{-1}\|\|z_k - \nabla^2 f(x_*)s_k\|}{(1-\rho^\sharp)\|Ms_k\|}$$

$$\leq \left(\frac{\|M^{-1}\|^2\zeta}{1-\rho^\sharp}\right)\sigma_k^p \leq t_1\sigma_k^p,$$

$$\|T_{k2}\|_F = \frac{1}{(\widehat{s}_k^T\widehat{z}_k)^2\|\widehat{s}_k\|^2}\Big(\|\widehat{s}_k\|^4|\widehat{z}_k^T\widehat{H}_k^\sharp\widehat{z}_k - \widehat{z}_k^T\widehat{H}_k^\sharp\widehat{s}_k| + \|\widehat{s}_k\|^4|\widehat{z}_k^T\widehat{H}_k^\sharp\widehat{s}_k - \widehat{s}_k^T\widehat{H}_k^\sharp\widehat{s}_k|$$
$$+ \widehat{s}_k^T\widehat{H}_k^\sharp\widehat{s}_k|\widehat{s}_k^T\widehat{s}_k - \widehat{s}_k^T\widehat{z}_k|(\widehat{s}_k^T\widehat{s}_k + \widehat{s}_k^T\widehat{z}_k)\Big)$$
$$\leq \frac{2K^\sharp\|\widehat{s}_k\|^2\|M\|^2(\|\widehat{z}_k\| + \|\widehat{s}_k\|)\|\widehat{z}_k - \widehat{s}_k\|}{(\widehat{s}_k^T\widehat{z}_k)^2}$$
$$\leq 8\left(\frac{1}{1-\rho^\sharp}\right)^2 K^\sharp\zeta\|M\|^2\|M^{-1}\|(\|M^{-1}\|\zeta\sigma_k^p + 2\|M\|)\varepsilon^2\sigma_k^p \leq t_2\sigma_k^p,$$

$$\|T_{k3}\|_F = \|T_{k4}\|_F$$
$$\leq K^\sharp\|M\|\|z_k - \nabla^2 f(x_*)s_k\|\left(\frac{1}{\|\widehat{s}_k\|} + \frac{\|z_k\|\|M^{-1}\|}{\widehat{s}_k^T\widehat{z}_k}\right)$$
$$\leq K^\sharp\zeta\|M\|\|M^{-1}\|\left(1 + \frac{\|M^{-1}\|}{1-\rho^\sharp}(\zeta\sigma_k^p + \|M\|^2)\right)\sigma_k^p \leq t_3\sigma_k^p.$$

Then setting $W = t_1 + t_2 + 2t_3$, we have

$$\|T_k\|_F \leq \|T_{k1}\|_F + \|T_{k2}\|_F + \|T_{k3}\|_F + \|T_{k4}\|_F$$
$$\leq (t_1 + t_2 + 2t_3)\sigma_k^p = W\sigma_k^p.$$

Denoting

$$D_k = \left\|\left(I - \frac{\widehat{s}_k\widehat{s}_k^T}{\|\widehat{s}_k\|^2}\right)(\widehat{H}_k^\sharp - I)\left(I - \frac{\widehat{s}_k\widehat{s}_k^T}{\|\widehat{s}_k\|^2}\right)\right\|_F,$$

we have

$$\frac{\|(\widehat{H}_k^\sharp - I)\widehat{s}_k\|^2}{\|\widehat{s}_k\|^2} \leq 2\|H_k^\sharp - \nabla^2 f(x_*)^{-1}\|_{F,M}(\|H_k^\sharp - \nabla^2 f(x_*)^{-1}\|_{F,M} - D_k).$$

Since

$$\|H_k^\sharp - \nabla^2 f(x_*)^{-1}\|_{F,M} \leq \delta^\sharp,$$

the inequality

$$\frac{\|(\widehat{H}_k^\sharp - I)\widehat{s}_k\|^2}{\|\widehat{s}_k\|^2} \leq 2\delta^\sharp(\|H_k^\sharp - \nabla^2 f(x_*)^{-1}\|_{F,M} - D_k)$$

holds, and hence

$$D_k \leq \|H_k^\sharp - \nabla^2 f(x_*)^{-1}\|_{F,M} - \frac{1}{2\delta^\sharp}\frac{\|(\widehat{H}_k^\sharp - I)\widehat{s}_k\|^2}{\|\widehat{s}_k\|^2}.$$

Therefore we obtain the result

$$\|H_{k+1}^{\mathrm{BFGS}} - \nabla^2 f(x_*)^{-1}\|_{F,M} \leq \|H_k^\sharp - \nabla^2 f(x_*)^{-1}\|_{F,M} - \frac{\|\widehat{s}_k - \widehat{H}_k^\sharp\widehat{s}_k\|^2}{2\delta^\sharp\|\widehat{s}_k\|^2} + W\sigma_k^p. \qquad \square$$

LEMMA 7. *Assume that, for constants $\rho^\sharp$ and $\delta^\sharp$,*

$$\widehat{s}_k^T\widehat{z}_k \geq (1-\rho^\sharp)\|\widehat{s}_k\|^2 \qquad \text{and} \qquad \|H_k^\sharp - \nabla^2 f(x_*)^{-1}\|_{F,M} \leq \delta^\sharp$$

with $0 < \rho^\sharp < 1$ *and* $0 < \delta^\sharp < 1$. *Then*

$$\|\Delta H_k\|_{F,M} \leq \frac{4}{(1-\rho^\sharp)^2(1-\delta^\sharp)} \left(\frac{\|\widehat{s}_k - \widehat{H}_k^\sharp \widehat{s}_k\|}{\|\widehat{s}_k\|} + \|\widehat{H}_k^\sharp\| \frac{\|\widehat{z}_k - \widehat{s}_k\|}{\|\widehat{s}_k\|} \right)^2.$$

Proof. We use the same estimate as Lemma 3.4 in Stachurski [16]. Using the inequalities

$$\widehat{s}_k^T \widehat{z}_k \geq (1-\rho^\sharp)\|\widehat{s}_k\|^2$$

and

$$\begin{aligned}
\widehat{z}_k^T \widehat{H}_k^\sharp \widehat{z}_k &= z_k^T M^{-1} M^{-1} z_k + z_k^T M^{-1} M (H_k^\sharp - \nabla^2 f(x_*)^{-1}) M M^{-1} z_k \\
&\geq \|M^{-1} z_k\|^2 - |z_k^T M^{-1} M (H_k^\sharp - \nabla^2 f(x_*)^{-1}) M M^{-1} z_k| \\
&\geq \|M^{-1} z_k\|^2 - \|H_k^\sharp - \nabla^2 f(x_*)^{-1}\|_{F,M} \|M^{-1} z_k\|^2 \\
&\geq (1-\delta^\sharp)\|M^{-1} z_k\|^2,
\end{aligned}$$

we have

$$\frac{1}{\widehat{s}_k^T \widehat{z}_k} \leq \frac{1}{(1-\rho^\sharp)\|\widehat{s}_k\|^2}$$

and

$$\frac{1}{\widehat{z}_k^T \widehat{H}_k^\sharp \widehat{z}_k} \leq \frac{1}{(1-\delta^\sharp)\|\widehat{z}_k\|^2}.$$

Since

$$\begin{aligned}
&\|(\widehat{z}_k^T \widehat{H}_k^\sharp \widehat{z}_k)\widehat{s}_k - (\widehat{s}_k^T \widehat{z}_k)\widehat{H}_k^\sharp \widehat{z}_k\| \\
&= \|\widehat{z}_k^T \widehat{H}_k^\sharp (\widehat{z}_k - \widehat{s}_k)\widehat{s}_k + \widehat{z}_k^T (\widehat{H}_k^\sharp \widehat{s}_k - \widehat{s}_k)\widehat{s}_k + \widehat{s}_k^T \widehat{z}_k (\widehat{s}_k - \widehat{H}_k^\sharp \widehat{s}_k + \widehat{H}_k^\sharp (\widehat{s}_k - \widehat{z}_k))\| \\
&\leq \|\widehat{z}_k^T \widehat{H}_k^\sharp (\widehat{z}_k - \widehat{s}_k)\widehat{s}_k + \widehat{z}_k^T (\widehat{H}_k^\sharp \widehat{s}_k - \widehat{s}_k)\widehat{s}_k\| + (\widehat{s}_k^T \widehat{z}_k)\|\widehat{s}_k - \widehat{H}_k^\sharp \widehat{s}_k + \widehat{H}_k^\sharp (\widehat{s}_k - \widehat{z}_k)\| \\
&\leq (\|\widehat{s}_k - \widehat{H}_k^\sharp \widehat{s}_k\| + \|\widehat{H}_k^\sharp\| \|\widehat{z}_k - \widehat{s}_k\|)(\|\widehat{s}_k\| \|\widehat{z}_k\| + \widehat{s}_k^T \widehat{z}_k) \\
&\leq 2(\|\widehat{s}_k - \widehat{H}_k^\sharp \widehat{s}_k\| + \|\widehat{H}_k^\sharp\| \|\widehat{z}_k - \widehat{s}_k\|) \|\widehat{s}_k\| \|\widehat{z}_k\|,
\end{aligned}$$

we can write

$$\begin{aligned}
\|\Delta H_k\|_{F,M} &= \|M \Delta H_k M\|_F \\
&= \frac{\|(\widehat{z}_k^T \widehat{H}_k^\sharp \widehat{z}_k)\widehat{s}_k - (\widehat{s}_k^T \widehat{z}_k)\widehat{H}_k^\sharp \widehat{z}_k\|^2}{(\widehat{z}_k^T \widehat{H}_k^\sharp \widehat{z}_k)(\widehat{s}_k^T \widehat{z}_k)^2} \\
&\leq \frac{4\|\widehat{s}_k\|^2 \|\widehat{z}_k\|^2 (\|\widehat{s}_k - \widehat{H}_k^\sharp \widehat{s}_k\| + \|\widehat{H}_k^\sharp\| \|\widehat{z}_k - \widehat{s}_k\|)^2}{(1-\rho^\sharp)^2 \|\widehat{s}_k\|^4 (1-\delta^\sharp)\|\widehat{z}_k\|^2} \\
&= \frac{4}{(1-\rho^\sharp)^2(1-\delta^\sharp)} \left(\frac{\|\widehat{s}_k - \widehat{H}_k^\sharp \widehat{s}_k\|}{\|\widehat{s}_k\|} + \|\widehat{H}_k^\sharp\| \frac{\|\widehat{z}_k - \widehat{s}_k\|}{\|\widehat{s}_k\|} \right)^2. \qquad \square
\end{aligned}$$

We now prove local and linear convergence of our method.

THEOREM 1. *Suppose that the standard Assumptions* A1 *and* A2 *are satisfied. Let* ϕ' *be any constant such that* $0 \leq \phi_k \leq \phi'$. *Let the matrix* L_k *be updated by* (28). *Let the sequence* $\{x_k\}$ *be generated by*

$$x_{k+1} = x_k + s_k \quad \text{and} \quad (J(x_k) + L_k)^T (J(x_k) + L_k) s_k = -J(x_k)^T r(x_k). \tag{46}$$

Then, there exist positive constants ε *and* δ *such that if*

$$\|x_0 - x_*\| \leq \varepsilon, \qquad x_0 \in D$$

and

$$\|[(J(x_0) + L_0)^T (J(x_0) + L_0)]^{-1} - \nabla^2 f(x_*)^{-1}\|_{F,M} \leq \delta,$$

the sequence $\{x_k\}$ *is well defined and converges linearly to the local minimizer* x_*.

Proof. For given $\nu \in (0,1)$, choose δ and ε such that

$$\delta < \frac{1}{3}, \tag{47}$$

$$\varepsilon \leq 1, \tag{48}$$

$$\delta \leq \frac{\nu}{2\eta_2 \zeta_3}, \tag{49}$$

$$2\delta\eta_2 \|\nabla^2 f(x_*)\| + \zeta_1 \xi_1 \varepsilon^p \leq \nu, \tag{50}$$

$$\rho_1 \equiv 2^{p+1} \xi_2 \zeta_1 (2\zeta_2 + \sqrt{\zeta_3}) \varepsilon^p < 1, \tag{51}$$

$$\rho_2 \equiv \|M^{-1}\|^2 \zeta_4 \varepsilon^p < 1, \tag{52}$$

$$\frac{\eta_1}{1-\rho_1} \zeta_1^2 2^{p+1} \xi_2 (2\zeta_2 + \sqrt{\zeta_3}) \varepsilon^p \leq \delta, \tag{53}$$

$$\psi' \tau - \frac{1}{6\delta} < 0, \tag{54}$$

$$\frac{\mu \varepsilon^p}{1 - \nu^p} \leq \delta, \tag{55}$$

where

$$\begin{aligned}
\zeta_1 &= \eta_2 + \|\nabla^2 f(x_*)^{-1}\|, \\
\zeta_2 &= \xi_2 + \|J(x_*)\|, \\
\zeta_3 &= (1+\nu)\|\nabla^2 f(x_*)\|, \\
\zeta_4 &= \xi_1 + \frac{2^p (p+2)}{p+1} \xi_2 \zeta_2, \\
\psi' &= \max(1, \phi'),
\end{aligned}$$

and the positive constants τ and μ are defined below.

Set

$$N_1 = \{x \in R^n \mid \|x - x_*\| \leq \varepsilon\} \subset D, \tag{56}$$

$$N_2 = \{H \in R^{n \times n} \mid \|H - \nabla^2 f(x_*)^{-1}\|_{F,M} \leq 2\delta\}. \tag{57}$$

Now we prove, by mathematical induction, that the following expressions (E1;k)–(E9;k) hold for all $k \geq 0$:

$$H_k \in N_2, \tag{E1;k}$$

(E2; k) $$\|H_k\| \le \zeta_1, \quad \|B_k\| \le \zeta_3, \quad \|L_k\| \le \zeta_2 + \sqrt{\zeta_3},$$

(E3; k) $$\|x_{k+1} - x_*\| \le \nu \|x_k - x_*\|, \quad x_{k+1} \in N_1,$$

(E4; k) $$\|B_k^\sharp\| \le 2^{p+1}\xi_2(2\zeta_2 + \sqrt{\zeta_3}) + \zeta_3 \quad \text{and} \quad \|H_k^\sharp\| \le \frac{\zeta_1}{1-\rho_1},$$

(E5; k) $$\|z_k - \nabla^2 f(x_*)s_k\| \le \zeta_4 \sigma_k^p \|s_k\|,$$

(E6; k) $$(1-\rho_2)\|Ms_k\|^2 \le s_k^T z_k \le (1+\rho_2)\|Ms_k\|^2,$$

(E7; k) $$\|H_k^\sharp - H_k\|_{F,M} \le \frac{\eta_1}{1-\rho_1}\zeta_1^2 2^{p+1}\xi_2(2\zeta_2 + \sqrt{\zeta_3})\sigma_k^p < \delta,$$

(E8; k)
$$\|H_{k+1}^{\mathrm{BFGS}} - \nabla^2 f(x_*)^{-1}\|_{F,M} \le \|H_k^\sharp - \nabla^2 f(x_*)^{-1}\|_{F,M} + W\sigma_k^p - \frac{1}{6\delta}\frac{\|\widehat{s}_k - \widehat{H}_k^\sharp \widehat{s}_k\|^2}{\|\widehat{s}_k\|^2},$$

(E9; k) $$\|H_{k+1} - \nabla^2 f(x_*)^{-1}\|_{F,M} \le \|H_k - \nabla^2 f(x_*)^{-1}\|_{F,M} + \mu\sigma_k^p + \left(\psi'\tau - \frac{1}{6\delta}\right)\frac{\|\widehat{s}_k - \widehat{H}_k^\sharp \widehat{s}_k\|^2}{\|\widehat{s}_k\|^2},$$

where W is some positive constant,

$$\mu = W + \psi'\tau\rho + \frac{\eta_1}{1-\rho_1}\zeta_1^2 2^{p+1}\xi_2(2\zeta_2 + \sqrt{\zeta_3}),$$
$$\tau = \frac{4}{(1-\rho_2)^2(1-3\delta)},$$

and

$$\rho = \frac{\zeta_1}{1-\rho_1}\zeta_4\|M\|^2\|M^{-1}\|^2\left(6\eta_2\|M\|^2 + \frac{\zeta_1}{1-\rho_1}\|M\|^2\|M^{-1}\|^2\zeta_4\right).$$

We first consider the case of $k = 0$.
(E1;0) It is clear from the choice of the initial matrix.
(E2;0) Since

$$\begin{aligned}\|H_0\| &\le \|H_0 - \nabla^2 f(x_*)^{-1}\| + \|\nabla^2 f(x_*)^{-1}\| \\ &\le 2\delta\eta_2 \ + \ \|\nabla^2 f(x_*)^{-1}\| \\ &\le \zeta_1,\end{aligned}$$

the matrix H_0 is bounded. It follows from (49) and (E1;0) that

$$\|\nabla^2 f(x_*)\|\,\|H_0 - \nabla^2 f(x_*)^{-1}\| \le 2\eta_2\delta\|\nabla^2 f(x_*)\| \le \frac{\nu}{1+\nu} < 1.$$

By the Banach perturbation lemma,

$$\|B_0\| = \|H_0^{-1}\| \leq (1+\nu)\|\nabla^2 f(x_*)\| = \zeta_3,$$

so we have

$$\begin{aligned}\|L_0\| &\leq \|J(x_0) + L_0\| + \|J(x_0)\| \\ &= \|B_0\|^{\frac{1}{2}} + \|J(x_0)\| \\ &\leq \sqrt{\zeta_3} + \xi_2\|x_0 - x_*\|^p + \|J(x_*)\| \\ &\leq \zeta_2 + \sqrt{\zeta_3}.\end{aligned}$$

(E3;0) It follows easily from (32) and (50) that

$$\begin{aligned}\|x_1 - x_*\| &\leq \|H_0\|\|\nabla f(x_0) - \nabla f(x_*) - \nabla^2 f(x_*)(x_0 - x_*)\| \\ &\quad + \|\nabla^2 f(x_*)\|\|H_0 - \nabla^2 f(x_*)^{-1}\|\|x_0 - x_*\| \\ &\leq (\xi_1\zeta_1\varepsilon^p + 2\eta_2\delta\|\nabla^2 f(x_*)\|)\ \|x_0 - x_*\| \\ &\leq \nu\|x_0 - x_*\| \leq \varepsilon.\end{aligned}$$

(E4;0) Since $x_1 \in N_1$, $J(x_1)$ is available. Thus the matrix $B_0^\sharp$ is well defined and we have

$$\begin{aligned}\|B_0^\sharp - B_0\| &\leq \|(J(x_1) + L_0)^T(J(x_1) - J(x_0))\| + \|(J(x_1) - J(x_0))^T(J(x_0) + L_0)\| \\ &\leq (2\|L_0\| + \|J(x_0)\| + \|J(x_1)\|)\|J(x_1) - J(x_0)\| \\ &\leq 2^{p+1}\xi_2(\zeta_2 + \sqrt{\zeta_3} + \xi_2\varepsilon^p + \|J(x_*)\|)\sigma_0^p \\ &\leq 2^{p+1}\xi_2(2\zeta_2 + \sqrt{\zeta_3})\sigma_0^p.\end{aligned}$$

Then

$$\|B_0^\sharp\| \leq \|B_0^\sharp - B_0\| + \|B_0\| \leq 2^{p+1}\xi_2(2\zeta_2 + \sqrt{\zeta_3}) + \zeta_3.$$

Furthermore, it follows from (51) that

$$\begin{aligned}\|B_0^{-1}\|\|B_0^\sharp - B_0\| &\leq 2^{p+1}\xi_2\zeta_1(2\zeta_2 + \sqrt{\zeta_3})\sigma_0^p \\ &\leq 2^{p+1}\xi_2\zeta_1(2\zeta_2 + \sqrt{\zeta_3})\varepsilon^p \\ &= \rho_1 < 1.\end{aligned}$$

Thus by the Banach perturbation lemma, the matrix $B_0^\sharp$ is nonsingular. This implies that $B_0^\sharp$ is positive definite. Since

$$H_0^\sharp = (B_0^\sharp)^{-1},$$

we have

$$\|H_0^\sharp\| \leq \frac{\|B_0^{-1}\|}{1-\rho_1} \leq \frac{\zeta_1}{1-\rho_1}.$$

(E5;0) and (E6;0). These follow directly from Lemma 5.

From the positive definiteness of $B_0^\sharp$ and the positivity of $s_k^T z_k$, the matrix L_1 is well defined. Thus Lemma 3 implies that the matrices B_1 and H_1 are nonsingular and $H_1 = B_1^{-1}$. Then the next point x_2 can be defined by

$$\begin{aligned}x_2 &= x_1 - B_1^{-1}\nabla f(x_1) \\ &= x_1 - H_1\nabla f(x_1).\end{aligned}$$

(E7;0) It follows from (E4;0) and (53) that

$$\|H_0^\sharp - H_0\|_{F,M} \le \eta_1 \|H_0^\sharp\| \, \|H_0\| \, \|B_0^\sharp - B_0\| \le \frac{\eta_1}{1-\rho_1} \zeta_1^2 2^{p+1} \xi_2 (2\zeta_2 + \sqrt{\zeta_3}) \sigma_0^p \le \delta.$$

(E8;0) Recall that

$$\|H_0^\sharp - \nabla^2 f(x_*)^{-1}\|_{F,M} \le \|H_0^\sharp - H_0\|_{F,M} + \|H_0 - \nabla^2 f(x_*)^{-1}\|_{F,M} \le 3\delta. \tag{58}$$

Therefore, setting

$$\delta^\sharp = 3\delta, \qquad \zeta = \zeta_4, \qquad \rho^\sharp = \rho_2 \qquad \text{and} \qquad K^\sharp = \frac{\zeta_1}{1-\rho_1}$$

in Lemma 6, we obtain

$$\|H_1^{\mathrm{BFGS}} - \nabla^2 f(x_*)^{-1}\|_{F,M} \le \|H_0^\sharp - \nabla^2 f(x_*)^{-1}\|_{F,M} - \frac{\|\widehat{s}_0 - \widehat{H}_0^\sharp \widehat{s}_0\|^2}{6\delta \|\widehat{s}_0\|^2} + W\sigma_0^p,$$

where W is some positive constant.
(E9;0) The update formula (35) for H_1 yields

$$\|H_1 - \nabla^2 f(x_*)^{-1}\|_{F,M} \le \|H_1^{\mathrm{BFGS}} - \nabla^2 f(x_*)^{-1}\|_{F,M} + |1 - \psi_0| \, \|\Delta H_0\|_{F,M}.$$

Using (E6;0) and (58), and setting $\rho^\sharp = \rho_2$ and $\delta^\sharp = 3\delta$ in Lemma 7, we have

$$\|\Delta H_0\|_{F,M} \le \frac{4}{(1-\rho_2)^2(1-3\delta)} \left(\frac{\|\widehat{s}_0 - \widehat{H}_0^\sharp \widehat{s}_0\|}{\|\widehat{s}_0\|} + \|\widehat{H}_0^\sharp\| \frac{\|\widehat{z}_0 - \widehat{s}_0\|}{\|\widehat{s}_0\|} \right)^2.$$

Recalling that

$$\|\widehat{z}_0 - \widehat{s}_0\| \le \|M^{-1}\| \|z_0 - \nabla^2 f(x_*) s_0\| \le \|M^{-1}\|^2 \|\widehat{s}_0\| \zeta_4 \sigma_0^p,$$

$$\begin{aligned} \|\widehat{s}_0 - \widehat{H}_0^\sharp \widehat{s}_0\| &\le \|M(H_0^\sharp - \nabla^2 f(x_*)^{-1}) M\| \|\widehat{s}_0\| \\ &\le 3\eta_2 \|M\|^2 \|\widehat{s}_0\| \delta, \end{aligned}$$

and

$$\|\widehat{H}_0^\sharp\| \le \|H_0^\sharp\| \|M\|^2 \le \frac{\zeta_1}{1-\rho_1} \|M\|^2,$$

we have

$$\begin{aligned} \|\Delta H_0\|_{F,M} &\le \frac{4}{(1-\rho_2)^2(1-3\delta)} \left[\frac{\|\widehat{s}_0 - \widehat{H}_0^\sharp \widehat{s}_0\|^2}{\|\widehat{s}_0\|^2} \right. \\ &\qquad \left. + \left(2 \frac{\|\widehat{s}_0 - \widehat{H}_0^\sharp \widehat{s}_0\|}{\|\widehat{s}_0\|} + \frac{\|\widehat{H}_0^\sharp\| \|\widehat{z}_0 - \widehat{s}_0\|}{\|\widehat{s}_0\|} \right) \frac{\|\widehat{H}_0^\sharp\| \|\widehat{z}_0 - \widehat{s}_0\|}{\|\widehat{s}_0\|} \right] \\ &\le \tau \left(\frac{\|\widehat{s}_0 - \widehat{H}_0^\sharp \widehat{s}_0\|^2}{\|\widehat{s}_0\|^2} + \rho \sigma_0^p \right). \end{aligned}$$

By Lemma 4, we have

$$\|H_1 - \nabla^2 f(x_*)^{-1}\|_{F,M} \leq \|H_1^{\mathrm{BFGS}} - \nabla^2 f(x_*)^{-1}\|_{F,M} + \psi'\tau\left(\frac{\|\widehat{s}_0 - \widehat{H}_0^\sharp \widehat{s}_0\|^2}{\|\widehat{s}_0\|^2} + \rho\sigma_0^p\right).$$

Finally, (E7;0), (E8;0), and the inequality

$$\|H_0^\sharp - \nabla^2 f(x_*)^{-1}\|_{F,M} \leq \|H_0^\sharp - H_0\|_{F,M} + \|H_0 - \nabla^2 f(x_*)^{-1}\|_{F,M}$$

imply that

$$\begin{aligned}
\|H_1 - \nabla^2 f(x_*)^{-1}\|_{F,M} &\leq \|H_0^\sharp - \nabla^2 f(x_*)^{-1}\|_{F,M} - \frac{1}{6\delta}\frac{\|\widehat{s}_0 - \widehat{H}_0^\sharp \widehat{s}_0\|^2}{\|\widehat{s}_0\|^2} \\
&\quad + W\sigma_0^p + \psi'\tau\left(\frac{\|\widehat{s}_0 - \widehat{H}_0^\sharp \widehat{s}_0\|^2}{\|\widehat{s}_0\|^2} + \rho\sigma_0^p\right) \\
&= \|H_0^\sharp - \nabla^2 f(x_*)^{-1}\|_{F,M} + \left(\psi'\tau - \frac{1}{6\delta}\right)\frac{\|\widehat{s}_0 - \widehat{H}_0^\sharp \widehat{s}_0\|^2}{\|\widehat{s}_0\|^2} \\
&\quad + (W + \psi'\tau\rho)\sigma_0^p \\
&\leq \|H_0 - \nabla^2 f(x_*)^{-1}\|_{F,M} + \left(\psi'\tau - \frac{1}{6\delta}\right)\frac{\|\widehat{s}_0 - \widehat{H}_0^\sharp \widehat{s}_0\|^2}{\|\widehat{s}_0\|^2} + \mu\sigma_0^p.
\end{aligned}$$

Therefore the case of $k = 0$ is proven.

We assume as an induction hypotheses that the expressions (E1;k)–(E9;k) hold for $k = 0, \ldots, t-1$. Then we have

$$\|H_{k+1} - \nabla^2 f(x_*)^{-1}\|_{F,M} \leq \|H_k - \nabla^2 f(x_*)^{-1}\|_{F,M} + \mu\sigma_k^p + \left(\psi'\tau - \frac{1}{6\delta}\right)\frac{\|\widehat{s}_k - \widehat{H}_k^\sharp \widehat{s}_k\|^2}{\|\widehat{s}_k\|^2}$$

for $k = 0, \ldots, t-1$, and by summing both sides from $k = 0$ to $t-1$, it follows that

$$\begin{aligned}
\|H_t - \nabla^2 f(x_*)^{-1}\|_{F,M} &\leq \|H_0 - \nabla^2 f(x_*)^{-1}\|_{F,M} \\
&\quad + \mu\sum_{k=0}^{t-1}\sigma_k^p + \left(\psi'\tau - \frac{1}{6\delta}\right)\sum_{k=0}^{t-1}\frac{\|\widehat{s}_k - \widehat{H}_k^\sharp \widehat{s}_k\|^2}{\|\widehat{s}_k\|^2}.
\end{aligned}$$

Noting that

$$\sigma_k = \max(\|x_{k+1} - x_*\|, \|x_k - x_*\|) = \|x_k - x_*\| \leq \nu^k\varepsilon,$$

and using the conditions (54) and (55), we obtain

$$\begin{aligned}
\|H_t - \nabla^2 f(x_*)^{-1}\|_{F,M} &\leq \|H_0 - \nabla^2 f(x_*)^{-1}\|_{F,M} + \mu\varepsilon^p\sum_{k=0}^{t-1}(\nu^p)^k \\
&\quad + \left(\psi'\tau - \frac{1}{6\delta}\right)\sum_{k=0}^{t-1}\frac{\|\widehat{s}_k - \widehat{H}_k^\sharp \widehat{s}_k\|^2}{\|\widehat{s}_k\|^2} \\
&\leq \|H_0 - \nabla^2 f(x_*)^{-1}\|_{F,M} + \frac{\mu}{1-\nu^p}\varepsilon^p \\
&\quad + \left(\psi'\tau - \frac{1}{6\delta}\right)\sum_{k=0}^{t-1}\frac{\|\widehat{s}_k - \widehat{H}_k^\sharp \widehat{s}_k\|^2}{\|\widehat{s}_k\|^2} \\
&\leq \|H_0 - \nabla^2 f(x_*)^{-1}\|_{F,M} + \frac{\mu}{1-\nu^p}\varepsilon^p \\
&\leq 2\delta,
\end{aligned}$$

which implies $H_t \in N_2$. We can prove (E2;k)–(E9;k) for $k = t$ in the same way as the case of $k = 0$.

This concludes the induction and the proof. □

The following theorem shows q-superlinear convergence of our method.

THEOREM 2. *Suppose that all conditions of Theorem* 1 *hold. Then the sequence* $\{x_k\}$ *generated by the scheme* (46) *with the factorized Broyden-like family* (28) *converges q-superlinearly to* x_*.

Proof. It follows directly from the proof of Theorem 1 that, for all $t \geq 0$,

$$\|H_t - \nabla^2 f(x_*)^{-1}\|_{F,M} \leq \|H_0 - \nabla^2 f(x_*)^{-1}\|_{F,M} + \frac{\mu}{1-\nu^p}\varepsilon^p + \left(\psi'\tau - \frac{1}{6\delta}\right)\sum_{k=0}^{t-1}\frac{\|\widehat{s}_k - \widehat{H}_k^\sharp \widehat{s}_k\|^2}{\|\widehat{s}_k\|^2}.$$

The preceding yields

$$\begin{aligned}\left(\frac{1}{6\delta} - \psi'\tau\right)\sum_{k=0}^{t-1}\frac{\|\widehat{s}_k - \widehat{H}_k^\sharp \widehat{s}_k\|^2}{\|\widehat{s}_k\|^2} &\leq \|H_0 - \nabla^2 f(x_*)^{-1}\|_{F,M} + \frac{\mu}{1-\nu^p}\varepsilon^p \\ &\quad - \|H_t - \nabla^2 f(x_*)^{-1}\|_{F,M} \\ &\leq \|H_0 - \nabla^2 f(x_*)^{-1}\|_{F,M} + \frac{\mu}{1-\nu^p}\varepsilon^p \\ &\leq 2\delta\end{aligned}$$

and hence

$$\sum_{k=0}^{t-1}\frac{\|\widehat{s}_k - \widehat{H}_k^\sharp \widehat{s}_k\|^2}{\|\widehat{s}_k\|^2} \leq \frac{2\delta}{\dfrac{1}{6\delta} - \psi'\tau},$$

which guarantees the convergence of the infinite series

$$\sum_{k=0}^{\infty}\frac{\|\widehat{s}_k - \widehat{H}_k^\sharp \widehat{s}_k\|^2}{\|\widehat{s}_k\|^2}.$$

Thus

$$\lim_{k\to\infty}\frac{\|\widehat{s}_k - \widehat{H}_k^\sharp \widehat{s}_k\|}{\|\widehat{s}_k\|} = 0. \tag{59}$$

Letting $\widehat{B}_k = M^{-1}B_k M^{-1}$ gives

$$\begin{aligned}\frac{\|\widehat{s}_k - \widehat{B}_k \widehat{s}_k\|}{\|\widehat{s}_k\|} &\leq \|M^{-1}\|^2\|B_k\|\frac{\|\widehat{s}_k - \widehat{H}_k \widehat{s}_k\|}{\|\widehat{s}_k\|} \\ &\leq \|M^{-1}\|^2\zeta_3\left(\frac{\|\widehat{s}_k - \widehat{H}_k^\sharp \widehat{s}_k\|}{\|\widehat{s}_k\|} + \|M\|^2\|H_k^\sharp - H_k\|\right).\end{aligned}$$

It follows from (E7;k) that

$$\lim_{k\to\infty}\|H_k^\sharp - H_k\| = 0,$$

so (59) yields

$$\lim_{k\to\infty} \frac{\|\widehat{s}_k - \widehat{B}_k\widehat{s}_k\|}{\|\widehat{s}_k\|} = 0.$$

Therefore we obtain

$$\frac{\|(B_k - \nabla^2 f(x_*))s_k\|}{\|s_k\|} = \frac{\|M(M^{-1}B_kM^{-1} - I)Ms_k\|}{\|Ms_k\|}\frac{\|Ms_k\|}{\|s_k\|} \le \|M\|^2 \frac{\|\widehat{B}_k\widehat{s}_k - \widehat{s}_k\|}{\|\widehat{s}_k\|},$$

which implies

$$\lim_{k\to\infty} \frac{\|(B_k - \nabla^2 f(x_*))s_k\|}{\|s_k\|} = 0.$$

This is the necessary and sufficient condition that the sequence $\{x_k\}$ converges q-superlinearly to x_* [6]. □

4. Concluding remarks. This paper has been concerned with structured quasi-Newton methods for nonlinear least squares problems. Among line search descent methods, the factorized versions of structured quasi-Newton methods were proposed by Yabe and Takahashi [19], [20]. They gave the BFGS-like and the DFP-like updates and proved local and q-superlinear convergence of these methods. In this paper, we have extended their updates. We have found a general solution to the matrix equations, and by using the structure principle given in [11], we have obtained a factorized Broyden-like family. Furthermore, we have shown local and q-superlinear convergence of our method by using a way similar to Stachurski [16]. The significant difference between our proof and that of Stachurski is that we dealt with estimates on the matrix L_k and the intermediate matrices $B_k^\sharp$ and $H_k^\sharp$.

For structured quasi-Newton methods, Engels and Martinez [12] and Martinez [15] proposed the convex class of structured secant update, i.e., $0 \le \phi_k \le 1$ in a Broyden-like family, based on (6) and showed local and q-superlinear convergence of their method. On the other hand, the factorized Broyden-like family proposed in this paper allows $\phi_k > 1$ in addition to $0 \le \phi_k \le 1$, and maintains the positive definiteness of the matrix B_k. It is very interesting to investigate the relationship between our factorized Broyden-like family for L_k and the Engels–Martinez family for A_k. This relationship has been slightly, but not completely discussed in [18]. Further research for this relationship is needed.

One of the main purposes of this paper is to obtain a descent search direction within the framework of the line search strategy. The discussions in §§1 and 2 indicate that the condition

$$s_k^T z_k > 0 \tag{60}$$

plays an important role in maintaining the positive definiteness of the matrix $(J(x_k)+L_k)^T(J(x_k)+L_k)$. Dennis, Martinez, and Tapia [11] proved that, for their structured BFGS update, this condition was locally satisfied and considered the neighborhood of x_* such as the intermediate matrix $J(x_{k+1})^TJ(x_{k+1}) + A_k$ was positive definite. On the other hand, we have shown that the condition (60) is also locally satisfied for our family and have considered the neighborhood of x_* such as the intermediate matrix $J(x_{k+1}) + L_k$ was of column full rank. Both of the two locally guarantee the positive definiteness of B_{k+1}.

Here we discuss a line search criterion such that the condition (60) holds. Recall that $z_k = \nabla f(x_{k+1}) - \nabla f(x_k)$ for standard quasi-Newton methods. In this case, the condition (60) always holds for $f(x)$ strictly convex, and can be satisfied by using a suitable line search criterion like the Wolfe condition for a general objective function. It is an open problem to find a line search criterion such that the condition (60) is satisfied within the framework of structured quasi-Newton methods. However, if we impose convexity on the objective function, we have the following theorem.

THEOREM 3. *Assume that the Hessian matrix $\nabla^2 f(x)$ is positive definite in R^n and that $d_k \neq 0$. Then there exists a positive constant α_k^* such that*

$$(\alpha d_k)^T \left(J(x_k + \alpha d_k)^T J(x_k + \alpha d_k)(\alpha d_k) + (J(x_k + \alpha d_k) - J(x_k))^T r(x_k + \alpha d_k) \right) > 0$$

for all α, $0 < \alpha < \alpha_k^*$.

Proof. Define

$$\theta(\alpha) = d_k^T \left(J(x_k + \alpha d_k)^T J(x_k + \alpha d_k)(\alpha d_k) + (J(x_k + \alpha d_k) - J(x_k))^T r(x_k + \alpha d_k) \right).$$

Since

$$\begin{aligned} \frac{d\theta}{d\alpha} = \sum_{i=1}^{m} \big(& ((\nabla r_i(x_k + \alpha d_k)^T d_k)^2 + 2\alpha (\nabla r_i(x_k + \alpha d_k)^T d_k) d_k^T \nabla^2 r_i(x_k + \alpha d_k) d_k \\ & + d_k^T \nabla r_i(x_k + \alpha d_k)(\nabla r_i(x_k + \alpha d_k) - \nabla r_i(x_k))^T d_k \\ & + r_i(x_k + \alpha d_k) d_k^T \nabla^2 r_i(x_k + \alpha d_k) d_k \big), \end{aligned}$$

we have

$$\left. \frac{d\theta}{d\alpha} \right|_{\alpha=0} = d_k^T \nabla^2 f(x_k) d_k > 0 \qquad \text{and} \qquad \theta(0) = 0.$$

Thus by the continuity of $\theta(\alpha)$, there exists a positive constant α_k^* such that

$$\theta(\alpha) > \theta(0) = 0$$

for all $\alpha, 0 < \alpha < \alpha_k^*$. Therefore the proof is complete. □

Since $s_k = \alpha_k d_k$, the preceding theorem enables us to obtain the next point x_{k+1} which satisfies the condition (60). Thus, as line search criteria, we may combine the condition (60) and the Armijo condition

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \mu \alpha_k \nabla f(x_k)^T d_k,$$

where $0 < \mu < 1/2$. This must be very useful in showing global convergence of our method when the Hessian matrix $\nabla^2 f(x)$ is positive definite in R^n.

In practice, we know that, for zero residual problems ($f(x_*) = 0$), the matrix L_k should ideally converge to zero. If the matrix does not at least become small in those cases, then our method cannot hope to compete with the Gauss–Newton method. Since the quasi-Newton updates do not generate the zero matrix, some remedies must be applied. As possible remedies, the hybrid method was proposed by Al-Baali and Fletcher [1], and the sizing technique was introduced by Bartholomew-Biggs [2] and Dennis, Gay, and Welsch [10]. It is easy to combine both of these methods with our factorized methods. For example, define β_k to be a sizing factor and set

$$N = J(x_{k+1}) + L_{k+1}, \quad a = s_k, \quad b = z_k, \quad \text{and} \quad \Phi = J(x_{k+1}) + \beta_k L_k$$

in (13). Then using (25), we obtain a factorized Broyden-like family with sizing:

$$L_{k+1} = \beta_k L_k + (1 - \sqrt{\phi_k}) \left(\frac{L_k^\sharp s_k}{s_k^T B_k^\sharp s_k} \right) (\sqrt{\lambda_k} z_k - B_k^\sharp s_k)^T + \sqrt{\phi_k}\, L_k^\sharp (\sqrt{\lambda_k} (B_k^\sharp)^{-1} z_k - s_k) \left(\frac{z_k}{s_k^T z_k} \right)^T, \tag{61}$$

where

$$L_k^\sharp = J(x_{k+1}) + \beta_k L_k, \qquad B_k^\sharp = (L_k^\sharp)^T L_k^\sharp,$$

$$\phi_k \geq 0 \quad \text{and} \quad \lambda_k = \frac{1}{(1 - \phi_k) \dfrac{s_k^T z_k}{s_k^T B_k^\sharp s_k} + \phi_k \dfrac{z_k^T (B_k^\sharp)^{-1} z_k}{s_k^T z_k}}.$$

We can consider a sizing factor similar to the factors of Bartholomew-Biggs and Dennis et al. Note that the factorized BFGS-like and the factorized DFP-like updates with sizing were first proposed by Yabe and Takahashi [19], [20]. Thus the preceding result (61) is an extension of the sized updates of Yabe and Takahashi to a factorized Broyden-like family.

Acknowledgments. The authors are grateful to the referees for the valuable comments. The authors also would like to thank Dr. Robert Michael Lewis, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, for his kind comments to the first draft of this paper.

REFERENCES

[1] M. Al-Baali and R. Fletcher, *Variational methods for non-linear least squares*, J. Oper. Res. Soc., 36 (1985), pp. 405–421.

[2] M. C. Bartholomew-Biggs, *The estimation of the Hessian matrix in nonlinear least squares problems with non-zero residuals*, Math. Programming, 12 (1977), pp. 67–80.

[3] C. G. Broyden, J. E. Dennis, Jr., and J. J. Moré, *On the local and superlinear convergence of quasi-Newton methods*, J. Inst. Math. Appl., 12 (1973), pp. 223–245.

[4] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses—Theory and Applications*, Robert E. Krieger Publishing Company, Malabar, FL, 1980.

[5] J. E. Dennis, Jr., *A brief survey of convergence results for quasi-Newton methods*, SIAM–AMS Proc., 9 (1976), pp. 185–199.

[6] J. E. Dennis, Jr. and J. J. Moré, *A characterization of superlinear convergence and its application to quasi-Newton methods*, Math. Comp., 28 (1974), pp. 549–560.

[7] J.E. Dennis, Jr. and R.B. Schnabel, *A new derivation of symmetric positive definite secant updates*, in Nonlinear Programming 4, O.L. Mangasarian, R.R. Meyer, and S.M. Robinson, eds., Academic Press, New York, 1981, pp. 167–199.

[8] ———, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, NJ, 1983.

[9] J.E. Dennis, Jr. and H.F. Walker, *Convergence theorems for least-change secant update methods*, SIAM J. Numer. Anal., 18 (1981), pp. 949–987.

[10] J.E. Dennis, Jr., D.M. Gay, and R.E. Welsch, *An adaptive nonlinear least-squares algorithm*, ACM Trans. Math. Software, 7 (1981), pp. 348–368.

[11] J.E. Dennis, Jr., H.J. Martinez, and R.A. Tapia, *Convergence theory for the structured* BFGS *secant method with an application to nonlinear least squares*, J. Optim. Theory Appl., 61 (1989), pp. 161–178.

[12] J.R. Engels and H.J. Martinez, *Local and superlinear convergence for partially known quasi-Newton methods*, SIAM J. Optim., 1 (1991), pp. 42–56.

[13] R. FLETCHER AND C. XU, *Hybrid methods for nonlinear least squares*, IMA J. Numer. Anal., 7 (1987), pp. 371–389.

[14] J. HUSCHENS, *On the use of product structure in secant methods for nonlinear least squares problems*, SIAM J. Optim., 4 (1994), pp. 108–129.

[15] H.J. MARTINEZ, *Local and superlinear convergence of structured secant methods from the convex class*, Ph.D. Thesis, Department of Mathematical Sciences, Rice University, Houston, TX, 1988.

[16] A. STACHURSKI, *Superlinear convergence of Broyden's bounded θ-class of methods*, Math. Programming, 20 (1981), pp. 196–212.

[17] H. YABE, *A family of variable-metric methods with factorized expressions*, TRU Math., 17 (1981), pp. 141–152.

[18] ———, *Variations of structured Broyden families for nonlinear least squares problems*, Optim. Methods Software, 2 (1993), pp. 107–144.

[19] H. YABE AND T. TAKAHASHI, *Structured quasi–Newton methods for nonlinear least squares problems*, TRU Math., 24 (1988), pp. 195–209.

[20] ———, *Factorized quasi-Newton methods for nonlinear least squares problems*, Math. Programming, 51 (1991), pp. 75–100.

[21] N. YAMAKI AND H. YABE, *On factorized variable-metric methods*, TRU Math., 17 (1981), pp. 285–294.

SIAM J. OPTIMIZATION
Vol. 5, No. 4, pp. 792–812, November 1995

004

SEQUENTIAL QUADRATIC PROGRAMMING WITH PENALIZATION OF THE DISPLACEMENT*

J. F. BONNANS† AND G. LAUNAY†

Abstract. In this paper we study the convergence of a sequential quadratic programming algorithm for the nonlinear programming problem. The Hessian of the quadratic program is the sum of an approximation of the Lagrangian and of a multiple of the identity that allows us to penalize the displacement. Assuming only that the direction is a stationary point of the current quadratic program we study the local convergence properties without strict complementarity. In particular, we use a very weak condition on the approximation of the Hessian to the Lagrangian. We obtain some global and superlinearly convergent algorithm under weak hypotheses. As a particular case we formulate an extension of Newton's method that is quadratically convergent to a point satisfying a strong sufficient second-order condition.

Key words. nonlinear programming, Newton's method, quasi-Newton algorithms, exact penalization, trust region

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

1. Introduction.

1.1. The family of Newton-type algorithms. In this paper we present a new algorithm for solving the standard nonlinear programming problem

$$(P) \qquad \min f(x)\ ;\ g(x) \ll 0,$$

with f, g smooth mapping from $\mathbb{R}^n$ to $\mathbb{R}$ and $\mathbb{R}^p$, and, given a partition (I, J) of $\{1, \ldots, p\}$, by $z \ll 0$ we mean $z_i \leq 0$, $i \in I$, $z_j = 0$, $j \in J$. Occasionally for $K \subset I$ we will denote

$$z \overset{K}{\ll} 0 \Leftrightarrow \begin{cases} z_i \leq 0, & i \in K, \\ z_j = 0, & j \in J. \end{cases}$$

With (P) is associated the first-order optimality system

$$(1) \qquad \begin{aligned} &\nabla f(x) + g'(x)^t \lambda = 0, \\ &g(x) \ll 0,\ \lambda_I \geq 0,\ \lambda^t g(x) = 0. \end{aligned}$$

If (x, λ) satisfies (1), then we say that λ is a multiplier associated to x. By extension we say that x is solution of (1) if there exists λ such that (x, λ) satisfies (1).

We define the quadratic problem

$$Q(x, M) \qquad \min_d \nabla f(x)^t d + \frac{1}{2} d^t M d\ ;\ g(x) + g'(x)d \ll 0,$$

with which is associated the optimality system

$$(2) \qquad \begin{aligned} &\nabla f(x) + Md + g'(x)^t \mu = 0, \\ &g(x) + g'(x)d \ll 0,\ \mu_I \geq 0,\ \mu^t(g(x) + g'(x)d) = 0. \end{aligned}$$

* Received by the editors November 3, 1992; accepted for publication(in revised form) March 13, 1994.

† Institut National de Recherche en Informatique et en Automatique, Domaine de Voluceau, 78153 Rocquencourt, France (Frederic.Bonnans@inria.fr).

Denote by $L(x,\lambda) := f(x) + \lambda^t g(x)$ the Lagrangian associated with (P). It has been observed by Wilson [26] that, when no inequality is present, the computation of the Newton step in (1) amounts to solving $Q(x,M)$ with $M = \nabla_x^2 L(x,\lambda)$, and this allows a natural generalization for problems with inequality constraints. In order to deal with the case when second derivatives are not available, a larger class of interest is the following.

ALGORITHM 0 (Newton-type algorithms).
0. Choose $x^0 \in \mathbb{R}^n$, M^0 an $n \times n$ symmetric matrix ; $k \leftarrow 0$.
1. Compute (d^k, μ^k) solution of the optimality system of $Q(x^k, M^k)$.
2. Linesearch: choose ρ_k in $[0,1]$.
3. $x^{k+1} \leftarrow x^k + \rho_k d^k$
   Choose M^{k+1}.
   $k \leftarrow k+1$, go to 1.

1.2. Local study. Let $\bar{x}$ be a local solution of (P) with which is associated a unique Lagrange multiplier $\bar{\lambda}$. The local analysis typically assumes that (x^0, M^0) is close to $(\bar{x}, \nabla^2 L(\bar{x}, \bar{\lambda}))$ and that $\rho_k = 1$. The question is to determine if convergence occurs, and at which rate. It happens that in this case d^k should not, in general, be taken as the global minimum of $Q(x^k, M^k)$.

Indeed, let us consider the simple example

$$\min_x \ell n(1+x) \; ; \; -x \leq 0, \; x \leq 10.$$

This problem has a unique solution $\bar{x} = 0$ associated to the unique multiplier $\bar{\lambda} = (1,0)^t$ and the strongest regularity hypothesis and sufficient second-order condition (see (8) and (29) below) are satisfied by $(\bar{x}, \bar{\lambda})$. Now let us start Newton's method at the solution. We get the quadratic problem

$$\min_d d - d^2/2 \; ; \; 0 \leq d \leq 10,$$

whose unique solution is $d = 10$, the worst possible displacement! As the Newton step is obtained by linearizing the data, is it clear that the quadratic program is meaningful only if the displacement is not too large. Indeed, in our example, the "good" displacement $d = 0$ is a local solution of the quadratic program.

Of course if $M^k \geq 0$, which is the case for some quasi-Newton algorithms based on positive definite updates, and also for Newton's method when (P) is convex, i.e., has convex cost and inequality constraints and linear equality constraints, then $Q(x^k, M^k)$ is itself convex, and local and global minima coincide. We now quote some recent results about the speed of convergence of Newton-type algorithms. For this purpose, we need to define the set of active inequality constraints:

$$I(x) := \{i \in I \; ; \; g_i(x) = 0\},$$

the set of active constraints

$$I(x) \cup J,$$

the extended critical cone

$$C(x) := \{d \in \mathbb{R}^n \; ; \; g'(x)d \overset{I(\bar{x})}{\ll} 0 \; ; \; g_i'(x)d = 0 \text{ if } \bar{\lambda}_i > 0, \; i \in I\}. \tag{3}$$

Note that when $x = \bar{x}$ we recover the usual critical cone, or cone of critical directions:

$$C(\bar{x}) := \{d \in \mathbb{R}^n \ ; \ g'(\bar{x})d \overset{I(\bar{x})}{\ll} 0 \ ; \ g_i'(\bar{x})d = 0 \text{ if } \bar{\lambda}_i > 0, \ i \in I\}. \tag{4}$$

We also define the (standard) second-order sufficient condition

$$d^t \nabla_x^2 L(\bar{x}, \bar{\lambda}) d > 0 \quad \text{for all } d \in C(\bar{x}), d \neq 0, \tag{5}$$

and the orthogonal projection onto $C(x^k)$, denoted by P^k.

Note that usually the critical cone is defined as

$$C(\bar{x}) := \{d \in \mathbb{R}^n \ ; \ \nabla f(\bar{x})^t d \leq 0 \ ; \ g'(\bar{x})d \overset{I(\bar{x})}{\ll} 0\}.$$

Both definitions coincide (as is easy to check using (2)) because we assume the existence of a Lagrange multiplier. We now quote two results of Bonnans [8].

THEOREM 1.1. *Let $\bar{x}$ be a local solution of (1) such that the gradients of active constraints are linearly independent, $\bar{\lambda}$ be the unique multiplier associated with $\bar{x}$, and the second-order sufficient condition holds. Then if (x^k, μ^k) computed by Algorithm 0 converge to $(\bar{x}, \bar{\lambda})$, then $\{x^k\}$ converges superlinearly if and only if (iff)*

$$P^k[(\nabla_x^2 L(\bar{x}, \bar{\lambda}) - M^k)d^k] = o(d^k).$$

THEOREM 1.2. *Assume that $\bar{x}$ is a local solution of (1), $\bar{\lambda}$ is the unique Lagrange multiplier associated to $\bar{x}$, and the second-order sufficiency condition holds. Then there exists $\varepsilon > 0$ such that if $\|x^0 - \bar{x}\| + \|\lambda^0 - \bar{\lambda}\| < \varepsilon$, and (x^{k+1}, λ^{k+1}) is chosen so that $\|x^{k+1} - x^k\| + \|\lambda^{k+1} - \lambda^k\| < 2\varepsilon$, then Algorithm 0 with $M^k = \nabla_x^2 L(x^k, \lambda^k)$ and $\rho_k = 1$, i.e., Newton's method, is well defined and converges at a quadratic rate to $(\bar{x}, \bar{\lambda})$.*

We note that the existence of a unique multiplier is a qualification hypothesis slightly weaker than the linear independence of gradients of active constraints (see Fletcher [14]). Note also that if the following strict complementarity hypothesis holds:

$$\bar{\lambda}_i > 0 \quad \text{for all } i \text{ in } I(\bar{x}),$$

then, for (x^k, M^k) close to $(\bar{x}, \nabla^2 L(\bar{x}, \bar{\lambda}))$, λ^k is close to $\bar{\lambda}$; hence if $i \in I(\bar{x})$, the corresponding inequality in $Q(x^k, M^k)$ is active and everything goes as if we were analyzing the problem

$$\min f(x) \ ; \ g_i(x) = 0, \ i \in I(\bar{x}) \cup J.$$

Then Theorem 1.1 reduces to a result of Boggs, Tolle, and Wang [6], whereas Theorem 1.2 reduces to the application of the general result on quadratic convergence of Newton's method for a system of equations. The novelty in the theorems above lies in the fact that no strict complementarity hypothesis holds and only the standard (weak) sufficient condition is assumed.

1.3. Globalization. The local results that we just presented insure a superlinear or quadratic convergence, provided that the data at the starting point are sufficiently close to the optimum. When these hypotheses are not satisfied, the algorithm must be modified, for different reasons.

(i) It may happen that the optimality system of $Q(x^k, M^k)$ has no feasible solution; a possible remedy is to solve a modified quadratic program. This has been discussed by Fletcher [14]. We will not address this point.

(ii) The point $x^k + d^k$ may be farther from any local solution than x^k. For this reason it is safe to introduce a linesearch on some potential function; the most popular potential is the so-called exact penalty function (see Eremin [13], Zangwill [28], Han [16], Pschenichny and Danilin [22])

$$\theta_r(x) := f(x) + r\|g(x)^\sharp\|$$

with $r > 0$ (the penalty parameter) and $.^\sharp$ defined as follows:

$$z_i^\sharp = \begin{cases} z_i^+ & \text{if} \quad i \in I, \\ z_i & \text{if} \quad i \in J. \end{cases}$$

Here $\|.\|$ stands for an arbitrary norm in $\mathbb{R}^p$, although we note that most often the ℓ^1 norm is chosen for practical reasons. The dual norm $\|.\|_*$ is defined as

$$\|\mu\|_* := \max\{z^t\mu \ ; \ \|z\| \leq 1\}.$$

Usually r is chosen so that $r > \|\mu^k\|_*$, where μ^k is the multiplier associated to d^k.

However this potential suffers from the Maratos effect (Maratos [19], Mayne and Polak [20]). Even when x is close to $\bar{x}$ and $x + d - \bar{x} = O(x - \bar{x})^2$, and r close to $\|\bar{\lambda}\|_*$, it may happen that $\theta_r(x + d) > \theta_r(x)$, and in the context of composite optimization it has been shown that this may occur an infinite number of times. See Yuan [27].

Various remedies have been proposed, the first of them being to make an additional restoration step (Mayne and Polak [20], Gabay [15]), i.e., denoting $\|.\|$ an arbitrary norm in $\mathbb{R}^n$, different from the one in $\mathbb{R}^p$, to compute v^k solution of

$$\min_v \|v\| \ ; g_i(x^k + d^k) + g'(x^k)v = 0, \ i \in I_k^*,$$

where I_k^* is some prediction of the set of active constraints, obtained as a byproduct of the computation of d^k, and to perform a linesearch along the arc

$$\rho \to x^k + \rho d^k + \rho^2 v^k.$$

Other possible remedies are to modify the potential, specifically to use a nondifferentiable augmented Lagrangian [7], and to compare the value of $\theta_r(x^{k+1})$ to the value of θ_r not only at x^k, but also at x^{k-1}, $x^{k-2}, \ldots$ (see, Chamberlain et al. [12], Panier and Tits [21], Bonnans et al. [9]). To our knowledge, all published papers concerning the Maratos effect assume that the strict complementarity hypothesis holds.

1.4. Our contribution. In this paper we present an algorithm that has global and local properties under weak hypotheses on the sequence $\{M^k\}$ of approximations of the Hessian of the Lagrangian. At step k of the algorithm, a parameter $\alpha_k \geq 0$ is set and a direction d^k is computed as a stationary point (if any) of the quadratic problem

$$Q_{\alpha_k}(x^k, M^k) \qquad \min_d \nabla f(x^k)^t d + \frac{1}{2} d^t M^k d + \frac{\alpha_k}{2}\|d\|_2^2 \ ; \ g(x^k) + g'(x^k)d \ll 0,$$

where $\|.\|_2$ is the Euclidean norm. This technique was first introduced by Bell [2].

We note for future reference that the first-order optimality system of $Q_{\alpha_k}(x^k, M^k)$ is, denoting by μ^k the Lagrange multiplier,

$$\begin{aligned} &\nabla f(x^k) + M^k d^k + \alpha_k d^k + g'(x^k)^t \mu^k = 0, \\ &g(x^k) + g'(x^k) d^k \ll 0; \quad \mu_I^k \geq 0; \quad (\mu^k)^t (g(x^k) + g'(x^k) d^k) = 0. \end{aligned} \tag{6}$$

The parameter $r = r_k$ of the exact penalty function $\theta_r(x)$ is adapted at each iteration in order to allow a linesearch; however, null steps may happen and in this case α_k is increased. We prove that $\{r_k\}$ and $\{\alpha_k\}$ are bounded and that any limit point of $\{x^k\}$ satisfies (1). Our hypotheses are as follows. First we assume

$$\{M^k\}, \{x^k\}, \text{ and } \{d^k\} \text{ are bounded.} \tag{7}$$

Note that if upper and lower bounds on x are present, then $\{x^k\}$ and $\{d^k\}$ are necessarily bounded. Second, we assume that

$$\text{the linearized constraints } g(x) + g'(x)d \ll 0 \text{ are feasible and qualified,} \tag{8}$$

which means that for any x, (8) is satisfied for at least one d, and for all (x, d) such that $g(x) + g'(x)d \ll 0$, the gradients of active constraints of this system are linearly independent.

Hypothesis (8) may seem excessively strong. If a nonlinear optimization problem is solved with a random starting point, it might not be satisfied in the neighborhood of the starting point. We have in mind large-scale real-world applications where, in order to solve the problem in a reasonable time, the initial point is the result of some heuristics so that in the region in which the sequence $\{x^k\}$ lies, (8) is satisfied, although a linesearch may be useful. This is, in particular, the case in the optimal load flow problem (see [5]).

We show also how to avoid the Maratos effect using a second-order correction; there we use a very weak hypothesis on the approximation of the Hessian of the Lagrangian. We show also how to combine this result with Theorem 1.1 in order to obtain a superlinearly convergent algorithm.

If second-order derivatives are available we show how to formulate a globally convergent algorithm that reduces locally to Newton's method, and this seems to be the first globally convergent extension of Newton's method for nonconvex constrained optimization. Other globally convergent algorithms have been published, e.g., Han [16] and Fletcher [14], but they assume the approximation to the Hessian to be bounded. The difficulty is that there is no a priori bound for the estimate of the multiplier. We give a device that overcomes this difficulty. We note that Bell [2] has a global convergence result comparable to ours, but he assumes the penalization coefficient r_k to be fixed. By contrast, we deal with the more difficult question of adapting this parameter.

It may seem surprising that the algorithm includes a penalization of the displacement as well as a linesearch; this is due to the presence of constraints. For fixed x, when $\alpha \to \infty$, d solution of $Q_\alpha(x, M)$ converges to $\pi(x)$ solution of

$$\min_d \|d\|_2 \; ; \; g(x) + g'(x)d \ll 0,$$

and (if $\pi(x)$ is nonzero) it may happen that $f(x + \pi(x)) > f(x)$ and $\|g(x + \pi(x))^\sharp\| > \|g(x)^\sharp\|$; in this case the step $\rho_k = 1$ cannot be accepted whenever α_k is large enough.

2. A globally convergent algorithm with fixed penalty parameter. In this section we will present some properties of the exact penalty function that allow the design of a linesearch that extends the one due to Armijo [1] for unconstrained minimization. The ideas that we present here are classical (see [16]) and this section must be considered mainly as a way to prepare the more sophisticated algorithms of §§3 and 4. We note however two specific features. The first is that our hypothesis on the norm is as follows:

$$z \to \|z^\sharp\| \quad \text{is a convex mapping.} \tag{9}$$

This hypothesis is easy to check for the ℓ_p norms, $1 \leq p \leq \infty$, and in that case $\|.^\sharp\|$ coincides with the distance to the cone generating the partial order $x \ll y$. (The property is not true for all norms, e.g., in $\mathbb{R}^2$ consider $\|x\| = |x_1| + |x_2 - x_1|$. If $J = \emptyset$ then $\|x^\sharp\| = \|x^+\| = x_1^+ + |x_2^+ - x_1^+|$. We compute $\|(1,0)^+\| = 2 > \frac{1}{2}\|(1,-1)^+\| + \frac{1}{2}\|(1,1)^+\| = \frac{3}{2}$.)

The second hypothesis is the choice of directions of sufficient descent. For this we use relation (10) below.

We define the directional derivative of θ_r at x in direction d as $\theta_r'(x,d)$. This is well defined, even if (9) does not hold, because $\rho \mapsto g(x+\rho d)^\sharp$ has a directional derivative $w(x,d)$ (that can easily be computed explicitly) and $z \to \|z\|$ is convex and Lipschitz, hence

$$\begin{aligned}\theta_r(x+\rho d) &= f(x) + \rho f'(x)d + r\|g(x)^\sharp + \rho w(x,d)\| + o(\rho)\\ &= \theta_r(x) + \rho[f'(x)d + r\mu^t w(x,d)] + o(\rho),\end{aligned}$$

where μ is some element of the subdifferential of $\|.\|$ at $g(x)^\sharp$.

We define the "linearized" (at point x^k) exact penalty function as follows:

$$\theta^k(d) = f(x^k) + f'(x^k)d + r_k\|(g(x^k) + g'(x^k)d)^\sharp\|.$$

For any d feasible for $Q_{\alpha_k}(x^k, M^k)$, we note that the decrease of the linearized exact penalty function when step $\rho_k = 1$ is accepted is equal to $\Delta_{r_k}(x^k, d)$, where

$$\Delta_r(x,d) := r\|g(x)^\sharp\| - f'(x)d.$$

We say that $\Delta_r(x,d)$ is feasible if

$$\Delta_r(x,d) \geq \|d\|^3. \tag{10}$$

By Δ_k we denote $\Delta_{r_k}(x^k, d^k)$.

LEMMA 2.1. *Let d be a stationary point of $Q_\alpha(x,M)$ and μ the associated Lagrange multiplier. Then*

(i) *if* (9) *holds, then*

$$\theta_r'(x,d) \leq -\Delta_r(x,d). \tag{11}$$

(ii) *The following relations hold:*

$$\Delta_r(x,d) \geq (r - \|\mu\|_*)\|g(x)^\sharp\| + \alpha\|d\|_2^2 + d^tMd + \mu^t(g(x)^\sharp - g(x)), \tag{12}$$

$$\Delta_r(x,d) \geq (r - \|\mu\|_*)\|g(x)^\sharp\| + \alpha\|d\|_2^2 + d^tMd. \tag{13}$$

Proof. (i) From (9) we deduce that

$$\theta'_r(x,d) = f'(x)d + r\eta^t g'(x)d,$$

where η is some subgradient of $\|.^\sharp\|$ at $g(x)$, i.e.,

$$\|z^\sharp\| \geq \|g(x)^\sharp\| + \eta^t(z - g(x)) \quad \forall z \in \mathbb{R}^p.$$

Choosing $z = g(x) + g'(x)d$, and noting that $z^\sharp = 0$, we deduce that $\eta^t g'(x)d \leq -\|g(x)^\sharp\|$, from which (11) follows.

(ii) From (6) we deduce

$$0 = f'(x)d + d^t M d + \alpha\|d\|_2^2 + \mu^t g'(x)d.$$

From the complementarity condition we get that $\mu^t g'(x)d = -\mu^t g(x)$, hence

$$-f'(x)d = d^t M d + \alpha\|d\|_2^2 - \mu^t g(x),$$

and so

$$\begin{aligned}\Delta_r(x,d) &= \alpha\|d\|_2^2 + d^t M d + r\|g(x)^\sharp\| - \mu^t g(x)\\ &= \alpha\|d\|_2^2 + d^t M d + r\|g(x)^\sharp\| - \mu^t g(x)^\sharp + \mu^t(g(x)^\sharp - g(x))\\ &\geq \alpha\|d\|_2^2 + d^t M d + (r - \|\mu\|_*)\|g(x)^\sharp\| + \mu^t(g(x)^\sharp - g(x)).\end{aligned}$$

Thus (12) is proved. Now, as $\mu_I \geq 0$, we get from the definition of $g(x)^\sharp$ that $\mu^t(g(x)^\sharp - g(x)) \geq 0$, and so (13) holds. □

Let x^k be the current point of the algorithm and d^k a stationary point of $Q_{\alpha_k}(x^k, M^k)$. From (13) it follows that, at least if $r_k > \|\mu^k\|_*$ and α_k is large enough, then Δ_k is feasible (note that for α_k sufficiently large, $\|d^k\| \approx \|\pi(x^k)\|$, hence (10) is satisfied).

From (11) it follows that d^k is a descent direction of θ_{r_k} if $\Delta_k > 0$. This allows us to define a linesearch in the following way.

Linesearch rule. LS1. Parameters $\gamma \in (0, 1/2), \beta \in (0,1)$. If Δ_k is feasible then compute $\rho_k = (\beta)^\ell$, with ℓ smallest integer such that

$$\begin{aligned}&\theta_{r_k}(x^k + (\beta)^\ell d^k) \leq \theta_{r_k}(x^k) - (\beta)^\ell \gamma \Delta_k,\\ &x^{k+1} \leftarrow x^k + \rho_k d^k.\end{aligned} \tag{14}$$

We note that (11) and the relation $\gamma < \frac{1}{2}$ imply that (14) is satisfied for ℓ large enough. Hence the linesearch is well defined. In order to analyse the global properties associated with this linesearch we deal in this section with the simple case when r_k is equal to some constant r.

We can now formulate a conceptual algorithm.

ALGORITHM 1

0. Data: $\alpha_0 \geq 0$, M^0 an $n \times n$ symmetric matrix, $x^0 \in \mathbb{R}^n$; $k \to 0$.
1. Computation of (d^k, μ^k) satisfying the optimality system of $Q_{\alpha_k}(x^k, M^k)$.
2. If Δ_k is not feasible, i.e., (10) not satisfied for Δ_k, stop.
3. Perform the linesearch LS1.
4. Choose α_{k+1} and M^{k+1};
   $k \to k+1$,
   go to 1.

THEOREM 2.1. *Assume that* (7) *and* (8) *hold. Let* x^k *be computed by Algorithm* 1 *in which* Δ_k *is assumed to be feasible at each step. Assume that* (α_k, M^k, d^k) *are bounded,* $r_k = r > 0$. *Then* $d^k \to 0$ *and the set of limit points of* (x^k, μ^k) *is a connected subset of the set of solutions of the first-order optimality system* (1).

Proof. We prove that $d^k \to 0$. We note that $\theta_r(x^k)$ decreases, hence converges, so that by (14) $\rho_k \Delta_k \to 0$. Assume that for some subsequence k', we have $(x^{k'}, \alpha_{k'}, M^{k'}, d^{k'}) \to (\hat{x}, \hat{\alpha}, \hat{M}, \hat{d})$ with $\hat{d} \neq 0$. We observe that $\Delta_{k'} \to \hat{\Delta} := \Delta_r(\hat{x}, \hat{d}) > 0$ by (10) and that $\hat{d}$ satisfies the first-order optimality system of $Q_{\hat{\alpha}}(\hat{x}, \hat{M})$; hence $\theta_r'(\hat{x}, \hat{d}) \leq -\hat{\Delta}$ by (11), which implies for ρ small enough

$$\begin{aligned} \theta_r(\hat{x} + \rho\hat{d}) &\leq \theta_r(\hat{x}) - \rho\hat{\Delta} + o(\rho) \\ &\leq \theta_r(\hat{x}) - \frac{2\rho}{3}\hat{\Delta}, \end{aligned}$$

hence for k' large enough by continuity (as $\Delta_{k'} \to \hat{\Delta} > 0$)

$$\theta_r(x^{k'} + \rho d^{k'}) \leq \theta_r(x^{k'}) - \frac{\rho}{2}\Delta_r(x^{k'}, d^{k'}),$$

which proves that $\rho_{k'}$ cannot converge to 0, hence we get $\hat{\Delta} = \lim \Delta_{k'} = 0$, from $\rho_k \Delta_k \to 0$, contradicting $\hat{\Delta} > 0$ obtained from our assumption $\hat{d} \neq 0$.

Now as $d^k \to 0$ for any converging subsequence of $(x^k, \alpha_k, M^k, d^k)$, we can pass to the limit in (6), deducing the boundedness of $\{\mu^k\}$ from (7) and (8), and so that any limit point of (x^k, μ^k) is solution of (1). Now as $d^k \to 0$, the set of limit points of $\{x^k\}$ is connected; by (8) the Lagrange multiplier of (1) (whenever it exists) must depend continuously on x; the conclusion follows. □

In the next section we relax the restrictive hypothesis on r^k and on the a priori feasibility of Δ_k.

3. A general globally convergent algorithm. This section is devoted to the statement and analysis of a globally convergent algorithm, more precisely an algorithm computing a sequence $\{x^k, \mu^k\}$ such that any of its limit-points satisfy the first-order optimality conditions (1). In this algorithm we must update the two parameters r_k and α_k.

For r_k the idea is the following: take $r_k = r_{k-1}$ whenever it is possible, i.e., if $\Delta_{r_{k-1}}(x^k, d^k)$ is feasible and $\rho_k = 1$ is accepted by the linesearch; otherwise choose r_k satisfying $r_k > \|\mu^k\|_*$. In order to make the sequence r_k constant after a finite number of steps we choose $r_k = \max(r_{k-1}, \text{int}(\|\mu^k\|_* + 2))$. Finally the update rule for r_k is as follows:

$$r_k = \begin{cases} r_{k-1} \text{ if } \Delta_{r_{k-1}}(x^k, d^k) \text{ is feasible and} \\ \theta_{r_{k-1}}(x^k + d^k) \leq \theta_{r_{k-1}}(x^k) - \gamma\Delta_{r_{k-1}}(x^k, d^k), \\ \max(r_{k-1}, \text{int}(\|\mu^k\|_* + 2)) \text{ if not.} \end{cases} \tag{15}$$

For α_k the idea is the following. If Δ_k is not feasible or ρ_k is close to 0, then choose $\alpha_{k+1} > \alpha_k + \varepsilon_1$, with $\varepsilon_1 > 0$ (because of Lemma 2.1 this will eventually yield the feasibility of Δ_k). On the other hand, if Δ_k is feasible and $\rho_k = 1$, then α_{k+1} will be taken smaller than α_k.

Finally we mention the possibility of null steps, i.e., when Δ_k is not feasible then x^{k+1} is taken equal to x^k (or equivalently $\rho_k = 0$) and α_k is increased. We now state the algorithm.

ALGORITHM 2

0. Data: $\alpha_0 \geq 0$, M^0 $n \times n$ symmetric matrix, $x^0 \in \mathbb{R}^n$. Parameters $0 < \varepsilon_1 < \varepsilon_2$, $0 < \varepsilon_3 < 1$; $k \leftarrow 0$.
1. Computation of (d^k, μ^k), satisfying the optimality system of $Q_{\alpha_k}(x^k, M^k)$.
2. If $k = 0$, set $r_{-1} \leftarrow \|\mu^0\|_* + 1$.
3. Choice of r_k using the rule (15).
4. If Δ_k is not feasible (null step):
   $\rho_k \leftarrow 0$,
   $x^{k+1} \leftarrow x^k$,
   go to 6.
5. If Δ_k is feasible: perform the linesearch LS1.
6. Update of α_k:
   If $\rho_k = 1$, choose $\alpha_{k+1} \leq \alpha_k/2$.
   If $\rho_k \in (\varepsilon_3, 1)$, choose $\alpha_{k+1} \leq \alpha_k + \varepsilon_2$.
   If $\rho_k \leq \varepsilon_3$ choose $\alpha_{k+1} \in [\alpha_k + \varepsilon_1, \alpha_k + \varepsilon_2]$.
   Choose M^{k+1}.
7. $k \leftarrow k + 1$,
   go to 1.

Remark 3.1. We observe that $\{r_k\}$ increases, and $\{r_k\}$ is bounded iff there exists $r > 0$ such that $r_k = r$ for $k \geq k_0$.

THEOREM 3.1. *Let x^k be computed by Algorithm* 2. *We assume that* (7) *and* (8) *hold. Then* (i) *the sequences $\{r_k\}$, $\{\alpha_k\}$, and $\{\mu^k\}$ are bounded*;

(ii) *the set of limit-points of $\{x^k\}$ is connected, and with each limit point is associated a Lagrange multiplier.*

We give a proof that makes use of some lemmas below.

Proof. (a) We prove that $\{r_k\}$ is bounded. If not, then there exists a subsequence k' with $r_{k'} > r_{k'-1}$, and by (15) $\|\mu^{k'}\|_* \to \infty$. This, and (6)–(8) imply that $\alpha_{k'}\|d^{k'}\| \to \infty$. Now by Lemma 3.1, we obtain $\|g(x^k)^\sharp\| \to 0$ and Lemma 3.2 ensures that for k' large enough, $r_{k'} = r_{k'-1}$, contrary to the definition of $\{k'\}$.

(b) We prove that $\{\alpha_k\}$ is bounded. As $\{r_k\}$ is bounded, we know from Remark 3.1 that r_k is constant, say equal to r for $k \geq k_0$. Lemma 3.3 says that there exists $\hat{\alpha} \geq 0$ such that Δ_k is feasible if $\alpha_k \geq \hat{\alpha}$ and $k \geq k_0$.

From step 6 of Algorithm 2, it follows that $\alpha_{k+1} \leq \alpha_k + \varepsilon_2$ for all k. By Lemma 3.3, if $\alpha_k \geq \hat{\alpha}$ and $k \geq k_0$, then $\rho_k = 1$ and $\alpha_{k+1} \leq \alpha_k/2$; hence $\alpha_{k+1} \leq \max(\hat{\alpha}, \alpha_{k_0}/2) + \varepsilon_2$ whenever $k \geq k_0$.

(c) We now prove (ii). Let $\hat{\alpha}$ be given by Lemma 3.3. By step 6 of Algorithm 2, after at most $\hat{\alpha}/\varepsilon_1$ successive null steps, one has $\alpha_k \geq \hat{\alpha}$; by Lemma 3.3 the next step is not a null step. This means that $\mathbb{K} := \{k \in \mathbb{N}; \rho_k > 0\}$ is not finite. The sequence $\{x^k\}_{k \in \mathbb{K}}$, can be viewed as generated by Algorithm 1, and we deduce from Theorem 2.1 that $\{d^k\}_{k \in \mathbb{K}} \to 0$ and that with each limit-point of $\{x^k\}$ is associated a Lagrange multiplier. As $\{x^k\}_{k \in \mathbb{N}}$ and $\{x^k\}_{k \in \mathbb{K}}$ obviously have the same limit-points, point (ii) follows. □

We now state and prove the three lemmas used in the proof of Theorem 3.1.

LEMMA 3.1. *Let $\{x^k\}$ be computed by Algorithm* 2. *Under hypotheses* (7) *and* (8), *if $r_k \nearrow \infty$ then $\|g(x^k)^\sharp\| \to 0$.*

Proof. (a) Let us verify that $\|g(x^k)^\sharp\|$ converges. Let $m := \inf\{f(x^k), k \in \mathbb{N}\}$. Note that $m > -\infty$ as $\{x^k\}$ is bounded. Then, as $\{r_k\}$ increases $\theta_{r_k}(x^{k+1}) \leq \theta_{r_k}(x^k)$

and so we deduce

$$\|g(x^{k+1})^\sharp\| + \frac{f(x^{k+1})-m}{r_k} \le \|g(x^k)^\sharp\| + \frac{f(x^k)-m}{r_k} \le \|g(x^k)^\sharp\| + \frac{f(x^k)-m}{r_{k-1}},$$

hence $\{\|g(x^k)^\sharp\| + (f(x^k)-m)/r_{k-1}\}$ is a decreasing sequence, and so converges since it is bounded. As $r_k \nearrow \infty$ and $\{f(x^k)\}$ is bounded since $\{x^k\}$ is bounded, it follows that $\|g(x^k)^\sharp\|$ converges.

(b) It suffices now to get a contradiction when assuming that $\lim \|g(x^k)^\sharp\|$ is positive. Let us note that by (6)–(8), if $\{\alpha_k\}$ is bounded, so is $\{\mu^k\}$ hence $\{r_k\}$ cannot go to ∞. Hence we may extract a subsequence k' such that $\alpha_{k'} \to \infty$ and $x^{k'} \to \bar{x}$. It is easily checked that $d^{k'} \to \bar{d} := \pi(\bar{x})$. Now since $\|.^\sharp\|$ is a Lipschitz mapping:

$$\begin{aligned}\theta_{r_{k'}}(x^{k'}) - \theta_{r_{k'}}(x^{k'} + \rho d^{k'}) &= r_{k'}\|g(x^{k'})^\sharp\| - r_{k'}\|g(x^{k'} + \rho d^{k'})^\sharp\| + O(1)\\ &= r_{k'}\|g(\bar{x})^\sharp\| - r_{k'}\|g(\bar{x} + \rho\bar{d})^\sharp\| + o(r_{k'}),\end{aligned}$$

with $o(r_{k'})/r_{k'} \to 0$ uniformly on $\rho \in [0,1]$.

As $g(x^{k'}) + g'(x^{k'})d^k \ll 0$ and (7) holds, it follows that

$$\|g(\bar{x} + \rho\bar{d})^\sharp\| \le (1-\rho)\|g(\bar{x})^\sharp\| + a_0\rho^2 \quad \text{for some } a_0 > 0,$$

hence since $\|.^\sharp\|$ is a Lipschitz mapping and (7) holds:

$$\begin{aligned}\theta_{r_{k'}}(x^{k'}) - \theta_{r_{k'}}(x^{k'} + \rho d^{k'}) &\ge \rho r_{k'}\|g(\bar{x})^\sharp\| - r_{k'}a_0\rho^2 + o(r_{k'})\\ &\ge \rho r_{k'}\|g(x^{k'})^\sharp\| - r_{k'}a_0\rho^2 + o(r_{k'})\\ &= \rho\Delta_{k'} - r_{k'}a_0\rho^2 + o(r_{k'}).\end{aligned}$$

We note that $\Delta_{k'}/r_{k'} \to \|g(\bar{x})^\sharp\|$ which is assumed to be positive. Using this we get for some $a_1 > 0$

$$\theta_{r_{k'}}(x^{k'}) - \theta_{r_{k'}}(x^{k'} + \rho d^{k'}) \ge \Delta_{k'}[\rho - a_1\rho^2 + o(1)]$$

and it follows that $\rho_{k'} \ge \hat{\rho}$ for some $\hat{\rho} > 0$. Then this implies that for some $a_2 > 0$

$$\overline{\lim}\|g(x^{k'+1})^\sharp\|/\|g(x^{k'})^\sharp\| \le 1 - a_2\hat{\rho},$$

in contradiction with our hypothesis. □

LEMMA 3.2. *Let x^k be computed by Algorithm 2. Under the hypotheses* (7) *and* (8), *if a subsequence $\{x^{k'}\}$ satisfies $\|g(x^{k'})^\sharp\| \to 0$ and $\alpha_{k'}\|d^{k'}\| \to \infty$, then*

(i) $\|d^{k'}\|_2/\|\pi(x^{k'})\|_2 \to 1$,

(ii) *for k' large enough, $r_{k'} = r_{k'-1}$.*

Proof. Denote by

$$q_k(d) := \nabla f(x^k)^t d + \frac{1}{2}d^t M^k d + \frac{1}{2}\alpha_k d^t d,$$

the cost function of $Q_{\alpha_k}(x^k, M^k)$. As $\|d^k\|$ is bounded it follows from the unboundedness of $\alpha_{k'}\|d^{k'}\|$ that $\alpha_{k'} \to \infty$. So we see that for $k' \ge k'_0$, $q_{k'}(d)$ is convex, hence $d^{k'}$ is a global solution of $Q_{\alpha_{k'}}(x^{k'}, M^{k'})$. In particular, denoting $\pi^k := \pi(x^k)$, we have

$$q_{k'}(d^{k'}) \le q_{k'}(\pi^{k'}). \tag{16}$$

From the definition of π^k we have $\|\pi^k\|_2 \leq \|d^k\|_2$. On the other hand, dividing (16) by $\alpha_{k'}\|d^{k'}\|_2^2$, remembering that $\alpha_{k'} \to \infty$ we obtain $1 \leq \underline{\lim}\|\pi^{k'}\|_2/\|d^{k'}\|_2$ and point (i) follows.

We now prove (ii). We may assume that $r_k \nearrow \infty$, for otherwise r_k is constant for k large enough (see Remark 3.1) and then the conclusion holds trivially. The idea of the proof is that the penalization term dominates in the linesearch. Indeed,

$$\begin{aligned}\Delta_{r_{k'-1}}(x^{k'}, d^{k'}) &= r_{k'-1}\|g(x^{k'})^\sharp\| - f'(x^{k'})d^{k'} \\ &= r_{k'-1}\|g(x^{k'})^\sharp\| \left(1 - \frac{1}{r_{k'-1}} \frac{f'(x^{k'})d^{k'}}{\|\pi^{k'}\|_2} \cdot \frac{\|\pi^{k'}\|_2}{\|g(x^{k'})^\sharp\|}\right).\end{aligned}$$

We claim that the term between parentheses converges to 1. By point (i), $f'(x^{k'})d^{k'}/\|\pi^{k'}\|_2$ is bounded. As $r_{k-1} \nearrow \infty$ it suffices to prove that $\|\pi^{k'}\|_2/\|g(x^{k'})^\sharp\|$ is bounded. If this is not the case, extracting if necessary a subsequence we may assume that $x^{k'} \to \hat{x}$. As $\|g(x^{k'})^\sharp\| \to 0$, $\hat{x}$ is feasible.

Let $D(x)$ be the set $\{d \in \mathbb{R}^n \ ; \ g(x) + g'(x)d \ll 0\}$. As (8) holds we may apply to the feasible sets of $Q_\alpha(x, M)$ a theorem of Robinson [23] that asserts that for x in a neighborhood of $\bar{x}$, $d = 0$ is at a distance of $D(x)$ of order $\|g(x)^\sharp\|$. It follows that the element of minimum norm $\pi(x)$ satisfies $\|\pi(x)\| = 0(\|g(x)^\sharp\|)$, and this proves our claim.

Now let us prove that if $\Delta_{r_{k'-1}}(x^{k'}, d^{k'})$ is feasible by (i) and the boundedness of $\|\pi^{k'}\|_2/\|g(x^{k'})^\sharp\|$ proved above it follows that $\|d^{k'}\|/\|g(x^{k'})^\sharp\|$ is bounded. Using (7) and $r_{k'-1} \nearrow \infty$ we deduce that $r_{k'-1}\|g(x^{k'})^\sharp\|/\|d^{k'}\|^3 \to \infty$. This and our claim above imply (10), i.e., feasibility of $\Delta_{r_{k'-1}}(x^{k'}, d^{k'})$.

On the other hand

$$\theta_{r_{k'-1}}(x^{k'}) - \theta_{r_{k'-1}}(x^{k'}+d^{k'}) = r_{k'-1}(\|g(x^{k'})^\sharp\| - \|g(x^{k'}+d^{k'})^\sharp\|) + f(x^{k'}) - f(x^{k'}+d^{k'})$$

and so

(17)
$$\theta_{r_{k'-1}}(x^{k'}) - \theta_{r_{k'-1}}(x^{k'} + d^{k'}) = \Delta_{r_{k'-1}}(x^{k'}, d^{k'}) - r_{k'-1}\|g(x^{k'} + d^{k'})^\sharp\| + O(d^{k'})^2.$$

But $\|.^\sharp\|$ is a Lipschitz mapping, and from (6) $(g(x^{k'})+g'(x^{k'})d^{k'})^\sharp = 0$, hence $\|g(x^{k'} + d^{k'})^\sharp\| = O(d^{k'})^2$. Also $d^{k'} = O(g(x^{k'})^\sharp)$ hence, with (17),

$$\theta_{r_{k'-1}}(x^{k'}) - \theta_{r_{k'-1}}(x^{k'} + d^{k'}) = \Delta_{r_{k'-1}}(x^{k'}, d^{k'}) + o(\Delta_{r_{k'-1}}(x^{k'}, d^{k'})).$$

As the rule (15) is used in Algorithm 2, the two previous results imply $r_{k'-1} = r_{k'}$ for any $k' \geq k'_0$, in contradiction with the hypothesis $r_{k'} \nearrow \infty$. □

LEMMA 3.3. *Let x^k be computed by Algorithm 2. Under hypotheses (7) and (8), if $\{r_k\}$ is bounded, then there exists $\hat{\alpha} > 0$ and k_0 such that $\rho_k = 1$ whenever $\alpha_k \geq \hat{\alpha}$ and $k \geq k_0$.*

Proof. Since r_k is bounded, there exists r such that $r_k = r$ for $k \geq k_0$ (cf. Remark 3.1). Using (13) we know that

$$\Delta_k \geq (r - \|\mu^k\|_*)\|g(x^k)^\sharp\| + \alpha_k\|d^k\|_2^2 + d^{kt}M^k d^k,$$

and so, as from (7) $\{M^k\}$ is bounded, we obtain for some $a_3 > 0$

$$\Delta_k \geq (r - \|\mu^k\|_*)\|g(x^k)^\sharp\| + (\alpha_k - a_3)\|d^k\|_2^2,$$

for k large enough. If $\rho_k \neq 1$ then $r_k = r > \|\mu^k\|_*$; hence

$$\Delta_k \geq (\alpha_k - a_3)\|d^k\|_2^2. \tag{18}$$

As $\{d^k\}$ is bounded, we deduce that for α_k large enough, Δ_k is feasible. Now

$$\theta_r(x^k) - \theta_r(x^k+d^k) = r(\|g(x^k)^\sharp\| - \|g(x^k + d^k)^\sharp\|) + f(x^k) - f(x^k + d^k),$$

so since (7) holds and f, g are smooth, we get for some $a_4 > 0$

$$\theta_r(x^k) - \theta_r(x^k + d^k) \geq \Delta_k - a_4\|d^k\|_2^2, \tag{19}$$

hence using (18), for α_k large enough

$$\theta_r(x^k) - \theta_r(x^k+d^k) \geq \frac{1}{2}\Delta_k.$$

As $\gamma < \frac{1}{2}$, the rule (15) ensures that the two previous results imply $\rho_k = 1$, in contradiction with the hypothesis $\rho_k \neq 1$. □

4. A globally and superlinearly convergent algorithm. Let $\bar{x}$ be a local solution of (P) and $\bar{\lambda}$ its associated Lagrange multiplier. We know that Algorithm 2 is not generally superlinearly convergent, even if $x^k \to \bar{x}$ and $M^k \to \nabla_x^2 L(\bar{x}, \bar{\lambda})$. This is due to the Maratos effect (Maratos [19], Mayne and Polak [20]). In this section we show how to adapt the idea of a restoration step in order to accept the unit stepsize. We define

$$I^* := \{i \in I\ ;\ \bar{\lambda}_i > 0\} \cup J,$$
$$I_k^* := \{i \in I\ ;\ \mu_i^k > 0\} \cup J.$$

We first perform a local analysis in which our hypotheses are as follows:

$$\begin{array}{l} \{M^k\},\ \{x^k\},\ \{\alpha_k\},\ \{d^k\} \text{ are given such that } x^k \to \bar{x}, \\ \{M^k\} \text{ and } \{\alpha_k\} \text{ are bounded,} \\ d^k \text{ is stationary point of } Q_{\alpha_k}(x^k, M^k) \text{ and } d^k \to 0. \end{array} \tag{20}$$

We define v^k as the solution of

$$\min_v \|v\| \quad \left\{ \begin{array}{l} g(x^k + d^k) + g'(x^k)v \ll 0, \\ g_i(x^k + d^k) + g_i'(x^k)v = 0 \text{ for any } i \in I_k^*, \end{array} \right. \tag{21}$$

where $\|.\|$ is an arbitrary norm in $\mathbb{R}^n$. Under some reasonable assumptions we show in Proposition 4.1 below that the point $x^k + d^k + v^k$ insures a significant decrease of the exact penalty function. It could be argued that the computation of v^k may be expensive. A possibility ([20], [15]) is to compute v^k solution of

$$\min_v \|v\|\ ;\ g_i(x^k + d^k) + g_i'(x^k)v = 0 \quad \text{for any } i \in I_k^*. \tag{22}$$

If the strict complementarity hypothesis holds, the two corrections are, for k large enough, identical. This indicates that a reasonable way to solve (21) might be to solve (22) first and to check if its solution is also the solution of (21). We start with a technical lemma.

LEMMA 4.1. *Assume that* (8), (20), *and* (21) *hold. Then one has for some* $a > 0$, $k_0 \in \mathbb{N}$

$$I^* \subset I_k^* \text{ for } k > k_0 \ , \tag{23}$$

$$\|v^k\| \leq a\|d^k\|^2, \tag{24}$$

$$g(x^k + d^k + v^k)^\sharp = o(d^k)^2, \tag{25}$$

$$g_{I^*}(x^k + d^k + v^k) = o(d^k)^2. \tag{26}$$

Proof. (a) It follows from (20), (6), and (8) that $\mu^k \to \bar{\lambda}$. So for k large enough, $\{i \in I \ ; \ \bar{\lambda}_i > 0\} \subset \{i \in I \ ; \ \mu_i^k > 0\}$ and thus (23) is proved.

(b) Since (8) holds and by definition of I_k^*:

$$\begin{aligned} &g(x^k) + g'(x^k)d^k \ll 0, \\ &g_i(x^k) + g_i'(x^k)d^k = 0, \ i \in I_k^*, \end{aligned}$$

it follows that

$$\begin{aligned} &g(x^k + d^k) \ll O(d^k)^2, \\ &g_i(x^k + d^k) = O(d^k)^2, \ i \in I_k^*, \end{aligned}$$

hence using again (8), $v^k = O(d^k)^2$.

(c) Expanding $g(x^k + d^k + v^k)$ and using (24) we get

$$g(x^k + d^k + v^k) = g(x^k) + g'(x^k)(d^k + v^k) + \frac{1}{2}(d^k)^t g''(x^k)d^k + o(d^k)^2. \tag{27}$$

Moreover, since (21) implies $(g(x^k + d^k) + g'(x^k)v^k)^\sharp = 0$, expanding $g(x^k + d^k)$ and using $z \to \|z^\sharp\|$ Lipschitz, we obtain

$$\|(g(x^k) + g'(x^k)d^k + \frac{1}{2}(d^k)^t g''(x^k)d^k + g'(x^k)v^k)^\sharp\| = o(d^k)^2.$$

Then, as $z \to \|z^\sharp\|$ is Lipschitz, we have (25).

(d) Since v^k is solution of (21), the expansion of $g_{I_k^*}(x^k + d^k)$ yields

$$g_i(x^k) + g_i'(x^k)d^k + \frac{1}{2}(d^k)^t g_i''(x^k)d^k + g_i'(x^k)v^k = o(d^k)^2 \text{ for any } i \in I_k^*.$$

Hence (26) follows from (23) and (27). □

Then we compute x^{k+1} along the path $\rho \to x^k + \rho d^k + \rho^2 v^k$. The first trial point is $x^k + d^k + v^k$ and if it appears to be necessary to test a small value for ρ_k, then the contribution of v^k is small with respect to the one of d^k, and tl ws us to preserve the descent property on θ_r. Specifically the linesearch is as fouows.

Linesearch rule LS2. Parameters $\gamma \in (0, 1/2)$, $\beta \in (0, 1)$. Compute v^k solution of (21).

If Δ_k is feasible, i.e., (10) holds for Δ_k, then compute $\rho_k = (\beta)^\ell$ with ℓ smallest integer such that

$$\begin{aligned} &\theta_{r_k}(x^k + (\beta)^\ell d^k + (\beta)^{2\ell} v^k) \le \theta_{r_k}(x^k) - (\beta)^\ell \gamma \Delta_k, \\ &x^{k+1} \leftarrow x^k + \rho_k d^k + (\rho_k)^2 d^k. \end{aligned} \tag{28}$$

In order to perform a local analysis we are led to assume that $(\bar{x}, \bar{\lambda})$ (local solution (P) and associated multiplier) satisfies the following strong second-order sufficient condition (Robinson [24]):

$$\text{for any} \quad d \in \ker g'_{I^*}(\bar{x}) \backslash \{0\}, \quad d^t \nabla_x^2 L(\bar{x}, \bar{\lambda}) d > 0. \tag{29}$$

Recalling (8) we see that (29) is stronger than the standard sufficient condition (5), and that both coincide if the strict complementarity hypothesis holds at $\bar{x}$.

The next proposition insures that the new linesearch rule accepts the step $\rho^k = 1$, if x^k is close to $\bar{x}$ satisfying (28). Define

$$\begin{aligned} &d_T^k \text{ orthogonal projection of } d^k \text{ onto } \ker g'_{I^*}(x^k), \\ &d_N^k := d^k - d_T^k, \\ &H := \nabla_x^2 L(\bar{x}, \bar{\lambda}), \end{aligned}$$

and for $z \in \mathbb{R}^p, \tilde{z}$ by

$$\tilde{z}_i = \begin{cases} z_i & \text{if } i \in I^*, \\ z_i^+ & \text{if not.} \end{cases}$$

PROPOSITION 4.1. *Assume* $\{M^k\}$, $\{x^k\}$, $\{\alpha_k\}$, $\{r_k\}$, $\{d^k\}$ *given such that* (8), (20), (21), *and* (29) *hold and* $r_k = r$ *with* $r > \|\bar{\lambda}\|_*$. *If there exists* $\varepsilon_0 > 0$ *such that for* x^k *close enough to* $\bar{x}$,

$$(d_T^k)^t M^k d_T^k + \alpha_k \|d_T^k\|^2 \ge \frac{1}{2(1-\gamma)} (d_T^k)^t H d_T^k + \varepsilon_0 \|d_T^k\|^2 \tag{30}$$

then LS2 *accepts step* $\rho_k = 1$ *for* k *large enough.*

We call (30) the condition of sufficient curvature . A typical condition for the unit step to be accepted is that M^k is close to H, or maybe in some direction only in some sense. Our condition is of a somewhat different nature, as we require the curvature in the tangent direction, i.e., $(d_T^k)^t M^k d_T^k$ to be sufficiently positive. This condition is minimal in the following sense: in the framework of unconstrained optimization, so that $d^k = d_T^k$, then it can be checked that a necessary condition for the unit step to be accepted is (30) in which we change $+\varepsilon_0 \|d_T^k\|^2$ into $-\varepsilon_0 \|d_T^k\|^2$, as shown in Lemma 4.2.

LEMMA 4.2. *Let* $\bar{x}$ *be such that* $\nabla f(\bar{x}) = 0$ *and* $\nabla^2 f(\bar{x}) > 0$. *Let* $x^k \to \bar{x}$ *and* $\{M^k\}$ *be such that* $d^k = -(M^k)^{-1} \nabla f(x^k)$ *vanishes, and*

$$f(x^k + d^k) \le f(x^k) + \gamma f'(x^k) d^k.$$

Then for any $\varepsilon_0 > 0$ *we have for* k *large enough*

$$(d^k)^t M^k d^k \ge \frac{1}{2(1-\gamma)} d^t \nabla^2 f(\bar{x}) d - \varepsilon_0 \|d\|^2.$$

Proof. Choose $\varepsilon_0 > 0$. Set $H^k := \int_0^1 (1-\sigma)\nabla^2 f(x^k + \sigma d^k)d\sigma$. We have

$$f(x^k + d^k) = f(x^k) + f'(x^k)d^k + \frac{1}{2}(d^k)^t H^k d^k$$

with $\|M^k - \nabla^2 f(\bar{x})\| \leq 2(1-\gamma)\varepsilon_0$ for k large enough. It follows that

$$\begin{aligned} 0 &\leq f(x^k) + \gamma f'(x^k)d^k - f(x^k + d^k) \\ &= (\gamma - 1)f'(x^k)d^k - \frac{1}{2}(d^k)^t H^k d^k \\ &= (1-\gamma)\left[(d^k)^t H^k d^k - \frac{1}{2(1-\gamma)}(d^k)^t H^k d^k\right], \end{aligned}$$

so that

$$\begin{aligned} (d^k)^t H^k d^k &\geq \frac{1}{2(1-\gamma)}(d^k)^t H^k d^k \\ &\geq \frac{1}{2(1-\gamma)}(d^k)^t \nabla^2 f(\bar{x}) d^k - \varepsilon_0 \|d^k\|^2, \end{aligned}$$

as was to be proved. □

Before giving the proof we set some preliminary results.

LEMMA 4.3. *For any $n \times n$ symmetric matrix M and for any $\varepsilon > 0$ one has*

$$(d^k)^t M d^k \geq (d_T^k)^t M d_T^k - \varepsilon^2 \|d_T^k\|_2^2 - \|M\|(1 + \|M\|/\varepsilon^2)\|d_N^k\|_2^2, \tag{31}$$

$$(d_T^k)^t M d_T^k \geq (d^k)^t M d^k - \varepsilon^2 \|d_T^k\|_2^2 - \|M\|(1 + \|M\|/\varepsilon^2)\|d_N^k\|_2^2. \tag{32}$$

Proof. Since $d^k = d_T^k + d_N^k$ we get

$$(d^k)^t M d^k = (d_T^k)^t M d_T^k + 2(d_T^k)^t M d_N^k + (d_N^k)^t M d_N^k,$$

hence the following relation holds:

$$|(d^k)^t M d^k (d_T^k)^t M d_T^k| \leq 2\|M\|\|d_T^k\|_2 \|d_N^k\|_2 + \|M\|\|d_N^k\|_2^2. \tag{33}$$

As for all $\varepsilon > 0$, $a > 0$, $b > 0$, one has $2ab = 2(\varepsilon a)(b/\varepsilon) \leq \varepsilon^2 a^2 + b^2/\varepsilon^2$, it comes for $a = \|d_T^k\|_2$ and $b = \|d_N^k\|_2 \|M\|$:

$$2\|M\|\|d_T^k\|_2 \|d_N^k\|_2 \leq \varepsilon^2 \|d_T^k\|_2^2 + \|M\|^2 \|d_N^k\|_2^2/\varepsilon^2,$$

which with (33) gives the conclusion. □

LEMMA 4.4. *Under the hypotheses of Proposition* 4.1, *for k large enough, Δ_k is feasible and the following holds:*

$$\exists a_5 > 0; \qquad \Delta_k \geq a_5 \|d^k\|_2^2 \tag{34}$$

and (γ being the constant involved in LS2, *i.e.*, $\gamma \in (0, \frac{1}{2})$):

$$\exists \varepsilon_1 > 0; \qquad \Delta_k \geq \frac{1}{2(1-\gamma)}(d^k)^t H d^k + \varepsilon_1 \|d^k\|^2. \tag{35}$$

Proof. We restrict our attention to k such that x^k is close to $\bar{x}$.

(a) Preliminaries. It was already noticed (cf. proof of Lemma 4.1(a)) that under our hypotheses $\mu^k \to \bar{\lambda}$. From this result and the hypothesis $r > \|\bar{\lambda}\|_*$ one has for k large enough

$$r - \|\mu^k\|_* \geq (r - \|\bar{\lambda}\|_*)/2,$$

and also it comes for $\zeta := \min\{\bar{\lambda}_i \ ; \ i \in I^* \cap I\}$ (and so $\zeta > 0$) that for k large enough

$$\min\{\mu_i^k \ ; \ i \in I^* \cap I\} > \frac{\zeta}{2}.$$

Hence, as $\mu^k \geq 0$ and $g(x^k)^\sharp \geq g(x^k)$,

$$\begin{aligned}(\mu^k)^t(g(x^k)^\sharp - g(x^k)) &\geq \frac{\zeta}{2} \sum_{i \in I^* \cap I} (g_i(x^k)^\sharp - g_i(x^k)) \\ &= \frac{\zeta}{2} \sum_{i \in I^* \cap I} \max(0, -g_i(x^k)).\end{aligned}$$

From the definition of $g(x^k)^\sharp$ and $\tilde{g}(x^k)$ we finally get with (12) that there exists $\xi > 0$ such that for k large enough

$$\Delta_k \geq \xi\|\tilde{g}(x^k)\| + \alpha_k\|d^k\|^2 + (d^k)^t M^k d^k.$$

Now from (32) with $M = M^k$ it follows that for all $\varepsilon > 0$

$$\Delta_k \geq \xi\|\tilde{g}(x^k)\| + \alpha_k\|d^k\|_2^2 + (d_T^k)^t M^k d_T^k - \varepsilon^2\|d_T^k\|_2^2 - \|M^k\|(1 + \|M^k\|/\varepsilon^2)\|d_N^k\|_2^2.$$

As $\{M^k\}$ is bounded, (8) holds, and d_N^k is solution of

$$\min_d \|d\|_2 \ ; \ g_{I^*}(x^k) + g'_{I^*}(x^k)d = 0,$$

we have

$$d_N^k = O(g_{I^*}(x^k)) = O(\tilde{g}(x^k)), \tag{36}$$

hence for k large enough, since $\|d^k\|_2^2 = \|d_T^k\|_2^2 + \|d_N^k\|_2^2 \geq \|d_T^k\|_2^2$, we get

$$\Delta_k \geq \frac{\xi}{2}\|\tilde{g}(x^k)\| + (\alpha_k - \varepsilon^2)\|d_T^k\|_2^2 + (d_T^k)^t M^k d_T^k. \tag{37}$$

(b) *Proof of* (34). Since (8) and (29) hold, there exists $\delta > 0$ such that for x^k close enough to $\bar{x}$

$$\text{for any} \quad d \in \ker g'_{I^*}(x^k), \quad d^t H d \geq \delta\|d\|^2. \tag{38}$$

From (30), (37), and (38) one has for k large enough

$$\begin{aligned}\Delta_k &\geq \frac{\xi}{2}\|\tilde{g}(x^k)\| + \chi(d_T^k)^t H d_T^k - \varepsilon^2\|d_T^k\|_2^2 + o(d_T^k)^2 \\ &\geq \frac{\xi}{2}\|\tilde{g}(x^k)\| + (\chi\delta - \varepsilon^2)\|d_T^k\|_2^2 + o(d_T^k)^2.\end{aligned}$$

Hence for k large enough, taking $\varepsilon = \sqrt{\chi\delta/3}$ we get

$$\Delta_k \geq \frac{\xi}{2}\|\tilde{g}(x^k)\| + \chi\frac{\delta}{2}\|d_T^k\|_2^2.$$

Using (36) we deduce (34).

Hence, as we assume that $d^k \to 0$, it follows that Δ_k is asymptotically feasible.

(c) We now prove (35). From (30) and (37) we have for k large enough

$$\Delta_k \geq \frac{\xi}{2}\|\tilde{g}(x^k)\| + \chi(d_T^k)^t H d_T^k + \varepsilon^2\|d_T^k\|_2^2.$$

Then using (32) of Lemma 4.2 with $M = H$, we obtain for all $\varepsilon > 0$

$$\Delta_k \geq \frac{\xi}{2}\|\tilde{g}(x^k)\| + \chi(d^k)^t H d^k - 2\varepsilon^2\|d_T^k\|_2^2 - \|H\|(1 + \|H\|/\varepsilon^2)\|d_N^k\|_2^2.$$

Hence one has from (36) for k large enough,

$$\Delta_k \geq \chi(d^k)^t H d^k - 2\varepsilon^2\|d_T^k\|_2^2. \tag{39}$$

Take θ in (0,1) such that $\theta\chi = \frac{1}{2(1-\gamma)}$. It follows with (34), (39), and the relation $\|d_T^k\| \leq \|d^k\|$ that

$$\begin{aligned}\Delta_k &= \theta\Delta_k + (1-\theta)\Delta_k \\ &\geq \frac{1}{2(1-\gamma)}(d^k)^t H^k d^k + [(1-\theta)a_5 - 2\varepsilon^2]\|d^k\|^2.\end{aligned}$$

We now choose $\varepsilon_1 = (1-\theta)a_5/2$ and $\varepsilon = \sqrt{\varepsilon_1}$, so that $\varepsilon_1 = (1_\theta)a_5 - 2\varepsilon^2$; relation (35) follows. □

LEMMA 4.5. *Assume that the hypothesis of Proposition* 4.1 *holds. Define* $\bar{x}^k := x^k - \bar{x}$. *Then* $\bar{x}^k = O(d^k)$.

Proof. From the optimality system of $Q_{\alpha_k}(x^k, M^k)$ we deduce that x^k satisfies the optimality system of

$$\min_x f(x) + x^t c^k \;;\; g(x) + e^k \ll 0$$

with $c^k := M^k d^k + \alpha_k d^k$ and $e^k := g'(x^k)d^k$ and so $c^k = O(d^k)$ and $e^k = O(d^k)$.

Consider the family of perturbed problems

$$\min_x f(x) + x^t c \;;\; g(x) + e \ll 0. \tag{$P_{c,e}$}$$

For $\bar{c} = 0$, $\bar{e} = 0$, $\bar{x}$ is a local solution of $P_{\bar{c},\bar{e}}$ satisfying the regularity hypothesis (the linearized constraints are qualified) and the strong second-order sufficient condition. It follows that for c^k, e^k close to 0, any local solution x^k of the first-order optimality system of (P_{c^k,e^k}) which is in a given neighbourhood of $\bar{x}$ is such that $\bar{x}^k = O(c^k) + O(e^k) = O(d^k)$ (see Robinson [25]). □

Proof of Proposition 4.1. We know from Lemma 4.4 that, for k large enough, Δ_k is feasible; so it remains to check that (28) holds with $\ell = 0$. Define

$$\begin{aligned}\hat{x}^{k+1} &:= x^k + d^k + v^k, \\ \check{x}^{k+1} &:= \hat{x}^{k+1} - \bar{x}, \\ a &:= \theta_r(x^k) - \theta_r(\hat{x}^{k+1}).\end{aligned}$$

We must prove that $a \geq \gamma\Delta_k$. Indeed

$$a = L(x^k, \bar{\lambda}) - L(\hat{x}^{k+1}, \bar{\lambda}) + \bar{\lambda}^t(g(\hat{x}^{k+1}) - g(x^k)) + r(\|g(x^k)^\sharp\| - \|g(\hat{x}^{k+1})^\sharp\|). \tag{40}$$

Expanding $L(., \bar{\lambda})$ at $\bar{x}$ one obtains

$$L(x^k, \bar{\lambda}) - L(\hat{x}^{k+1}, \bar{\lambda}) = \frac{1}{2}(\bar{x}^k)^t H\bar{x}^k - \frac{1}{2}(\check{x}^{k+1})^t H\check{x}^{k+1} + o(\bar{x}^k)^2 + o(\check{x}^{k+1})^2. \tag{41}$$

Moreover one has

$$\begin{aligned}(\bar{x}^k)^t H\bar{x}^k - (\check{x}^{k+1})^t H\check{x}^{k+1} &= (\bar{x}^k - \check{x}^{k+1})^t H(\bar{x}^k + \check{x}^{k+1}) \\ &= -(d^k + v^k)^t H(2\bar{x}^k + d^k + v^k).\end{aligned}$$

So using (24) we get

$$(\bar{x}^k)^t H\bar{x}^k - (\check{x}^{k+1})^t H\check{x}^{k+1} = -2(d^k)^t H\bar{x}^k - (d^k)^t Hd^k + o(d^k)^2,$$

then, since (24) yields $\check{x}^{k+1} = \bar{x}^k + d^k + o(d^k)$ and using Lemma 4.4, we obtain from (41)

$$L(x^k, \bar{\lambda}) - L(\hat{x}^{k+1}, \bar{\lambda}) = -(d^k)^t H\bar{x}^k - \frac{1}{2}(d^k)^t Hd^k + o(d^k)^2.$$

Then from (25), (26), and Lemma 4.4 we get from (40)

$$a = -(d^k)^t H\bar{x}^k - \frac{1}{2}(d^k)^t Hd^k - \bar{\lambda}^t g(x^k) + r\|g(x^k)^\sharp\| + o(d^k)^2. \tag{42}$$

On the other hand we have

$$\begin{aligned}\Delta_k &= r\|g(x^k)^\sharp\| - f'(x^k)d^k \\ &= r\|g(x^k)^\sharp\| - \nabla_x L(x^k, \bar{\lambda})^t d^k + \bar{\lambda}^t g'(x^k)d^k.\end{aligned}$$

So expanding $\nabla_x L(x^k, \bar{\lambda})$ at $\bar{x}$ and using Lemma 4.4

$$\Delta_k = r\|g(x^k)^\sharp\| - (\bar{x}^k)^t Hd^k + \bar{\lambda}^t g'(x^k)d^k + o(d^k)^2.$$

Using (23) and the complementarity condition in (6), we get for any $i \in I^*$

$$g_i(x^k) + g_i'(x^k)d^k = 0,$$

hence $-\bar{\lambda}^t g(x^k) = \bar{\lambda}^t g'(x^k)d^k$ and so

$$\Delta_k = r\|g(x^k)^\sharp\| - (\bar{x}^k)^t Hd^k - \bar{\lambda}^t g(x^k) + o(d^k)^2.$$

Plugging this in (42) we obtain

$$a = -\frac{1}{2}(d^k)^t Hd^k + \Delta_k + o(d^k)^2.$$

We want $a \geq \gamma\Delta_k$, i.e.,

$$(1 - \gamma)\Delta_k \geq \frac{1}{2}(d^k)^t Hd^k + o(d^k)^2$$

which is a consequence of (35). □

According to §1.4, we now present an algorithm that is globally convergent (as in §3) and that converges superlinearly when we assume that $\{M^k\}$ approximates in some sense the Hessian of the Lagrangian of problem (P) (using §4 and properties of Newton type algorithms quoted in §1.2). We now state the algorithm.

ALGORITHM 3.

Perform the same steps as in Algorithm 2, replacing LS1 by LS2.

THEOREM 4.1. *Let x^k be computed by Algorithm* 3. *We assume that* (7) *and* (8) *hold. Then*

(i) *$\{r_k\}$ and $\{\alpha_k\}$ are bounded.*

(ii) *The set of limit points of $\{x^k\}$ is connected and to each of them is associated a Lagrange multiplier.*

(iii) *Assume that the algorithm computes the solution d^k of minimal norm of the optimality system of $Q_{\alpha_k}(x^k, M^k)$. If to some $\bar{x}$ limit-point of $\{x^k\}$ is associated a multiplier $\bar{\lambda}$ such that* (29) *and* (30) *hold, then $x^k \to \bar{x}$ and $\rho_k = 1$ for k large enough. If in addition $P^k[(\nabla_x^2 L(\bar{x}, \bar{\lambda}) - M^k)d^k] = o(d^k)$, then the convergence is superlinear.*

Proof. The arguments for proving (i), (ii) are essentially the same as for Theorem 3.1. As they are rather long we do not reproduce them in detail but rather analyse where the differences are.

Proof of (i). This proof relies on extension of Lemmas 3.1–3.3 for Algorithm 3. Lemma 3.1 is proved by checking that $\|g(x^k)^\sharp\|$ converges if $r_k \searrow \infty$, and on a first-order expansion (in ρ) of $\|g(x^k + \rho d^k)^\sharp\|$. These last arguments have immediate extensions as the paths $\rho \to x^k + \rho d^k$ and $\rho \to x^k + \rho d^k + \rho^2 v^k$ have the same first-order expansion, the term v^k being uniformly bounded. Simple considerations allow an immediate extension of Lemma 3.2. For the extension of Lemma 3.3, estimate (18) on Δ_k is still valid, and (19) also holds, but with a possibly different constant a_4 (because of the additional term v^k) and the conclusion follows. Now the same discussion of points (a), (b) of proof of Theorem 3.1 can be used in order to check that (i) holds.

Proof of (ii). The mechanism of adaptation of $\{x^k\}$ and Lemma 3.3 imply that $\rho_{k'} > 0$ for an infinite subsequence $\{k'\}$, and we may suppose that $\{x^{k'}\} \to \hat{x}$. If $\Delta_{k'} \to 0$ it follows that $d^{k'} \to 0$, hence $\hat{x}$ is a stationary point of (P). If not, assuming $d^{k'} \to \hat{d} \neq 0$ and $v^{k'} \to \hat{v}$ (note that $v^{k'}$ is bounded by (24) hence has limit-points) expanding $\rho \to \theta_r(\hat{x} + \rho\hat{d} + \rho^2\hat{v})$ as in the proof of Theorem 2.1 we deduce that $\rho_{k'}$ cannot converge to 0, hence $\theta_r(x^{k'}) \to \infty$, which is impossible. Henceforth $\hat{d} = 0$ and point (ii) follows. Using (29), (30), and applying the sensitivity result of Robinson [25] to $\bar{d} = 0$ solution of $Q(\bar{x}, \nabla_x^2 L(\bar{x}, \lambda))$ we deduce that $d^k \to 0$ for the considered subsequence.

Proof of (iii). That $\rho_{k'} = 1$ asymptotically for the subsequence $\{x^{k'}\} \to \bar{x}$ is then a consequence of Proposition 4.1. Indeed $\mu^{k'} \to \bar{\lambda}$ as $d^{k'} \to 0$ and (M^k, α_k) are bounded. If $\rho_{k'} < 1$ for a subsequence then (for k' large enough) $r_{k'+1} > \|\bar{\lambda}\|_*$, hence $r > \|\bar{\lambda}\|_*$ and the hypotheses of Proposition 4.1 are satisfied: it follows that $\rho_{k'} = 1$ for k' large enough, hence $\alpha_k \searrow 0$ at a geometric rate.

Now by (29), $\bar{x}$ is an isolated stationary point (see Robinson [25]), and by point (ii) is an isolated limit-point of $\{x^k\}$. As the set of limit points of $\{x^k\}$ is connected it follows that all the sequence converges to $\bar{x}$.

If in addition $P^k[(\nabla_x^2 L(\bar{x}, \bar{\lambda}) - M^k)d^k] = o(d^k)$, then as $\alpha_k \searrow 0$, $P^k[(\nabla_x^2 L(\bar{x}, \bar{\lambda}) - (M^k + \alpha_k I)d^k] = o(d^k)$ hence by Theorem 1.2, $x^k + d^k - \bar{x} = o(x^k - \bar{x})$. As $v^k =$

$O(d^k)^2 = o(x^k - \bar{x})$ we get $x^{k+1} - \bar{x} = o(x^k - \bar{x})$, as desired. □

We now formulate an algorithm that, assuming that the second derivatives of f and g are known, is an extension of Newton's method in the sense that, when x^k is close to some $\bar{x}$ satisfying (29), it computes d^k using $M^k = \nabla_x^2 L(x^k, \mu^{k-1})$ where μ^{k-1} is the multiplier associated to d^{k-1}, and $x^k \to \bar{x}$ with a quadratic rate. The rule is as follows:

$$\text{choose } M^{k+1} = \nabla_x^2 L(x^{k+1}, \lambda^{k+1}) \text{ with}$$

$$\lambda^{k+1} := \begin{cases} \mu^k & \text{if } \alpha_k \|d^k\| + \|M^k d^k\| \leq 1, \\ \mu^k/(1 + \alpha_k \|d^k\| + \|M^k d^k\|) & \text{if not.} \end{cases} \tag{43}$$

THEOREM 4.2. (a) *Let* $\{x^k\}$ *be computed by Algorithm* 3 *with* $\{M^k\}$ *computed by* (43). *We assume that* $\{x^k\}$, $\{d^k\}$ *are bounded, that* (8) *holds and that* $\alpha_{k+1} = 0$ *if* $\rho_k = 1$. *Then points* (i), (ii) *of Theorem* 4.1 *still hold.*

(b) *In addition, if* $\bar{x}$ *satisfying* (29) *is limit-point of* x^k *and* d^k *is the solution of minimal norm of the optimality system of* $Q_{\alpha_k}(x^k, M^k)$, *then all the sequence* $\{x^k\}$ *converges to* $\bar{x}$ *with a quadratic rate.*

Proof. (a) In order to get point (i), (ii) of Theorem 4.1 we must just check that $\{M^k\}$ is bounded; indeed λ^{k+1} is bounded by (8) and (42) hence so is $\{M^k\}$.

Now as $d^k \to 0$ and (M^k, α_k) are bounded, it follows that $\mu^k \to \bar{\lambda}$ and $\lambda^{k+1} = \mu^k$ by (43), hence $M^k \to \nabla_x^2 L(\bar{x}, \bar{\lambda})$ and point (iii) of Theorem 4.1 implies that $\rho_k = 1$ since (30) obviously holds which implies the convergence of all the sequence to $\bar{x}$ at a quadratic rate by Theorem 1.2. □

Acknowledgments. Thanks are due to M. J. D. Powell, P. Terpolilli, and an anonymous referee for their remarks that improved a preliminary version of this paper.

REFERENCES

[1] L. ARMIJO, *Minimization of function having Lipschitz continuous first partial derivatives*, Pacific J. Math., 16 (1966), pp. 1–3.

[2] B. M. BELL, *Global convergence of a semi-infinite optimization method*, Appl. Math. Optim., 21 (1990), pp. 69–88.

[3] A. BEN-TAL, *Second-order and related extremality conditions in nonlinear programming*, J. Optimi. Theory Appl., 21 (1980), pp. 143–165.

[4] D. P. BERTSEKAS, *Constrained optimization and Lagrange multipliers methods*, Academic Press, New York, 1982.

[5] G. BLANCHON, J. F. BONNANS, AND J. C. DODU, *Optimisation des réseaux électriques de grande taille*, in Lecture Notes Inf. Control Sci. no 144, A. Bensoussan and J. L. Lions, eds., 1990, pp. 423–431.

[6] J. W. BOGGS, P. W. TOLLE, AND P. WANG, *On the local convergence of quasi-Newton methods for constrained optimization*, SIAM J. Control Optim., 20 (1982), pp. 161–171.

[7] J. F. BONNANS, *Asymptotic admissibility of the unit stepsize in exact penalty methods*, SIAM J. Control Optim., 27 (1989), pp. 631–641.

[8] ———, *Local analysis of Newton type methods for variational inequalities and nonlinear programming*, Applied Math. Optim., 29 (1994), pp. 161–186.

[9] J. F. BONNANS, E. R. PANIER, A. L. TITS, AND J. L. ZHOU, *Avoiding the Maratos effect by means of a nonmonotone line search* II: *Inequality constrained problems–feasible iterates*, SIAM J. Numer. Anal., 29 (1992), pp. 1187–1202.

[10] J. F. BONNANS AND C. POLA, *A trust region interior point algorithm for linearly constrained optimization*, INRIA Report no 1948, 1993.

[11] E. CASAS AND C. POLA, *A sequential generalized quadratic programming algorithm using exact L_1 penalty functions*, Optim. Methods Software, to appear.

[12] R. M. CHAMBERLAIN, C. LEMARÉCHAL, H. C. PEDERSEN, AND M. J.D. POWELL, *The watchdog technique for forcing convergence in algorithm for constrained optimization*, Math. Programming Stud., 16 (1982), pp. 1–17.
[13] I. I. EREMIN, *The penalty method in convex programming*, Dokl. Akad. Nauk. SSSR 8 (1966); English translation, Soviet Math. Dokl., 8(1966), pp. 459–462.
[14] R. FLETCHER, *Practical Methods of Optimization*, 2nd edition, Wiley, Chichester, 1987.
[15] D. GABAY, *Reduced quasi-Newton methods with feasibility improvement for nonlinearly constrained optimization*, Math. Programming Stud., 16 (1982), pp. 18–44.
[16] P. HAN *A globally convergent method for nonlinear programming*, J. Optim. Theory Appl., 22 (1977), pp. 297–309.
[17] P. HAN AQND O. L. MANGASARIAN, *Exact penalty functions in nonlinear programming*, Math. Programming, 17 (1979), pp. 251–269.
[18] N. M. JOSEPHY, *Newton's method for generalized equations*, Tech. Summary Report 1965, Mathematics Research Center, University of Wisconsin-Madison, 1979.
[19] N. MARATOS, *Exact Penalty Function Algorithms for Finite Dimensional and Control Optimization Problems*, Ph. D. thesis, University of London, UK, 1978.
[20] D. Q. MAYNE AND E. POLAK *A superlinearly convergent algorithm for constrained optimization problems*, Math. Programming Stud., 16 (1982), pp. 45–61.
[21] 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.
[22] B. N. PSCHENICHNY AND Y. M. DANILIN, *Numerical methods in extremal problems*, MIR, Moscow, 1975. (English Translation 1978.)
[23] S. M. ROBINSON, *Stability theory for systems of inequalities, Part* II: *Differentiable nonlinear systems*, SIAM J. Numer. Anal., 13 (1976), pp. 497–513.
[24] ———, *Strongly regular generalized equations*, Math. Oper. Res., 5 (1980), pp. 43–62.
[25] ———, *Generalized equations and their applications, Part* II *: application to nonlinear programming*, Math. Programming Study, 19 (1982), pp. 200–221.
[26] R. B. WILSON, *A Simplicial Algorithm for Concave Programming,* Ph.D. thesis, Harvard University, Cambridge, MA, 1963.
[27] Y. YUAN, *An example of only linear convergence of trust region algorithms for non-smooth optimization*, IMA J. Numer. Anal., 4 (1984), pp. 327–335.
[28] W. I. ZANGWILL, *Nonlinear programming via penalty functions*, Man. Science, 13 (1967), pp. 344–358.

SIAM J. OPTIMIZATION
Vol. 5, No. 4, pp. 813–834, November 1995

005

GLOBAL OPTIMALITY CONDITIONS AND THEIR GEOMETRIC INTERPRETATION FOR THE CHEMICAL AND PHASE EQUILIBRIUM PROBLEM*

Y. JIANG†, W. R. SMITH†, AND G. R. CHAPMAN†

Abstract. A general class of nonlinear optimization problems motivated by the chemical and phase equilibrium problem in chemical thermodynamics is discussed. The relationships between Kuhn–Tucker points and the global minimum are investigated. The relationships are interpreted in terms of common tangent planes to a function with domain in $\mathcal{R}^{N-1}$ associated with the objective function, whose domain is in $\mathcal{R}^N$. Necessary and sufficient conditions for a global minimum are established, which we call the reaction tangent-plane criterion. The conditions related to the common tangent planes may be considered separately from the feasibility conditions, which allows a novel geometric interpretation of the overall optimality conditions. Illustrative examples are provided of systems involving up to three chemical species.

Key words. chemical equilibrium, phase equilibrium, Kuhn–Tucker points, global optimality

AMS subject classifications. 80A10, 80A15, 90C90, 49M37

1. Introduction.

The optimization problems discussed in this paper arise in the study of equilibria in multiphase multireaction chemical systems (see, for example, [1], [2]). We consider an objective function of the form

$$F(\boldsymbol{Y}) = \sum_{k=1}^{K} y_k f(\boldsymbol{x}_k), \tag{1}$$

where

$$\boldsymbol{Y} = (y_1, y_2, \ldots, y_K, \boldsymbol{x}_1^T, \boldsymbol{x}_2^T, \ldots, \boldsymbol{x}_K^T, z)^T \tag{2}$$

with $y_1, y_2, \ldots, y_K$ real, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K$ distinct points in E^n, and f a twice differentiable real-valued function defined on the strictly positive region of E^n.

Problem P is to minimize F subject to the constraints

$$\sum_{k=1}^{K} y_k \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{c}z = \boldsymbol{b}, \tag{3}$$

$$\sum_{k=1}^{K} y_k - z = 0, \tag{4}$$

$$y_k \geq 0,\ \boldsymbol{x}_k \in \boldsymbol{X} \quad (1 \leq k \leq K), \tag{5}$$

where $\boldsymbol{A}$ is a real $m \times n$ real matrix of rank m, $n \geq m$, $\boldsymbol{b} \neq \boldsymbol{0} \in E^m$, $\boldsymbol{c} \in E^m$, $\boldsymbol{X}$ is an open set in the strictly positive region of E^n, and we assume that the problem has at least one feasible solution. The most significant and unusual feature of this problem is the fact that K is unknown a priori. This feature is related to the behaviour of f, which is generally nonconvex.

* Received by the editors September 30, 1993; accepted for publication June 30, 1994. This research was partially supported by the Natural Sciences and Engineering Research Council of Canada and the University of Guelph.

† Department of Mathematics and Statistics, University of Guelph, Guelph, Ontario, N1G 2W1, Canada (wsmith@msnet.mathstat.uoguelph.ca).

K is the number of distinct chemical phases that may exist (see next section) and the number of positive y_k is the number of phases actually observed at chemical equilibrium.

In the thermodynamic setting, one is interested in the globally optimal solution of Problem P, which typically has multiple local optima. Conditions characterizing such local optima for an important special case of Problem P ($\boldsymbol{A} = \boldsymbol{I}$, $z = \text{constant} = 1$), the so-called *phase equilibrium problem*, are due originally to Gibbs [3]. In recent years, there has been renewed interest in phase equilibrium problems, as well as in the general case involving chemical reactions as well as phase equilibrium [4]–[13]. These treatments typically involve analyses of problems that may be interpreted as involving particular forms of the function f in (1) which arise in chemical applications.

The purpose of this paper is to derive some general analytical results for Problem P, which is shown to be a general formulation of the chemical and phase equilibrium problem, and to interpret these results geometrically. Of significant importance is their independence of any particular form for the function f. Limited versions of some of these results have been presented previously for the special case of phase equilibrium problems (for example, [8], [9]). The more general chemical and phase equilibrium problem has also recently been considered by Smith, Missen, and Smith [13], using an approach similar to that of this paper. Our problem formulation is different and the results are more general.

We derive our results from the application of Kuhn–Tucker theory. It will be shown that the resulting optimality conditions separate naturally into feasibility conditions (involving all elements of $\boldsymbol{Y}$ in (2)) and conditions related to the function f (involving only $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_K\}$). We denote a Kuhn–Tucker point for Problem P by $\boldsymbol{Y}^*$, where

$$\boldsymbol{Y}^* = (y_1^*, y_2^*, \ldots, y_K^*, (\boldsymbol{x}_1^*)^T, (\boldsymbol{x}_2^*)^T, \ldots, (\boldsymbol{x}_K^*)^T, z)^T. \tag{6}$$

For Problem P, the following conditions are proved.

1. *Kuhn–Tucker point and common tangent plane conditions.* $\boldsymbol{Y}^*$ is a Kuhn–Tucker point for Problem P if and only if it is feasible and, for those $y_k^* > 0$, the corresponding $\boldsymbol{x}_k^*$ are points of tangency to a common tangent plane to f whose normal lies in the image of $\boldsymbol{A}^T$. More precisely, the necessary and sufficient conditions for a Kuhn–Tucker point are that it is feasible and if $y_k^* > 0$ then there exist constants $\boldsymbol{\alpha}^*$ and β^* such that

$$\nabla f(\boldsymbol{x}_k^*) = \boldsymbol{A}^T \boldsymbol{\alpha}^*, \tag{7}$$

$$f(\boldsymbol{x}_k^*) = (\boldsymbol{\alpha}^*)^T \boldsymbol{A} \boldsymbol{x}_k^* + \beta^*. \tag{8}$$

2. *Local optimality of Kuhn–Tucker point.* Suppose that Problem P satisfies the condition that the equation $\boldsymbol{Ax} = \boldsymbol{b}$ has a solution. Then, if $\boldsymbol{Y}^*$ corresponds to a local minimum, then for any $y_k^* > 0$ it must hold that

$$\nabla^2 f(\boldsymbol{x}_k^*) \geq \boldsymbol{0}. \tag{9}$$

3. *Global optimality conditions.* A Kuhn–Tucker point $\boldsymbol{Y}^*$ corresponds to a global minimum of F if and only if the common tangent plane of item 1 above is nowhere above the graph of f (i.e., is a supporting hyperplane to the graph of f). We will refer to these conditions as the *reaction tangent-plane criterion.*

In the next section, we show how Problem P arises in the study of chemical reaction and phase equilibria in classical thermodynamics. From the point of view of the analysis that follows, the chemical equilibrium problem and the phase equilibrium problem are special cases of Problem P.

Section 3 establishes the equivalence of Kuhn–Tucker points and the common tangent plane condition. Section 4 discusses the second-order necessary condition for a local minimum, and in §5 the necessary and sufficient conditions for global optimality are derived. In §6, we discuss the geometrical interpretation of our results and give several examples.

2. Genesis of the problems. An important problem in chemical thermodynamics is the general chemical reaction and phase equilibrium problem for mixtures, which we refer to as the *chemical equilibrium* problem. A special case of this is the *phase equilibrium* problem. We discuss each of these in turn.

Consider a closed chemical system of N compounds, or substances, composed of M chemical elements at fixed temperature (T) and pressure (P). Each compound i is distinguished by a formula vector $\boldsymbol{a}'_i \in E^M$ $(1 \leq i \leq N)$, whose entries a'_{ji} denote the number of atoms of element j per molecule of substance i. The formula vectors are the columns of the $M \times N$ formula matrix

$$\boldsymbol{A}' = (\boldsymbol{a}'_1, \boldsymbol{a}'_2, \ldots, \boldsymbol{a}'_N).$$

The substances may exist in a number of distinct phases, indexed by $k = 1, 2, \ldots, \pi$. Let $n_{ki} \geq 0$ be the mass (in mol) of substance i in phase k, and

$$y_k = \sum_{i=1}^{N} n_{ki} \geq 0 \tag{10}$$

be the total mass of substances in phase k. Let $\boldsymbol{x}'_k \in E^N$ be the composition vector for the substances in phase k, so that x'_{ki} is the mole-fraction of substance i in phase k, where

$$n_{ki} = y_k x'_{ki}, \tag{11}$$

$$x'_{ki} \geq 0, \tag{12}$$

and

$$\sum_{i=1}^{N} x'_{ki} = 1. \tag{13}$$

Finally, conservation of mass for the closed system implies the constraints

$$\sum_{k=1}^{\pi} y_k \boldsymbol{A}' \boldsymbol{x}'_k = \boldsymbol{b}', \tag{14}$$

where $\boldsymbol{b}' \geq \boldsymbol{0}$ is an M-vector giving the abundance of the atomic elements in the system. We remark in passing that $b'_i > 0$ except for electronic charge, where it is 0. Similarly, $a'_{ji} > 0$ except for ionic substances, in which case it is unrestricted in sign.

At fixed T and P, equilibrium occurs when the Gibbs free energy of the system is a global minimum subject to (10)–(14) [2]. The Gibbs free energy is given by

$$G(\boldsymbol{y}, \boldsymbol{x}') = \sum_{k=1}^{\pi} y_k g(\boldsymbol{x}'_k). \tag{15}$$

The function g, giving the molar Gibbs free energy of a phase, is twice differentiable for $\boldsymbol{x} \geq \boldsymbol{0}$, and satisfies

$$\lim_{x_i \to 0+} \frac{\partial g(\boldsymbol{x})}{\partial x_i} = -\infty. \tag{16}$$

This condition implies that any phase that is present at equilibrium (i.e., has $y_k > 0$) has nonzero amount of every substance in the phase (see the Appendix for proof). A further consequence of (16) is that we may strengthen the inequality (12). The accumulated constraints are then

$$y_k \geq 0, \quad \boldsymbol{x}'_k > 0 \quad (1 \leq k \leq \pi), \tag{17}$$

$$\sum_{i=1}^{N} x'_{ki} = 1. \tag{18}$$

Problem P is obtained by setting $n = N - 1$ and eliminating the variables $x'_{k,N}$, via

$$x'_{k,N} = 1 - \sum_{i=1}^{n} x'_{ki} \quad (1 \leq k \leq \pi). \tag{19}$$

This gives

$$\boldsymbol{x}_k = (x'_{k1}, x'_{k2}, \ldots, x'_{kn})^T,$$
$$f(\boldsymbol{x}_k) = g(\boldsymbol{x}'_k) \quad (1 \leq k \leq \pi),$$

and $\boldsymbol{X} = \boldsymbol{S}$, where $\boldsymbol{S}$ is the standard open simplex

$$\boldsymbol{S} = \left\{ \boldsymbol{x} \in E^n; \sum_{i=1}^{n} x_i < 1, \, x_i > 0 \right\}.$$

Finally, we set

$$\boldsymbol{A} = (\boldsymbol{a}'_1 - \boldsymbol{a}'_N, \boldsymbol{a}'_2 - \boldsymbol{a}'_N, \ldots, \boldsymbol{a}'_n - \boldsymbol{a}'_N), \tag{20}$$

$$\boldsymbol{b} = \boldsymbol{b}' \quad \boldsymbol{c} = \boldsymbol{a}'_N, \tag{21}$$

and $K = \pi$, $m = M$. Then Problem P is obtained by a straightforward calculation.

For the phase equilibrium problem, $\boldsymbol{A}'$ is the identity matrix, $\boldsymbol{b}' > \boldsymbol{0}$, and $N = M$, thus making it a special case of the chemical equilibrium problem. The phase equilibrium problem is then

$$\begin{aligned} &\min \quad \sum_{k=1}^{K} y'_k g(\boldsymbol{x}'_k), \\ &\text{s.t.} \quad \sum_{k=1}^{K} y'_k \boldsymbol{x}'_k = \boldsymbol{b}', \\ &y'_k \geq 0, \quad \boldsymbol{x}'_k > \boldsymbol{0}, \end{aligned} \tag{22}$$

$$\sum_{i=1}^{N} x'_{ki} = 1. \tag{23}$$

From (22) and (23), we get

$$\sum_{k=1}^{K} y'_k = \sum_{i=1}^{N} b'_i = Q,$$

where $Q > 0$ is a constant. From (23),

$$x'_{k,N} = 1 - \sum_{i=1}^{N-1} x'_{ki} \quad (1 \le k \le N).$$

Substituting in the final equation of (22) gives

$$\sum_{k=1}^{K} y'_k \left(1 - \sum_{i=1}^{N-1} x'_{ki}\right) = b'_N;$$

or equivalently,

$$Q - \sum_{i=1}^{N-1} \sum_{k=1}^{K} y'_k x'_{ki} = Q - \sum_{i=1}^{N-1} b'_i.$$

However, this is a trivial consequence of the first $N-1$ members of (22), and hence the final equation of (22) is redundant. Now let

$$\boldsymbol{x}_k = (x'_{k1}, x'_{k2}, \ldots, x'_{k,N-1})^T,$$
$$f(\boldsymbol{x}_k) = g(\boldsymbol{x}'_k) \quad (1 \le k \le K),$$
$$\boldsymbol{b} = (b'_1, b'_2, \ldots, b'_{N-1})^T/Q,$$
$$\boldsymbol{y} = \boldsymbol{y}'/Q,$$

and let $n = N - 1$. The phase equilibrium problem then becomes

$$\min \quad \sum_{k=1}^{K} y_k f(\boldsymbol{x}_k),$$

$$\text{s.t.} \quad \sum_{k=1}^{K} y_k \boldsymbol{x}_k = \boldsymbol{b}, \tag{24}$$

$$\sum_{k=1}^{K} y_k = 1, \tag{25}$$

$$0 \le y_k \le 1, \quad \boldsymbol{x}_k \in \boldsymbol{S}.$$

This is thus a special case of Problem P with $\boldsymbol{X} = \boldsymbol{S}$, $\boldsymbol{A}$ is the identity matrix, $\boldsymbol{c} = \boldsymbol{0}$, $m = n = N - 1$, and $z \equiv 1$. Note that $\boldsymbol{x}_k$ and $\boldsymbol{b}$ have the same dimension, and that $\boldsymbol{x}_k \in \boldsymbol{S}, \boldsymbol{b} \in \text{cl}(\boldsymbol{S})$.

3. Kuhn–Tucker conditions and the common tangent plane. For Problem P, the Fritz–John necessary conditions for $\boldsymbol{Y}^*$ in (6) to be a local minimum of F are that there exists $\alpha_0^*, \boldsymbol{\alpha}^* \in E^m, \beta^*, \gamma^* \in E^K$ such that

$$\alpha_0^* f(\boldsymbol{x}_k^*) - \boldsymbol{\alpha}^{*T} \boldsymbol{A} \boldsymbol{x}_k^* - \beta^* - \gamma_k^* = 0, \tag{26}$$

$$\alpha_0^* y_k^* \nabla f(\boldsymbol{x}_k^*) - y_k^* \boldsymbol{A}^T \boldsymbol{\alpha}^* = 0 \quad (1 \le k \le K), \tag{27}$$

$$-\boldsymbol{\alpha}^{*T} \boldsymbol{c} + \beta^* = 0, \tag{28}$$

$$\boldsymbol{h} = \sum_{k=1}^{K} y_k^* \boldsymbol{A}\boldsymbol{x}_k^* - \boldsymbol{b} + \boldsymbol{c}z^* = \boldsymbol{0}, \tag{29}$$

$$h_0 = \sum_{k=1}^{K} y_k^* - z^* = 0, \tag{30}$$

$$\gamma_k^* y_k^* = 0 \quad (1 \le k \le K), \tag{31}$$

$$\boldsymbol{g} = \boldsymbol{y}^* \ge \boldsymbol{0}, \tag{32}$$

$$\alpha_0^* \ge 0, \quad \gamma_k^* \ge 0, \quad \boldsymbol{x}_k \in X \ (1 \le k \le K). \tag{33}$$

Note that for the phase equilibrium problem, $z =$ constant, so (28) does not arise; the theorems that follow do not depend on this equation. Its significance will become apparent in the discussion of the chemical equilibrium problem (§6).

Let $I = \{k : y_k^* = 0\}$. In view of (29) and (30), $I \neq \{1, 2, \ldots, N\}$.

LEMMA 1 (Constraint qualifications). *If the rows of the matrix* $\boldsymbol{A}$ *are linearly independent (i.e., if* rank $\boldsymbol{A} = m$*), then* $\{\nabla h_0, \nabla h_j \ (1 \le j \le m), \nabla g_k \ (k \in I)\}$ *are linearly independent, where all functions are evaluated at* $\boldsymbol{Y}^*$.

Proof. We have

$$\nabla h_0^T = (1, \ldots, 1, \boldsymbol{0}, \ldots, \boldsymbol{0}, -1), \tag{34}$$

$$\nabla h_j^T = (\boldsymbol{r}_j^T \boldsymbol{x}_1^*, \ldots, \boldsymbol{r}_j^T \boldsymbol{x}_K^*, y_1^* \boldsymbol{r}_j^T, \ldots, y_K^* \boldsymbol{r}_j^T, c_j), \tag{35}$$

$$\nabla g_k^T = (\boldsymbol{e}_k^T, \boldsymbol{0}, \ldots, \boldsymbol{0}, 0), \tag{36}$$

where $\boldsymbol{r}_j^T$ is the jth row of $\boldsymbol{A}$, c_j is the jth element of $\boldsymbol{c}$, $(1 \le j \le m)$, and $\boldsymbol{e}_k$ is the kth unit vector.

Consider

$$c_0' \nabla h_0 + \sum_{j=1}^{m} d_j' \nabla h_j + \sum_{k \in I} c_k' \nabla g_k = \boldsymbol{0}. \tag{37}$$

From coordinates $K + 1, \ldots, K(n + 1)$ of (37), we obtain

$$\sum_{j=1}^{m} y_k^* d_j' \boldsymbol{r}_j^T = \boldsymbol{0} \quad (1 \le k \le K). \tag{38}$$

Since not all y_k^* are zero, it follows that

$$\sum_{j=1}^{m} d_j' \boldsymbol{r}_j^T = \boldsymbol{0}. \tag{39}$$

Therefore $d_1' = d_2' = \cdots = d_m' = 0$, since the rows of $\boldsymbol{A}$ are linearly independent. From the final coordinate of (37),

$$-c_0' + d_j' c_j = 0. \tag{40}$$

Since $d'_j = 0$ $(1 \leq j \leq m)$, we must have $c'_0 = 0$. Now (37) becomes

$$\sum_{k \in I} c'_k \nabla g_k = \mathbf{0}. \tag{41}$$

It then follows from (36) that $c'_k = 0$ $(i \in I)$, and the lemma is proved.

Since the constraint qualifications hold, the Kuhn–Tucker necessary conditions give the following theorem.

THEOREM 1. *Let f be differentiable and the rows of $\boldsymbol{A}$ be linearly independent. If $\boldsymbol{Y}^*$ is a local minimum for Problem P, then there exists $\boldsymbol{\alpha}^* \in E^m, \beta^*, \boldsymbol{\gamma}^* \in E^K$ such that*

$$f(\boldsymbol{x}_k^*) - \boldsymbol{\alpha}^{*T} \boldsymbol{A} \boldsymbol{x}_k^* - \beta^* - \gamma_k^* = 0, \tag{42}$$

$$y_k^* (\nabla f(\boldsymbol{x}_k^*) - \boldsymbol{A}^T \boldsymbol{\alpha}^*) = \mathbf{0} \quad (1 \leq k \leq K), \tag{43}$$

$$\sum_{k=1}^{K} y_k^* \boldsymbol{A} \boldsymbol{x}_k^* = \boldsymbol{b} - \boldsymbol{c} z^*, \tag{44}$$

$$\sum_{k=1}^{K} y_k^* = z^*, \tag{45}$$

$$\gamma_k^* y_k^* = 0, \quad \gamma_k^* \geq 0, \quad y_k^* \geq 0, \quad \boldsymbol{x}_k^* \in \boldsymbol{X} \ (1 \leq k \leq K). \tag{46}$$

Furthermore, when z is a variable (a chemical equilibrium problem), we have also that

$$-\boldsymbol{\alpha}^{*T} \boldsymbol{c} + \beta^* = 0. \tag{47}$$

The Kuhn–Tucker conditions may be characterized geometrically by the following corollary.

COROLLARY 1. *If $\boldsymbol{Y}^*$ is a local minimum for Problem P, and $J = \{k : y_k^* > 0\}$, then there exists an m-dimensional hyperplane*

$$\Theta^*(\boldsymbol{x}) = (\boldsymbol{\alpha}^*)^T \boldsymbol{A} \boldsymbol{x} + \beta^*,$$

which is tangent to $f(\boldsymbol{x})$ at the points $\{\boldsymbol{x}_k^* : k \in J\}$.

Proof. Let $k \in J$. Then $y_k^* > 0$ and (43) gives

$$\nabla f(\boldsymbol{x}_k^*) = \boldsymbol{A}^T \boldsymbol{\alpha}^*. \tag{48}$$

Furthermore, $\gamma_k^* = 0$ (by 46), so (42) can be written

$$f(\boldsymbol{x}_k^*) = (\boldsymbol{A}^T \boldsymbol{\alpha}^*)^T \boldsymbol{x}_k^* + \beta^*. \tag{49}$$

An important consequence of Theorem 1 and Corollary 1 is the fact that $\boldsymbol{Y}^*$ is a Kuhn–Tucker point if and only if it is feasible and, for those $y_k^* > 0$, the corresponding $\boldsymbol{x}_k$ are points of tangency to a common tangent plane to f whose normal lies in the image of $\boldsymbol{A}^T$.

4. A second-order necessary condition. The Lagrangian for Problem P is

$$L(\boldsymbol{Y}) = \sum_{k=1}^{K} y_k f(\boldsymbol{x}_k) - \boldsymbol{\alpha}^T \left(\sum_{k=1}^{K} y_k \boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b} + \boldsymbol{c}z \right) - \beta \left(\sum_{k=1}^{K} y_k - z \right) - \sum_{k=1}^{K} \gamma_k y_k. \tag{50}$$

It follows that

$$\nabla^2 L = \begin{pmatrix} 0 & \cdots & 0 & \nabla f(\boldsymbol{x}_1)^T - \boldsymbol{\alpha}^T \boldsymbol{A} & \cdots & 0 & 0 \\ & \ddots & & & \ddots & & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & \nabla f(\boldsymbol{x}_K)^T - \boldsymbol{\alpha}^T \boldsymbol{A} & 0 \\ \nabla f(\boldsymbol{x}_1) - \boldsymbol{A}^T \boldsymbol{\alpha} & \cdots & 0 & y_1 \nabla^2 f(\boldsymbol{x}_1) & \cdots & 0 & 0 \\ & \ddots & & & \ddots & & \vdots \\ 0 & \cdots & \nabla f(\boldsymbol{x}_K) - \boldsymbol{A}^T \boldsymbol{\alpha} & 0 & \cdots & y_K \nabla^2 f(\boldsymbol{x}_K) & 0 \\ 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \end{pmatrix}. \tag{51}$$

Let

$$\tilde{\boldsymbol{U}} = (\boldsymbol{u}^T, \boldsymbol{v}_1^T, \ldots, \boldsymbol{v}_K^T, w)^T, \tag{52}$$

where $\boldsymbol{u} \in E^K$ and $\boldsymbol{v}_k \in E^n$. Then

$$\tilde{\boldsymbol{U}}^T \nabla^2 L \tilde{\boldsymbol{U}} = 2 \sum_{k=1}^{n} \boldsymbol{v}_k^T (\nabla f(\boldsymbol{x}_k) - \boldsymbol{A}^T \boldsymbol{\alpha}) v_k + \sum_{k=1}^{K} y_k \boldsymbol{v}_k^T \nabla^2 f(\boldsymbol{x}_k) \boldsymbol{v}_k. \tag{53}$$

Define the set

$$M = \{\tilde{\boldsymbol{U}} : \nabla g_k^T(\boldsymbol{x}^*)\tilde{\boldsymbol{U}} = 0,\ k \in I;\ \nabla \boldsymbol{h}_0^T \tilde{\boldsymbol{U}} = 0;\ \nabla \boldsymbol{h}_j^T \tilde{\boldsymbol{U}} = 0,\ 1 \le j \le m\}. \tag{54}$$

Referring to (34)–(36), this becomes, in the notation of Lemma 1,

$$M = \left\{ \tilde{\boldsymbol{U}} : u_k = 0,\ k \in I;\ \sum_{k=1}^{K} u_k = w;\ \sum_{k=1}^{K} u_k \boldsymbol{A}\boldsymbol{x}_k + \sum_{k=1}^{K} y_k \boldsymbol{A}\boldsymbol{v}_k + \boldsymbol{c}w = \boldsymbol{0} \right\}. \tag{55}$$

The following results directly from Bazaraa and Shetty [14].

THEOREM 2. *Let f be twice differentiable and the rows of $\boldsymbol{A}$ be linearly independent. If $\boldsymbol{Y}^*$ is a local minimum for Problem P, then there exists $\boldsymbol{\alpha}^* \in E^m$, $\beta^*, \gamma_k^* \in E$ such that*

$$\tilde{\boldsymbol{U}}^T \nabla^2 L^* \tilde{\boldsymbol{U}} = 2 \sum_{k=1}^{K} \boldsymbol{v}_k^T (\nabla f(\boldsymbol{x}_k^*) - \boldsymbol{A}^T \boldsymbol{\alpha}^*) u_k + \sum_{k=1}^{K} y_k^* \boldsymbol{v}_k \nabla^2 f(\boldsymbol{x}_k^*) \boldsymbol{v}_k \ge 0 \tag{56}$$

for $\tilde{\boldsymbol{U}} \in M^*$, where * denotes evaluation at $\boldsymbol{Y}^*$.

COROLLARY 2. *Suppose Problem P satisfies the condition that $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ has a solution (which, from the assumed properties of $\boldsymbol{A}$ and $\boldsymbol{b}$, must be nontrivial). If $\boldsymbol{Y}^*$ is a local minimum, then for any $y_k^* > 0$ (i.e., $k \notin I$), $\nabla^2 f(\boldsymbol{x}_k^*)$ is positive semidefinite, which we write as*

$$\nabla^2 f(\boldsymbol{x}_k^*) \ge \boldsymbol{0}. \tag{57}$$

Proof. From (48) of the proof of Corollary 1,

$$\nabla f(\boldsymbol{x}_k^*) - \boldsymbol{A}^T\boldsymbol{\alpha}^* = \boldsymbol{0} \tag{58}$$

for $k \notin I$. Furthermore, from Theorem 2, for $\tilde{\boldsymbol{U}} \in M^*$, $u_k = 0$ for $k \in I$. Hence, the first term on the right-hand side of (56) vanishes, and we have

$$\sum_{k=1}^{K} y_k^* \boldsymbol{v}_k^T \nabla^2 f(\boldsymbol{x}_k^*)\boldsymbol{v}_k \geq 0. \tag{59}$$

Suppose there is an ℓ, $\ell \notin I$ such that

$$\nabla^2 f(\boldsymbol{x}_\ell^*) < 0. \tag{60}$$

Now consider a $\tilde{\boldsymbol{U}}$ given by

$$\tilde{\boldsymbol{U}} = (y_1^*, y_2^*, \ldots, y_K^*, \boldsymbol{0}^T, \ldots, \boldsymbol{0}^T, \boldsymbol{v}_\ell^T, \boldsymbol{0}^T, \ldots, \boldsymbol{0}^T, z^*)^T, \tag{61}$$

where $\boldsymbol{v}_\ell / y_\ell^* \neq \boldsymbol{0}$ is a (nontrivial) solution of

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}. \tag{62}$$

It is easy to verify that $\tilde{\boldsymbol{U}} \in M^*$, so (56) becomes

$$\tilde{\boldsymbol{U}}^T \nabla^2 L \tilde{\boldsymbol{U}} = y_\ell^* \boldsymbol{v}_\ell^T \nabla^2 f(\boldsymbol{x}_\ell^*)\boldsymbol{v}_\ell < 0, \tag{63}$$

which contradicts (59).

5. Necessary and sufficient conditions for a global minimum. For notational convenience, we let

$$\tilde{\boldsymbol{Y}} = (\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_K, \tilde{\boldsymbol{x}}_1^T, \tilde{\boldsymbol{x}}_2^T, \ldots, \tilde{\boldsymbol{x}}_K^T, \tilde{z})^T \tag{64}$$

denote a feasible point for Problem P. Similarly, let

$$\boldsymbol{Y}^+ = (y_1^+, \ldots, y_K^+, (\boldsymbol{x}_1^+)^T, \ldots, (\boldsymbol{x}_K^+)^T, z^+)^T \tag{65}$$

denote a global minimum for Problem P. Then from (3) and (4),

$$\sum_{k=1}^{K} y_k^+ \boldsymbol{A}\boldsymbol{x}_k^+ + \boldsymbol{c}z^+ = \sum_{k=1}^{K} \tilde{y}_k \boldsymbol{A}\tilde{\boldsymbol{x}}_k + \boldsymbol{c}\tilde{z}. \tag{66}$$

Multiplying by $(\boldsymbol{\alpha}^+)^T$ and rearranging, we obtain

$$\sum_{k=1}^{K} y_k^+ (\boldsymbol{\alpha}^+)^T \boldsymbol{A}\boldsymbol{x}_k^+ = \sum_{k=1}^{K} \tilde{y}_k (\boldsymbol{\alpha}^+)^T \boldsymbol{A}\tilde{\boldsymbol{x}}_k + (\boldsymbol{\alpha}^+)^T \boldsymbol{c}(\tilde{z} - z^+). \tag{67}$$

Furthermore, $\boldsymbol{Y}^+$ is a Kuhn–Tucker point, and by Theorem 1 and Corollary 1, there exist Lagrange multipliers $\boldsymbol{\alpha}^+, \beta^+$ such that if

$$\Theta^+(\boldsymbol{x}) = (\boldsymbol{\alpha}^+)^T \boldsymbol{A}\boldsymbol{x} + \beta^+, \tag{68}$$

then

$$f(\boldsymbol{x}_k^+) = \Theta^+(\boldsymbol{x}_k^+) \quad \text{for } y_k^+ > 0. \tag{69}$$

Note that, by (47), we can write (for the chemical equilibrium problem)

$$\Theta^+(\boldsymbol{x}) = (\boldsymbol{\alpha}^+)^T(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{c}). \tag{70}$$

LEMMA 2. *In the above notation,*

$$F(\boldsymbol{Y}^+) = \sum_{k=1}^{K} \tilde{y}_k \Theta^+(\tilde{\boldsymbol{x}}_k). \tag{71}$$

Proof. By (1), (4), (68), and (69), we have

$$F(\boldsymbol{Y}^+) = \sum_{k=1}^{K} y_k^+ (\boldsymbol{\alpha}^+)^T \boldsymbol{A}\boldsymbol{x}_k^+ + z^+\beta^+. \tag{72}$$

Using (67), we may write

$$\begin{aligned} F(\boldsymbol{Y}^+) &= \sum_{k=1}^{K} \tilde{y}_k (\boldsymbol{\alpha}^+)^T \boldsymbol{A}\tilde{\boldsymbol{x}}_k + (\boldsymbol{\alpha}^+)^T \boldsymbol{c}(\tilde{z} - z^+) + z^+\beta^+ \\ &= \sum_{k=1}^{K} \tilde{y}_k (\boldsymbol{\alpha}^+)^T \boldsymbol{A}\tilde{\boldsymbol{x}}_k + \tilde{z}\beta^+ + [(\beta^+ - (\boldsymbol{\alpha}^+)^T)(z^+ - \tilde{z})]. \end{aligned} \tag{73}$$

For the phase equilibrium problem, z is constant and so

$$z^+ - \tilde{z} = 0. \tag{74}$$

For the chemical equilibrium problem,

$$\beta^+ - (\boldsymbol{\alpha}^+)^T \boldsymbol{c} = 0. \tag{75}$$

Thus, for Problem P, (73) becomes (using (4))

$$F(\boldsymbol{Y}^+) = \sum_{k=1}^{K} \tilde{y}_k ((\boldsymbol{\alpha}^+)^T \boldsymbol{A}\tilde{\boldsymbol{x}}_k + \beta^+) \tag{76}$$

and the lemma follows from (68).

LEMMA 3. *$\boldsymbol{Y}^+$ is a global minimum for Problem P if and only if it is a Kuhn–Tucker point, and for any feasible $\boldsymbol{Y}$*

$$\sum_{k=1}^{K} \tilde{y}_k (f(\tilde{\boldsymbol{x}}_k) - \Theta^+(\tilde{\boldsymbol{x}}_k)) \geq 0. \tag{77}$$

Proof. It follows from (1) and Lemma 2 that

$$F(\tilde{\boldsymbol{Y}}) - F(\boldsymbol{Y}^+) = \sum_{k=1}^{K} \tilde{y}_k (f(\tilde{\boldsymbol{x}}_k) - \Theta^+(\tilde{\boldsymbol{x}}_k)), \tag{78}$$

and the result follows.

The essence of the following theorem is that a Kuhn–Tucker point of F is a global minimum for Problem P if and only if the common tangent plane to which it gives rise is a supporting hyperplane of f, i.e., if and only if the graph of $f(\boldsymbol{x})$ is never below the tangent plane.

THEOREM 3 (Reaction tangent-plane criterion). *$\boldsymbol{Y}^+$ is a global minimum for P if and only if it is a Kuhn–Tucker point, and*

$$f(\boldsymbol{x}) - \Theta^+(\boldsymbol{x}) \geq 0 \quad \forall \boldsymbol{x} \in \boldsymbol{X}, \tag{79}$$

where $\Theta^+(\boldsymbol{x})$ is defined by (68).

Proof. Let $\boldsymbol{Y}^+$ be a Kuhn–Tucker point and assume (79) holds. If $\tilde{\boldsymbol{Y}}$ is a feasible point for Problem P, then $\tilde{y}_k \geq 0$, $\tilde{\boldsymbol{x}}_k \in \boldsymbol{X}$. Thus by (79),

$$\tilde{y}_k(f(\tilde{\boldsymbol{x}}_k) - \Theta^+(\tilde{\boldsymbol{x}}_k)) \geq 0 \quad (1 \leq k \leq K). \tag{80}$$

Summing from 1 to K, we obtain (77), and hence $\boldsymbol{Y}^+$ is a global minimum by Lemma 3.

Now let $\boldsymbol{Y}^+$ be a global minimum for Problem P. By Theorem 1 it is a Kuhn–Tucker point. Suppose there exists $\boldsymbol{x}_0 \in \boldsymbol{X}$ such that

$$f(\boldsymbol{x}_0) - \Theta^+(\boldsymbol{x}_0) < 0. \tag{81}$$

Now there exists at least one of $y_1^+, \ldots, y_K^+$, say y_j^+, which is strictly positive (since $\boldsymbol{b} \neq \boldsymbol{0}$ in (3)). From (69) and (81),

$$f(\boldsymbol{x}_0) - f(\boldsymbol{x}_j^+) < \Theta^+(\boldsymbol{x}_0) - \Theta^+(\boldsymbol{x}_j^+). \tag{82}$$

Consider $\tilde{\boldsymbol{Y}} \in E^{(K+1)(n+1)+1}$ defined by

$$\begin{aligned}\tilde{\boldsymbol{Y}} = (&y_1^+, \ldots, y_{j-1}^+, y_0, y_0', y_{j+1}^+, \ldots, y_K^+, (\boldsymbol{x}_1^+)^T, \ldots, \\ &(\boldsymbol{x}_{j-1}^+)^T, (\boldsymbol{x}_0)^T, (\boldsymbol{x}_0')^T, (\boldsymbol{x}_{j+1}^+)^T, \ldots, (\boldsymbol{x}_L^+)^T, z^+)^T,\end{aligned} \tag{83}$$

where

$$y_0 = \frac{t}{t+1} y_j^+, \qquad y_0' = \frac{1}{t+1} y_j^+, \tag{84}$$

$$\boldsymbol{x}_0' = t(\boldsymbol{x}_j^+ - \boldsymbol{x}_0) + \boldsymbol{x}_j^+ \qquad (t > 0). \tag{85}$$

(We remark that this $\tilde{\boldsymbol{Y}}$ is physically equivalent to replacing phase j by two phases with compositions given by $\boldsymbol{x}_0$ and $\boldsymbol{x}_0'$ and amounts given by y_0 and y_0'.) Since $\boldsymbol{X}$ is open, $\boldsymbol{x}_0' \in \boldsymbol{X}$ for t sufficiently small. Furthermore, a straightforward calculation shows that

$$y_0 + y_0' = y_j^+, \qquad y_0 \boldsymbol{A}\boldsymbol{x}_0 + y_0' \boldsymbol{A}\boldsymbol{x}_0' = y_j^+ \boldsymbol{A}\boldsymbol{x}_j^+, \tag{86}$$

so that $\tilde{\boldsymbol{Y}}$ is feasible for Problem P.

By (83) and (84) we have

$$\begin{aligned} F(\tilde{\boldsymbol{Y}}) - F(\boldsymbol{Y}^+) &= y_0 f(\boldsymbol{x}_0) + y_0' f(\boldsymbol{x}_0') - y_j^+ f(\boldsymbol{x}_j^+) \\ &= y_j^+ \left(\frac{t}{1+t} f(\boldsymbol{x}_0) + \frac{1}{1+t} f(\boldsymbol{x}_0') - f(\boldsymbol{x}_j^+) \right). \end{aligned} \tag{87}$$

A Taylor series expansion about $\boldsymbol{x}_j^+$, based on (85), gives

$$f(\boldsymbol{x}_0') = f(\boldsymbol{x}_j^+) + t\nabla f(\boldsymbol{x}_j^+)^T(\boldsymbol{x}_j^+ - \boldsymbol{x}_0) + o(t^2).$$

By (48) and (68), this gives

$$f(\boldsymbol{x}_0') = f(\boldsymbol{x}_j^+) + t(\Theta^+(\boldsymbol{x}_j^+) - \Theta^+(\boldsymbol{x}_0)) + o(t^2). \tag{88}$$

By (82) we obtain, for sufficiently small $t > 0$,

$$f(\boldsymbol{x}_0') < f(\boldsymbol{x}_j^*) + t(f(\boldsymbol{x}_j^+) - f(\boldsymbol{x}_0)).$$

When rearranged and divided by $(t+1)$, this becomes

$$\frac{t}{t+1}f(\boldsymbol{x}_0) + \frac{1}{t+1}f(\boldsymbol{x}_0') - f(\boldsymbol{x}_j^+) < 0. \tag{89}$$

Combined with (87) this gives

$$F(\tilde{\boldsymbol{Y}}) - F(\boldsymbol{Y}^+) < 0,$$

which contradicts the minimality of $F(\boldsymbol{Y}^+)$.

6. Discussion. In this section, we discuss the geometric interpretation of our results for both the phase equilibrium and the chemical equilibrium problem. We then give illustrations for the cases of binary and ternary chemical systems ($n = 1$ and $n = 2$, respectively). Although the geometric interpretation of the equilibrium criteria for the simplest case, that of binary phase equilibrium problems, has been discussed previously (for example, [8], [9]) in a similar way to that contained in the following, this is not the case for ternary systems. Our discussion in the case of the more general chemical equilibrium problem arises from the new analysis of this paper. The essence of the approach is to consider separately the conditions arising from the common tangent-plane conditions and the reaction tangent-plane criterion on the one hand, and from the feasibility conditions on the other.

The global minimum for Problem P is characterized by the Kuhn–Tucker conditions and the global optimality criterion of Theorem 3. The Kuhn–Tucker conditions can be separated into the feasibility conditions

$$\sum_{k=1}^{K} y_k^* \boldsymbol{A}\boldsymbol{x}_k^* = \boldsymbol{b} - \boldsymbol{c}z^*, \tag{90}$$

$$\sum_{k=1}^{K} y_k^* = z^*, \tag{91}$$

$$\boldsymbol{x}_k^* \in \boldsymbol{X}, \tag{92}$$

$$y_k^* \geq 0 \quad (1 \leq k \leq K), \tag{93}$$

and the common tangent plane conditions (involving only $\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_K^*$)

$$f(\boldsymbol{x}_k^*) = \alpha^{*T}\boldsymbol{A}\boldsymbol{x}_k^* + \beta^*, \qquad \nabla f(\boldsymbol{x}_k^*) = \boldsymbol{A}^T\alpha^*. \tag{94}$$

We remark that, for the chemical equilibrium problem, it also holds that

$$\beta^* = \alpha^{*T}\boldsymbol{c} \quad (k \in J), \tag{95}$$

where J is the set $\{k : y_k^* > 0\}$.

We say that $\{\boldsymbol{x}_k^*; k \in J\}$ is a CT *set for* f *relative to* $\boldsymbol{A}$ if there exists α^*, β^* such that (94) holds. This means that the $\boldsymbol{x}_k^*$ are points of common tangency of a tangent hyperplane to f. It follows directly from Corollary 1 that $\boldsymbol{Y}^* = (y_1^*, \ldots, y_K^*, \boldsymbol{x}_1^{*T}, \ldots, \boldsymbol{x}_K^{*T}, z)^T$ is a Kuhn–Tucker point for P if and only if it is feasible and $\{\boldsymbol{x}_k^* : k \in J\}$ is a CT set for f relative to $\boldsymbol{A}$. The fact that the Kuhn–Tucker conditions can be separated in this way is a combined consequence of the linearity of the objective function F in $y_1, \ldots, y_K$, and the bilinearity with respect to $y_1, \ldots, y_K$ and $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K$ of the constraints.

The conditions for a global minimum separate naturally into those involving the CT set and the reaction tangent-plane criterion of Theorem 3 on the one hand, and the feasibility conditions on the other. If a CT set satisfies Corollary 2, it is said to be a *local* CT *set* (LCT), and if it also satisfies Theorem 3, it is said to be a *supporting* CT *set* (SCT). Thus, a CT set for f relative to $\boldsymbol{A}$ yields a global minimum if and only if it is also an SCT set and it is feasible.

A key to the geometrical interpretation of our results in the case of the chemical equilibrium problem is (70) (which does not apply to the case of the phase equilibrium problem). Equation (70) states that the intersection of the tangent supporting hyperplane to f with the plane $f = 0$ is constrained to coincide with the intersection of the m hyperplanes

$$\boldsymbol{A}\boldsymbol{x} + \boldsymbol{c} = \boldsymbol{0}. \tag{96}$$

Varying the unknown Lagrange multiplier vector $\boldsymbol{\alpha}$ in (70) is geometrically equivalent to rotating the tangent hyperplane subject to this constraint.

In the following, we consider general examples of both phase and chemical equilibrium in binary and ternary systems. The purpose is to show the interactions between the tangent plane conditions and the feasibility conditions. For binary systems, we use an objective function f given by

$$\begin{aligned} f(x) = {} & 100x + 120(1-x) + x\ln x + (1-x)\ln(1-x) \\ & + x(1-x)(-157 - 183(2x-1) - 2679(2x-1)^2 + 417(2x-1)^3 \\ & + 11207(2x-1)^4 + 341(2x-1)^5 - 117212(2x-1)^6, \end{aligned} \tag{97}$$

where x is the mole fraction of one of the species. We refer to values of x as the *composition* of the system. For ternary systems we use f given by

$$\begin{aligned} f(x_1, x_2) = {} & 0.76x_1 + 0.77x_2 + 0.78(1 - x_1 - x_2) \\ & + x_1\ln x_1 + x_2\ln x_2 + (1 - x_1 - x_2)\ln(1 - x_1 - x_2) \\ & + 10x_1x_2(1 - x_1 - x_2), \end{aligned} \tag{98}$$

where x_1 and x_2 are the mole fractions of two of the species. These functional forms are typical of those used in the chemical engineering literature, and are chosen solely for illustrative purposes. We remind the reader that f is the molar Gibbs free energy with the argument reduced by one dimension.

6.1. Phase equilibrium problems. In this case, $\boldsymbol{A}$ is the identity matrix, and the analysis is relatively straightforward. The most interesting situations occur when CT sets contain more than a single point.

For the binary system in Fig. 1, the lines $t_1 - t_4$ illustrate several possible common tangent planes and their corresponding CT sets. For example, $\{x_1, x_2\}$ and $\{x_3, x_4\}$ are SCT

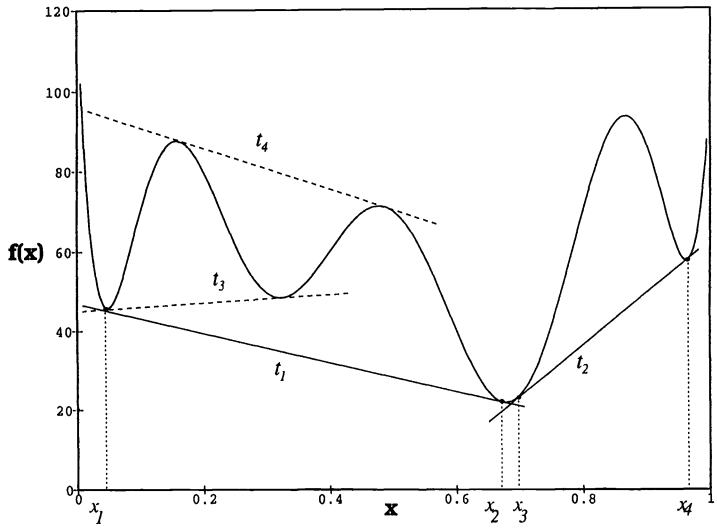


FIG. 1. *Phase equilibrium for the binary system with molar Gibbs free energy, $f(x)$, given by* (97). *x is the mole fraction of one of the species.*

sets corresponding, respectively, to t_1 and t_2. Line t_4 does not satisfy the criteria of Corollary 2 (is not an LCT set), and t_3 is an LCT set but not an SCT set.

Although t_1 and t_2 both give rise to SCT sets, the feasibility conditions provide the required additional criteria to determine which (if either) of them solves a given problem, as well as values of the remaining solution variables. The feasibility conditions (90)–(93) (recall that $\boldsymbol{c} = \boldsymbol{0}$ and $z^* = 1$) indicate that for $\boldsymbol{Y}^* = (y_1^*, \ldots, y_k^*, \boldsymbol{x}_1^{*T}, \ldots, \boldsymbol{x}_k^{*T})$ to be a global minimum, $\boldsymbol{b}$ must be a convex combination of the k points of the SCT set, $(\boldsymbol{x}_1^{*T}, \ldots, \boldsymbol{x}_k^{*T})$. In Fig. 1, if $b \in (x_1, x_2)$ or $b \in (x_3, x_4)$, then there exists, respectively, a feasible (y_1, y_2) or (y_3, y_4) that, together with the SCT set, forms the complete solution of the problem. For b in either of the above two ranges, there are said to exist two phases in the system at equilibrium, whose compositions are given by the appropriate SCT set. If b lies outside these ranges, there exists only one phase at equilibrium, with composition $x^* = b$.

Different possibilities for a ternary system are illustrated in Fig. 2. For the particular $\boldsymbol{b}$ shown in Fig. 2(a), the SCT set is $\{\boldsymbol{b}\}$, corresponding to a one-phase solution. In Fig. 2(b), the tangent plane shown touches the f surface at the two indicated lobes. As this tangent plane rotates around the lobes, the line joining the intersection of the points of common tangency moves within the triangular feasibility region $\boldsymbol{S}$ in the (x_1, x_2) plane. The solution occurs when the position of the tangent plane satisfies the feasibility requirement that $\boldsymbol{b}$ be a convex combination of the points of the SCT set, as depicted in the figure.

Figure 2(c) shows a tangent plane that touches the f surface simultaneously at three points. For any $\boldsymbol{b}$ lying in the interior of the indicated triangle, the corresponding SCT set shown gives the compositions of the three phases in equilibrium, since any such $\boldsymbol{b}$ may be expressed as a convex combination of the three vertices. The coefficients of this combination yield the amounts $\{y_1, y_2, y_3\}$ of the three phases.

Finally, in Fig. 2(d) we show a case in which the indicated plane that is tangent to the two rear lobes in the figure also intersects the lobe at the front of the figure, yielding an LCT set, but not an SCT set. This case is the analogue of line t_3 in Fig. 1.

6.2. Chemical equilibrium problems. In a system at chemical equilibrium, the situation is complicated by the fact that the tangent plane is constrained to be of the form given by (70).

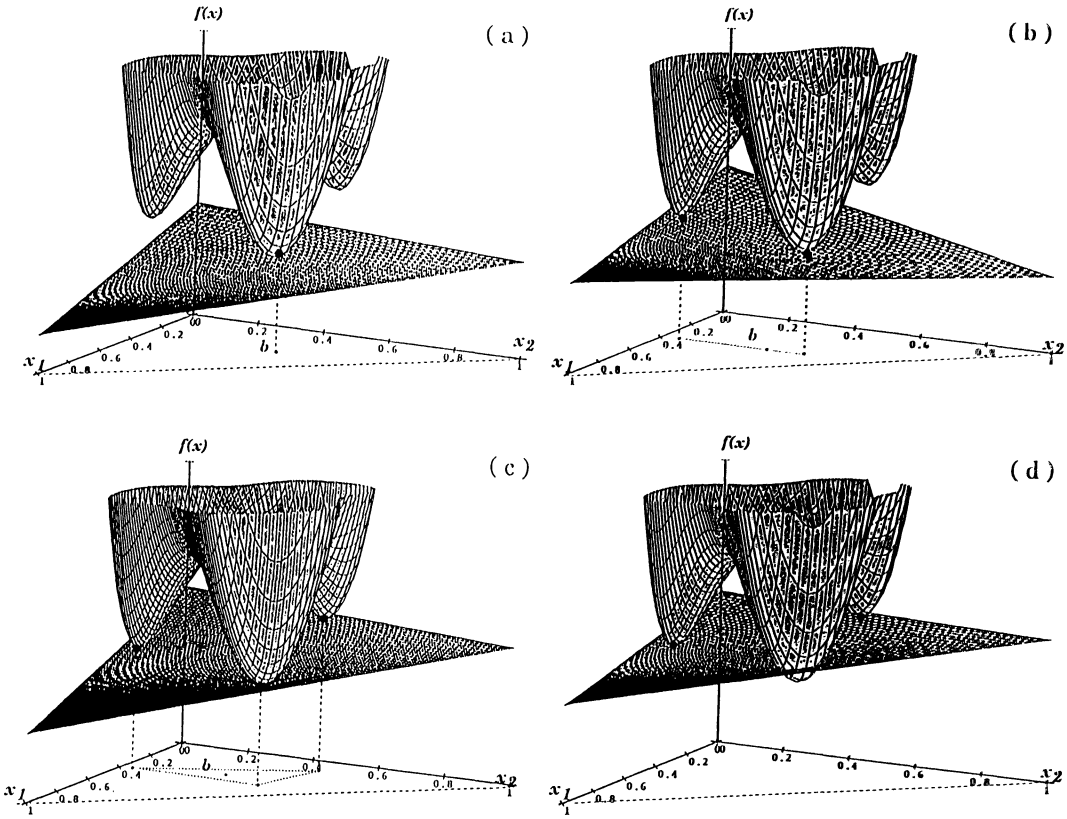


FIG. 2. *Phase equilibrium for the ternary system with molar Gibbs free energy,* $f(x_1, x_2)$, *given by* (98). x_1 *and* x_2 *are the mole fractions of two of the species.*

Unlike the case for phase equilibrium problems, this causes CT sets containing only a single point (single-phase cases) to be interesting, as well as those containing more than one point.

In the case of a binary system ($n = 1$), we may have at most $m = 1$. The general case is

$$\boldsymbol{A}' = (a_1, a_2) \tag{99}$$

giving

$$A = (a_1 - a_2) \tag{100}$$

and

$$c = (a_2). \tag{101}$$

The tangent lines to f are given by

$$\Theta(x) = \alpha[(a_1 - a_2)x + a_2]. \tag{102}$$

This is a family of lines with intercept $a_2/(a_2 - a_1)$, for any value of α. Figure 3 shows the example $a_1 = 1$, $a_2 = 2$, using the same objective function f as considered in Fig. 1 for phase equilibrium. For $m = 1$ (for any n), the feasibility conditions (90)–(93) are satisfied by *any* $\boldsymbol{x} \in \boldsymbol{S}$. Hence, CT sets are obtained by rotating the line with the fixed intercept until it becomes tangent to f. Four cases in which this can occur are shown in the figure. The three

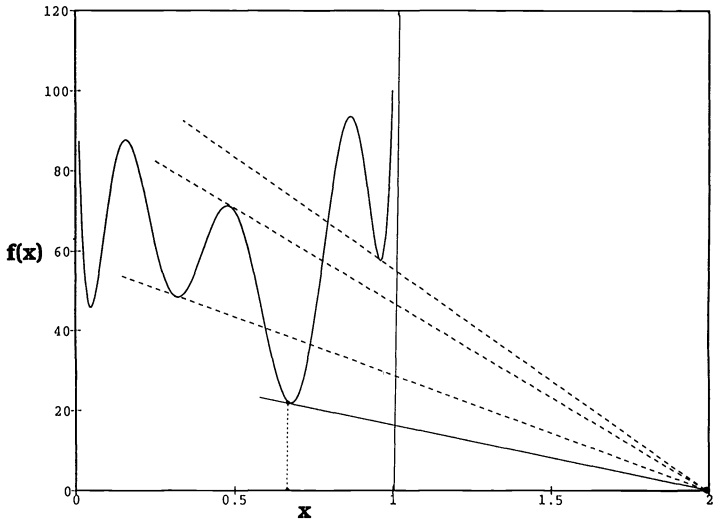


FIG. 3. *Chemical equilibrium for a single-element binary system corresponding to that of Fig.* 1, *for the formula matrix* $\boldsymbol{A} = (1, 2)^T$.

dashed tangent lines do not satisfy the reaction tangent-plane criterion (two yield an LCT set, but none yields an SCT set), and the solid tangent line represents the solution of the problem (i.e., yields an SCT set). The system has a single phase with composition given by the value of x at the point of tangency. Note that, had this tangent line been simultaneously tangent to f at two (or more) points, then the corresponding SCT set would represent the compositions of the appropriate number of different phases at equilibrium.

For a single-element ($m = 1$) ternary ($n = 2$) system, we have

$$\boldsymbol{A}' = (a_1, a_2, a_3), \tag{103}$$

$$\boldsymbol{A} = (a_1 - a_2, a_2 - a_3), \tag{104}$$

and

$$c = a_3. \tag{105}$$

The tangent planes are given by

$$\Theta(x_1, x_2) = \alpha[(a_1 - a_3)x_1 + (a_2 - a_3)x_2 + a_3]. \tag{106}$$

This is a family of planes that intersect the (x_1, x_2) plane in the fixed line ℓ given by

$$(a_1 - a_3)x_1 + (a_2 - a_3)x_2 + a_3 = 0. \tag{107}$$

Since $m = 1$, the feasibility conditions are again satisfied by any $\boldsymbol{x} \in \boldsymbol{S}$. Figure 4 shows the example $\boldsymbol{A}' = (1, 2, 3)$. CT sets are obtained by rotating the plane about the line ℓ until it becomes tangent to the surface f. The figure depicts the SCT set determining the solution. In general, the number of phases is given by the number of points in the SCT set (a one-phase solution is depicted).

The final examples consider a two-element ($m = 2$) ternary system, with formula matrix

$$\boldsymbol{A}' = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix}. \tag{108}$$

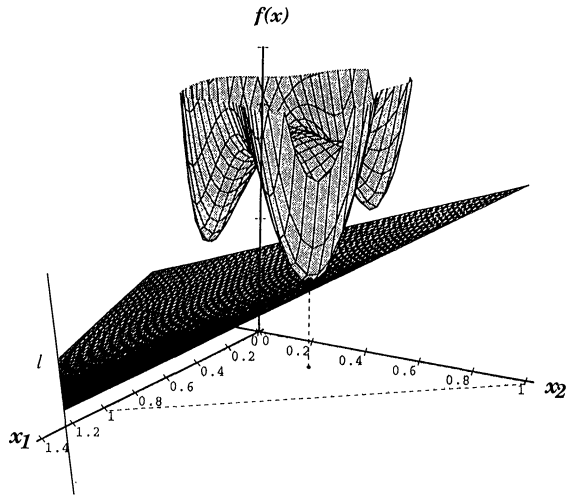


FIG. 4. *Chemical equilibrium for a single-element ternary system corresponding to that of Fig. 2, for the formula matrix* $\boldsymbol{A} = (1, 2, 3)^T$.

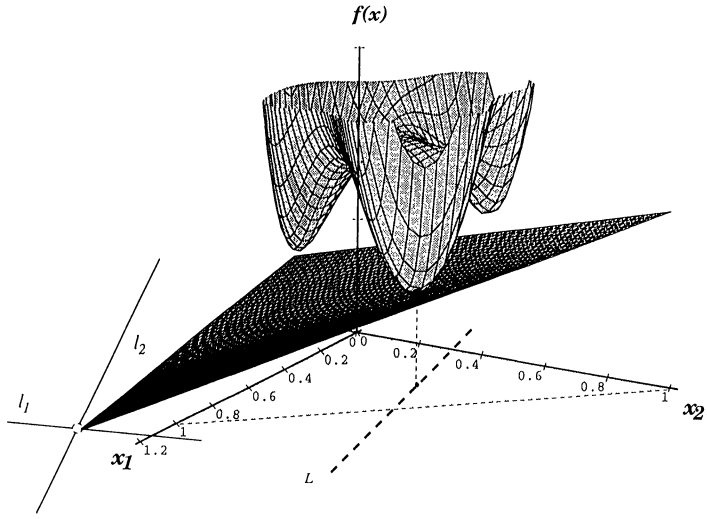


FIG. 5. *Chemical equilibrium for a two-element ternary system corresponding to that of Fig. 2, in a case when there are two phases at equilibrium.*

The tangent planes are given by

(109)
$$\Theta(x_1, x_2) = \alpha_1[(a_{11} - a_{13})x_1 + (a_{12} - a_{13})x_2 + a_{13}] + \alpha_2[(a_{21} - a_{23})x_1 + (a_{21} - a_{23})x_2 + a_{23}].$$

This is a family of planes that intersect the (x_1, x_2) plane at the common intersection point of the two lines ℓ_1 and ℓ_2 given by

(110) $$(a_{11} - a_{13})x_1 + (a_{12} - a_{13})x_2 + a_{13} = 0,$$

(111) $$(a_{21} - a_{23})x_1 + (a_{22} - a_{23})x_2 + a_{23} = 0.$$

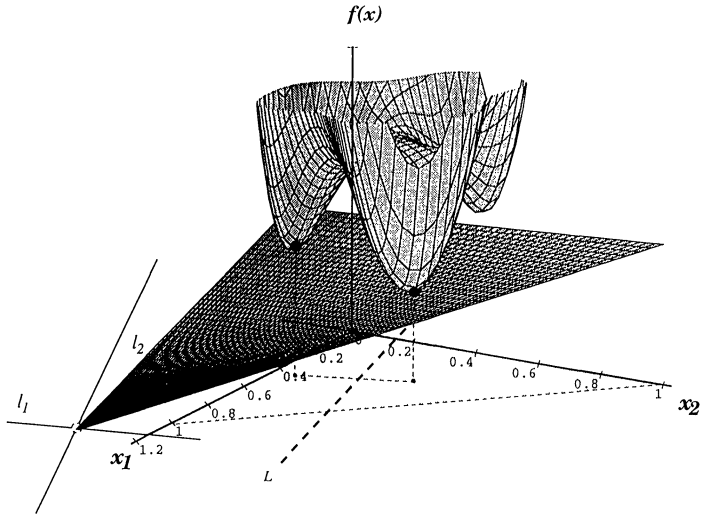


FIG. 6. *Chemical equilibrium for a two-element ternary system corresponding to that of Fig. 2, in a case when there is one phase at equilibrium.*

For a two-element system, the feasibility conditions (90)–(93) imply that, in a single-phase case, the SCT set must lie on the line L given by

(112)
$$[b_1(a_{21} - a_{23}) - b_2(a_{11} - a_{13})]x_1 + [b_1(a_{22} - a_{23}) - b_2(a_{12} - a_{13})]x_2 = b_2a_{13} - b_1a_{23}$$

in the (x_1, x_2) plane. In Fig. 5 we show an example of this case. CT sets are obtained by rotating the plane until it becomes tangent to the surface f. In the case shown in the figure, feasibility is obtained by rotating the indicated plane until the (singleton) SCT set lies on the line L.

Finally, in Fig. 6, we show an example of the case when there are two points in the SCT set. To satisfy the feasibility conditions, these points must lie on opposite sides of the indicated dashed feasibility line L.

Appendix. With reference to §2, let $\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_\pi \in E^N$ and consider

(A1)
$$G(y_1, \ldots, y_\pi, (\boldsymbol{x}'_1)^T, \ldots, (\boldsymbol{x}'_\pi)^T) = \sum_{k=1}^{\pi} y_k g(\boldsymbol{x}'_k)$$

subject to the constraints

(A2)
$$\sum_{k=1}^{\pi} y_k \boldsymbol{A}' \boldsymbol{x}'_k = \boldsymbol{b}',$$

(A3)
$$\sum_{i=1}^{N} x'_{ki} = 1,$$

(A4)
$$y_k \geq 0, \ \boldsymbol{x}'_k \geq 0, \quad (1 \leq k \leq \pi).$$

Let $g(\boldsymbol{x})$ be continuous for $\boldsymbol{x} \geq 0$ and have continuous second derivatives with respect to x_i for $\boldsymbol{x} \geq 0$, $x_i \neq 0$, $i = 1, 2, \ldots, N$, $\lim_{x_i \to 0+} \frac{\partial g}{\partial x_i} = -\infty \, (1 \leq i \leq N)$, and let

$$\boldsymbol{Z}^* = (y_1^*, \ldots, y_\pi^*, (\boldsymbol{x}_1^*)^T, \ldots, (\boldsymbol{x}_\pi^*)^T)^T \tag{A5}$$

be a local minimum for (A1) subject to (A2)–(A4). We wish to show that if $y_{k_1}^* > 0$ for some k_1, then $\boldsymbol{x}_{k_1}^* > 0$. If it were possible to construct a feasible path that led to $\boldsymbol{Z}^*$, for which $y_{k_1} > 0$ for some i, then the conditions on g would imply that an arbitrarily small movement along this path (in the direction of $x_{k_1,i}^*$ increasing) would yield a decrease in the value of G. Thus $\boldsymbol{Z}^*$ could not be a local minimum. We must therefore establish the existence of such a path, but generally this requires conditions on $\boldsymbol{A}', \boldsymbol{b}'$. In Lemma 4 we establish the existence of a feasible path under a condition on the chemical composition of the system at the local minimum. This condition is always satisfied for the phase equilibrium problem.

LEMMA 4. *Let g, $\boldsymbol{Z}^*$ be as above, and suppose that every substance in the chemical system is present in the composition determined by $\boldsymbol{Z}^*$. If $y_k^* > 0$, then $\boldsymbol{x}_k^* > 0$, $(1 \leq k \leq \pi)$.*

Proof. Suppose there exists k_1 $(1 \leq k_1 \leq \pi)$ such that $y_{k_1}^* > 0$ and $x_{k_1 i} = 0$, where $x_{k_1 i}$ is an element i of $\boldsymbol{x}_{k_1}$. Then we can find a k_2 $(1 \leq k_2 \leq \pi)$ such that $y_{k_2}^* > 0$ and $x_{k_2 i} > 0$. Otherwise, substance i is not present in the system determined by $\boldsymbol{Z}^*$, contrary to hypothesis.

Now let us construct a feasible path which approaches $\boldsymbol{Z}^*$. For $t_1 > 0$ define

$$t_2 = \frac{y_{k_1}^*}{y_{k_2}^*} t_1, \tag{A6}$$

$$y'_{k_1} = (1 + t_1) y_{k_1}^*, \tag{A7}$$

$$y'_{k_2} = (1 - t_2) y_{k_2}^*, \tag{A8}$$

$$\boldsymbol{x}'_{k_1} = \frac{1}{1 + t_1} (x_{k_1,1}^*, \ldots, x_{k_1,i-1}^*, x_{k_1,i}^* + t_1, x_{k_1,i+1}^*, \ldots, x_{k_1,N}^*)^T, \tag{A9}$$

$$\boldsymbol{x}'_{k_2} = \frac{1}{1 - t_2} (x_{k_2,1}^*, \ldots, x_{k_2,i-1}^*, x_{k_2,i}^* - t_2, x_{k_2,i+1}^*, \ldots, x_{k_2,N}^*)^T, \tag{A10}$$

$$\begin{aligned} \boldsymbol{Z}(t_1) = \big(&y_1^*, \ldots, y_{k_1-1}^*, y'_{k_1}, y_{k_1+1}^*, \ldots, y_{k_2-1}^*, y'_{k_2}, y_{k_2+1}^*, \ldots, \\ &y_K^*, \boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_{k_1}^*, \boldsymbol{x}'_{k_1}, \boldsymbol{x}_{k_1+1}^*, \ldots, \boldsymbol{x}_{k_2-1}^*, \boldsymbol{x}'_{k_2}, \ldots, \boldsymbol{x}_K^*\big) . \end{aligned} \tag{A11}$$

Clearly $\boldsymbol{Z}(0) = \boldsymbol{Z}^*$. We will verify that $\{\boldsymbol{Z}(t_1); t_1 \in [0, \varepsilon)\}$ is a feasible path for $\varepsilon(> 0)$ sufficiently small.

For $t_1 > 0$ sufficiently small, it follows from (A6)–(A10) that $y'_{k_1} > 0$, $y'_{k_2} > 0$, $\boldsymbol{x}'_{k_1} > 0$, $\boldsymbol{x}'_{k_2} > 0$. Now $\boldsymbol{Z}^*$ is feasible, so that by (A3) we have

$$\sum_{i=1}^{N} x_{ki}^* = 1 \quad (1 \leq k \leq \pi). \tag{A12}$$

From (A9), (A10), and (A12) we can now deduce

$$\sum_{i=1}^{N} x'_{k_1,i} = 1, \qquad \sum_{i=1}^{N} x'_{k_2,i} = 1. \tag{A13}$$

It follows from (A11) and (A13) that $\boldsymbol{Z}(t_1)$ satisfies (A3).

It is easy to verify (from (A7)–(A10)) that

$$y'_{k_1}\boldsymbol{A}'\boldsymbol{x}'_{k_1} + y'_{k_2}\boldsymbol{A}'\boldsymbol{x}'_{k_2} = y^*_{k_1}\boldsymbol{A}'\boldsymbol{x}^*_{k_1} + y^*_{k_2}\boldsymbol{A}'\boldsymbol{x}^*_{k_2}. \tag{A14}$$

Since $\boldsymbol{Z}^*$ is feasible it follows from (A2) that

$$\boldsymbol{b}' = \sum_{k\neq k_1,k_2} y^*_k\boldsymbol{A}\boldsymbol{x}^*_k + y^*_{k_1}\boldsymbol{A}'\boldsymbol{x}^*_{k_1} + y^*_{k_2}\boldsymbol{A}'\boldsymbol{x}^*_{k_2}. \tag{A15}$$

The equation obtained by substituting (A14) into (A15) shows that $\boldsymbol{Z}(t_1)$ satisfies condition (A2), and hence that $\boldsymbol{Z}(t_1)$ is feasible for sufficiently small $t_1 > 0$.

We now prove that, for $t_1 > 0$ sufficiently small

$$G(\boldsymbol{Z}(t_1)) - G(\boldsymbol{Z}^*) < 0. \tag{A16}$$

This contradicts the local minimality of $\boldsymbol{Z}^*$, and establishes the lemma.

$\boldsymbol{Z}^*$ and $\boldsymbol{Z}(t_1)$ differ only in the elements $y_{k_1}, y_{k_2}, \boldsymbol{x}_{k_1}, \boldsymbol{x}_{k_2}$. Consequently,

$$G(\boldsymbol{Z}(t_1)) - G(\boldsymbol{Z}^*) = y'_{k_1}g(\boldsymbol{x}'_{k_1}) + y'_{k_2}g(\boldsymbol{x}'_{k_2}) - y^*_{k_1}g(\boldsymbol{x}^*_{k_1}) - y^*_{k_2}g(\boldsymbol{x}^*_{k_2}). \tag{A17}$$

We write the right-hand side of (A17) as the sum of three terms

$$y'_{k_1}g(\boldsymbol{x}'_{k_1}) - y^*_{k_1}g\left(\boldsymbol{x}^*_{k_1} + \frac{t_1}{1+t_1}\boldsymbol{e}_i\right), \tag{A18}$$

$$y'_{k_2}g(\boldsymbol{x}'_{k_2}) - y^*_{k_2}g(\boldsymbol{x}^*_{k_2}), \tag{A19}$$

$$y^*_{k_1}g\left(\boldsymbol{x}^*_{k_1} + \frac{t_1}{1+t_1}\boldsymbol{e}_i\right) - y^*_{k_1}g(\boldsymbol{x}^*_{k_1}), \tag{A20}$$

(where $\boldsymbol{e}_i$ is the ith unit vector) and consider each in turn.

We regard $yg(\boldsymbol{x})$ as a function of $(y, \boldsymbol{x})$, and expand a Taylor series about

$$\left(y^*_{k_1}, \boldsymbol{x}^*_{k_1} + \frac{t_1}{1+t_1}\boldsymbol{e}_i\right).$$

After considerable calculation, (A18) becomes

$$t_1y^*_{k_1}g\left(\boldsymbol{x}^*_{k_1} + \frac{t_1}{1+t_1}\boldsymbol{e}_i\right) + \sum_{j\in L_1} y^*_{k_1}\frac{\partial g\left(\boldsymbol{x}_{k_1} + \frac{t_1}{1+t_1}\boldsymbol{e}_i\right)}{\partial x_j}\left(-\frac{t_1}{1+t_1}x^*_{k_1,j}\right) + o((t_1y^*_{k_1})^2) + o\left(\left(\frac{t_1}{1+t_1}x^*_{k_1,j}\right)^2\right), \tag{A21}$$

where $L_1 = \{j : x^*_{k_1,j} > 0\}$. Further calculation shows that this is

$$\leq (\pi + 1)y^*_{k_1}c_1t_1 + o(t_1^2), \tag{A22}$$

where

$$c_1 = \max_{j\in L_1, t_1\in[0,\varepsilon)}\left\{\left\| g\left(\boldsymbol{x}^*_{k_1} + \frac{t_1}{1+t_1}\boldsymbol{e}_i\right)\right\|, \left\| x^*_{k_1,j}\frac{\partial g}{\partial x_j}\left(\boldsymbol{x}^*_{k_1} + x^*_{k_1,j}\frac{t_1}{1+t_1}\boldsymbol{e}_i\right)\right\|\right\}.$$

In an analogous way, one can show that (A19) is

$$\leq (\pi + 1) y^*_{k_1} c_2 t_1 + o(t_1^2), \tag{A23}$$

where

$$c_2 = \max_{j \in L_2, j \neq i} \left\{ \|g(\boldsymbol{x}^*_{k_2})\|, \left\| x_{k_2 j} \frac{\partial g}{\partial x_j}(\boldsymbol{x}^*_{k_2}) \right\| \right\},$$

$$L_2 = \{j : x^*_{k_2 j} > 0\}.$$

Finally, (A20) can be written

$$y^*_{k_1} \left(\frac{t_1}{1+t_1} \right) \left\{ \frac{g\left(\boldsymbol{x}^*_{k_1} + \dfrac{t_1}{1+t_1} \boldsymbol{e}_i \right) - g(\boldsymbol{x}^*_{k_1})}{\dfrac{t_1}{1+t_1}} \right\}. \tag{A24}$$

Thus, in (A17) as $t_1 \to 0^+$, k_1, c_1, c_2, and $y^*_{k_1}$ are bounded but, since

$$\lim_{x_i \to 0^+} \frac{\partial g}{\partial x_i} = -\infty,$$

then

$$\left(\frac{g\left(\boldsymbol{x}^*_{k_1} + \dfrac{t_1}{1+t_1} \boldsymbol{e}_i \right) - g(\boldsymbol{x}^*_{k_1})}{\dfrac{t_1}{1+t_1}} \right)$$

is unbounded below. Thus for small enough t_1, we have

$$G(\boldsymbol{Z}(t_1)) - G(\boldsymbol{Z}^*) < 0$$

as required.

Note added in proof. Further elaboration and discussion of the geometric interpretation results of this paper are contained in Y. Jiang, G. R. Chapman, and W. R. Smith, *On the Geometry of Chemical Reaction and Phase Equilibria*, Fluid Phase Equilib., in press.

REFERENCES

[1] S. Sandler, *Chemical and Engineering Thermodynamics*, Chapter 9, J. Wiley & Sons, Inc., New York, 1989.

[2] W. R. Smith and R. W. Missen, *Chemical Reaction Equilibrium Analysis: Theory and Algorithms*, Krieger, Malabar, FL, 1991. (Reprint of same title, Wiley-Interscience, New York, 1982.)

[3] J. W. Gibbs, *The Scientific Papers of J. Willard Gibbs, Vol. 1, Thermodynamics*, Dover, New York, 1961.

[4] H. S. Caram and L. E. Scriven, *Non-unique reaction equilibria in non-ideal systems*, Chem. Engr. Sci., 31 (1976), pp. 163–168.

[5] H. G. Othmer, *Non-uniqueness of equilibria in closed reacting systems*, Chem. Engr. Sci., 31 (1976), pp. 993–1003.

[6] R. A. Heidemann, *Non-uniqueness in phase and reaction equilibrium calculations*, Chem. Engr. Sci., 33 (1978), pp. 1517–1528.

[7] R. Gautam and W. D. Seider, *Computation of phase and chemical equilibrium, Part 1. Local and constrained minima in Gibbs free energy*, Amer. Inst. Chem. Engr. J., 25 (1979), pp. 995–998.

[8] L. E. Baker, A. C. Pierce, and K. D. Luks, *Gibbs energy analysis of phase equilibria*, Soc. Petroleum Engr. J., 22 (1982), pp. 731–742.
[9] M. L. Michelsen, *The isothermal flash problem, Part* 1. *Stability*, Fluid Phase Equilib., 9 (1982), pp. 1–19.
[10] A. E. Mather, *Phase equilibria and chemical reaction*, Fluid Phase Equilib., 30 (1986), pp. 83–100.
[11] M. P. Castier, P. Rasmussen, and A. Fredenslund, *Calculation of simultaneous chemical and phase equilibria*, Chem. Engr. Sci., 44 (1989), p. 237.
[12] A. P. Gupta, P. R. Bishnoi, and N. Kalogerakis, *A method for the simultaneous phase equilibria and stability conditions for multiphase reacting and non-reacting systems*, Fluid Phase Equilib., 63 (1991), pp. 65–89.
[13] J. V. Smith, R. W. Missen, and W. R. Smith, *General optimality criteria for multiphase multireaction chemical equilibrium*, Amer. Inst. Chem. Engr. J., 39 (1993), pp. 707–710.
[14] M. S. Bazaraa and C. M. Shetty, *Nonlinear Programming, Theory and Algorithms*, John Wiley & Sons, Inc., New York, 1979.

SIAM J. OPTIMIZATION
Vol. 5, No. 4, pp. 835–857, November 1995

006

THE MOLECULE PROBLEM: EXPLOITING STRUCTURE IN GLOBAL OPTIMIZATION *

BRUCE HENDRICKSON†

Abstract. The molecule problem is that of determining the relative locations of a set of objects in Euclidean space relying only upon a sparse set of pairwise distance measurements. This NP-hard problem has applications in the determination of molecular conformation. The molecule problem can be naturally expressed as a continuous, global optimization problem, but it also has a rich combinatorial structure. This paper investigates how that structure can be exploited to simplify the optimization problem. In particular, we present a novel divide-and-conquer algorithm in which a large global optimization problem is replaced by a sequence of smaller ones. Since the cost of the optimization can grow exponentially with problem size, this approach holds the promise of a substantial improvement in performance. Our algorithmic development relies upon some recently published results in graph theory. We describe an implementation of this algorithm and report some results of its performance on a sample molecule.

Key words. global optimization, graph rigidity, molecular conformation

AMS subject classifications. 05C10, 49M27, 51K99

1. Introduction.

Consider a set of objects in Euclidean three-space at unknown locations. We wish to determine the relative locations of the objects, but the only information available to us is some subset of their pairwise distances. How can we use this information to compute their positions? We call this the *molecule problem.* It has obvious applications in surveying and satellite ranging [19], [31], and a less obvious but potentially more important application in determining molecular conformation. It is possible to interpret the nuclear magnetic resonance spectra of a molecule to obtain pairwise interatomic distance information [10], [33], [32]. Solving the molecule problem in this context would determine the three-dimensional shape of the molecule, which is critical for understanding its chemical and biological properties.

The data in an instance of the molecule problem can be succinctly represented by a graph $G = (V, E)$. The vertices V correspond to the objects or atoms, and an edge $e_{ij} \in E$ connects vertices i and j if the distance between the corresponding objects is known. We will denote the number of vertices and edges by n and m, respectively, and the distance associated with edge e_{ij} by d_{ij}. A *realization* of a graph is a mapping p that takes each vertex to a point in Euclidean space. (Some authors prefer the term *embedding*, but a realization need not be an embedding in the strict topological sense.)

The molecule problem can be naturally phrased as a nonlinear global optimization problem. Denoting the position of a vertex i as p_i, we can construct a simple cost function $F(p)$ that penalizes a realization for unsatisfied constraints. One simple such function is

$$F(p) = \sum_{e_{ij}} (|p_i - p_j|^2 - d_{ij}^2)^2, \tag{1}$$

where $|\cdot|$ denotes the Euclidean norm. This function is everywhere infinitely differentiable, and (assuming all the distances are correctly given) it has a global minimum

* Received by the editors January 28, 1992; accepted for publication (in revised form) July 27, 1994.

† Applied and Numerical Mathematics Department, Sandia National Labs, Albuquerque, New Mexico 87185–1110 (bah@cs.sandia.gov). This research was performed while the author was at Cornell University, supported by a fellowship from the Fannie and John Hertz Foundation.

of zero, attained when all the distance constraints are satisfied. In principle, $F(p)$ could be used with any global optimization technique to solve the molecule problem. Unfortunately, this naive approach is unlikely to work well in practice, due to the computational complexity of the problem. Saxe has shown that the molecule problem is strongly NP-complete in one dimension, and strongly NP-hard in higher dimensions [29], so it is unlikely that a general polynomial time algorithm exists. It can also be shown that $F(p)$ can have an exponential number of local minimizers, which makes the global optimization problem daunting.

In this paper, we describe an approach to the molecule problem that attempts to avoid having to solve a large global optimization by instead solving a sequence of smaller optimizations. Since the cost of an optimization can grow exponentially with the size of the problem, this approach holds the prospect of a substantial reduction in computational effort. To achieve this reduction, we will need to exploit some complex combinatorial structure inherent in the molecule problem, which will allow us to devise a novel divide-and-conquer algorithm. Although an important computer science paradigm, divide-and-conquer methods have not previously found many applications in optimization. The purpose of this paper is twofold: on the one hand we present a novel algorithm for a practically important optimization problem, and on the other hand we provide a case study of how divide-and-conquer ideas can be applied to optimization. It is our hope that the underlying ideas will find application to a variety of additional problems, a possibility we will reconsider in our conclusions.

A simple observation underlies our divide-and-conquer approach to the molecule problem. Within a large problem there are often subproblems that can be solved independently. If we can identify a subgraph that has many edges, it may be possible to determine the relative positions of the vertices in the subgraph by only considering the subgraph's induced edge constraints. Once this subproblem is solved, the entire subgraph can be treated as a rigid body. In three-space, a rigid body has six degrees of freedom, but considered independently each vertex has three. So by treating a set of vertices collectively the number of variables in the problem can be substantially reduced, greatly simplifying the original problem. Using this approach, the initial large optimization problem is replaced by a sequence of smaller ones.

If we are to treat a subgraph as a rigid body, we must be certain that the relative positions of the vertices in the rigid body match their relative positions in the solution to the full problem. This can only be guaranteed if the subgraph allows only a single, unique realization (modulo translations, rotations, and reflections), a property we refer to as *unique realizability.* In addition to characterizing uniquely realizable graphs, we need to be able to find subgraphs that have this property within the larger graph that represents the full problem.

This approach to the molecule problem has been implemented in a code named ABBIE. Since it decomposes a large global optimization into a sequence of smaller, more localized problems, the program is named in honor of Abbie Hoffman for his admonition to "think globally, act locally," although it is doubtful he had nonlinear optimization in mind! The structure of the ABBIE program is depicted in Fig. 1.

For the purposes of this paper we need to make two assumptions about the data, which make for an idealized problem. The first assumption is that we know edge distances with a high degree of accuracy. The second is that there is no special relationship among the locations of the vertices that generated the data for the problem. More formally, a realization is said to be *generic* if the vertex coordinates are algebraically independent over the rationals. We will assume the realization that gen-

```
1.    Find maximal uniquely realizable subgraphs
      For Each such subgraph
          If subgraph is small enough Then
2.            Position graph with global optimization
3.        Else Break into smaller pieces
              For Each piece call ABBIE
4.            Combine pieces with global optimization
      Return (realized subgraphs)
```

FIG. 1. *The structure of the* ABBIE *program.*

erated our data was generic, but this is actually a much stronger condition than we need. There is only a small set of algebraic dependencies that we need to avoid in the uniqueness analysis. However, within the space of all realizations, the set of generic realizations is dense, and the nongeneric realizations comprise a set of measure zero. These assumptions are unrealistic for true data, which can be noisy and imprecise, but they are necessary for the formal derivation of the algorithm. We believe these constraints can be relaxed somewhat in practice, as we will discuss in the conclusions.

Most previous work on the molecule problem has been performed by chemists interested in molecular conformation. Various heuristics have been developed that rely in various ways upon knowledge about chemical structures. A survey of this previous work is beyond the scope of this paper, but a good overview can be found in Chapter 6 of Crippen and Havel [10]. A more detailed description of the current work is provided in Hendrickson [14].

This paper is structured as follows. The characterization of uniquely realizable graphs is the topic of the next section. This is followed in §3 by an algorithm to identify uniquely realizable subgraphs, step 1 in Fig. 1. In §4 we describe ABBIE's technique for breaking a large, uniquely realizable graph into pieces (step 3 in Fig. 1). To finally determine coordinates, steps 2 and 4, ABBIE uses a global optimization procedure that is described in §5. Experimental results are presented in §6, followed by discussion and conclusions in §7.

2. Conditions for unique realizability. Does an instance of the molecule problem have a unique solution? Saxe has shown this problem to be as difficult as the original molecule problem: strongly NP-complete in one dimension, and strongly NP-hard in higher dimensions [29]. However, these results depend upon very special realizations in which the locations of the vertices are not algebraically independent. Since we are assuming that our problem is generic, these cases can be excluded, and the uniqueness question becomes much easier. Strong results can be derived that depend only upon the underlying graph, independent of the edge lengths.

Two independent necessary conditions for unique realizability are established in Hendrickson [15], along with algorithms for their detection, and we briefly summarize these below. Unfortunately, in three and higher dimensions these conditions aren't sufficient. We present a sufficiency condition for unique realizability in §2.3 and an algorithm for identifying it in §2.3.2.

The two independent necessary conditions derived in Hendrickson [15] for a graph to be uniquely realizable in d dimensions are

1. vertex $(d+1)$–connectivity and
2. redundant rigidity.

Of these, vertex connectivity is a well-studied graph property, and efficient al-

gorithms for verifying $(d+1)$-connectivity have been developed [1], [5], [16], [18]. Redundant rigidity is less familiar, but efficient algorithms are known [15] and reviewed below. For later reference we will review some simple rigidity theory in §§2.1 and 2.2; a more complete discussion can be found in some of the references [2], [3], [9], [28]. In §2.3 a previously unpublished sufficient condition for unique realizability is derived and an algorithm to identify it is sketched.

2.1. Graph rigidity. We will call the combination of a graph G and a realization p a *framework*, denoted by $p(G)$. A realization is *satisfying* if all the pairwise distance constraints are satisfied. Intuitively, a framework is flexible if the vertices can move while keeping all the edge distance constraints obeyed. More formally, a *finite flexing* of a framework $p(G)$ is a family of realizations of G, parameterized by t so that the location of each vertex i is a differentiable function of t and

$$\forall e_{ij} \in E, \quad |p_i(t) - p_j(t)|^2 = \textit{constant}. \tag{2}$$

Note that a motion of the Euclidean space itself, a rotation or translation, satisfies the definition of a finite flexing, and such flexings are said to be *trivial*. If the only finite flexings allowed by a framework are trivial, then the framework is said to be *rigid*. Otherwise it is *flexible*. In d-space there are $d(d+1)/2$ independent trivial flexings.

A linearized version of flexibility is more convenient, so thinking of t as time and differentiating (2) we find that

$$\forall e_{ij} \in E, \quad (v_i - v_j) \cdot (p_i - p_j) = 0, \tag{3}$$

where v_i is the instantaneous velocity of vertex i. An assignment of velocities that satisfies (3) for a particular framework is an *infinitesimal motion* of that framework. If a framework has a nontrivial infinitesimal motion it is *infinitesimally flexible*, and if not it is *infinitesimally rigid*.

Clearly the existence of a finite flexing implies an infinitesimal motion, but the converse is not always true. However, for generic realizations the converse is true [28], and, since we are considering only generic realizations, we will drop the prefix and refer to frameworks as either rigid or flexible. Whether a generic framework is rigid or flexible is purely a property of the underlying graph as indicated by the following theorem [13].

THEOREM 2.1 (Gluck). *If a graph has a single infinitesimally rigid realization, then all its generic realizations are infinitesimally rigid.*

This theorem is crucial for a graph theoretic approach to the molecule problem. Since the frameworks built from a graph are either all infinitesimally flexible or almost all rigid, graphs can be characterized according to the typical behavior of their frameworks, without reference to a specific realization. This also allows us to be somewhat cavalier in the distinction between rigid frameworks and graphs that have rigid realizations. Henceforth such graphs will be referred to as *rigid graphs*.

In one dimension, rigidity is equivalent to connectivity. In two dimensions a combinatorial characterization of rigidity was first discovered by Laman [20], and several different $O(n^2)$ algorithms for rigidity testing have been developed [11], [15], [17].

In three and higher dimensions, no combinatorial characterization of rigidity is known. However, there is an efficient randomized algorithm based on Theorem 2.1 and (3). Begin by randomly positioning all the vertices. With probability one, the rigidity of the corresponding framework will be the same as that of the graph. Now construct

the set of equations (3), where the velocities are the unknowns. The coefficients of the velocities can be formed into a matrix of size $m \times nd$, known as the *rigidity matrix*, denoted by M. Each row of M corresponds to the constraint imposed by a single edge. The null space of this matrix represents the allowed infinitesimal motions of the framework. Clearly the $d(d+1)/2$ trivial infinitesimal motions are in the null space. So if the rank of the rigidity matrix is $nd - d(d+1)/2$, then the graph is rigid, otherwise with probability one it is flexible.

2.2. Redundant rigidity. A graph is defined to be *redundantly rigid* if it is rigid after the removal of any single edge. Redundant rigidity is a necessary condition for a graph to be generically uniquely realizable [15]. We will define an edge of a rigid graph to be *redundant* if the graph remains rigid after the removal of the edge.

In one dimension, redundant rigidity is equivalent to edge biconnectivity. In two dimensions, an efficient algorithm built upon the combinatorial characterization of rigidity is described in Hendrickson [15]. In higher dimensions, since no graph theoretic characterization of rigidity is known, no characterization of redundant rigidity exists either. However, the randomized approach for rigidity testing described above can be extended to check for redundant rigidity.

Since rows of the rigidity matrix, M, correspond to edges of the graph, a framework is redundantly rigid if and only if M^T has maximal rank after the removal of any single column. A column of M^T is said to be *redundant* if the rank of M^T remains the same after its removal. If M^T has a set of $nd - d(d+1)/2$ linearly independent, redundant columns, then the framework is redundantly rigid.

Our algorithm for redundant rigidity builds upon a QR factorization of M^T. We maintain a list of linearly independent columns, and a new column is added to the list if it is linearly independent of the current set, otherwise it is discarded. A discarded column of M^T can be expressed as a linear combination of some set of the independent columns. The discarded column could replace any of the columns in the linear combination which form it, without altering the span of the independent set. Hence, a discarded column identifies a set of redundant columns within the list.

The columns within the list whose linear combination equals a discarded column can be easily determined. Assume the algorithm has identified k independent columns of M^T, placed together to form an $nd \times k$ matrix, A_k. The QR factorization has been proceeding on these columns as they are identified, so there is a $k \times k$ orthogonal matrix Q_k and an $nd \times k$ upper triangular matrix R_k satisfying $Q_k R_k = A_k$. If a new column b of M^T is linearly dependent upon the columns of A_k then there must be a vector c satisfying $A_k c = Q_k R_k c = b$ or, alternately, $R_k c = Q_k^T b$. In the course of the QR factorization the column b has been overwritten with $Q_k^T b$, so it is easy to solve the upper triangular system for c. The nonzero elements of c identify which columns of A_k contribute to the linear combination composing b, that is, which columns are redundant.

This procedure requires the solution of $O(m)$ triangular systems, each of which requires $O(k^2)$ operations, where k is always $O(nd)$, so the total additional time is of the same order as the QR factorization itself, $O(mn^2d^2)$.

2.3. A sufficient condition for unique realizability. In one dimension, the necessary conditions for generic unique realizability discussed above reduce to edge biconnectivity, which can also be shown to be sufficient. In two dimensions, we know of no examples of graphs that satisfy the necessary conditions while not being unique, but the sufficiency of these conditions has not been proven.

In three and higher dimensions, Connelly has discovered a set of bipartite graphs that satisfy the necessary conditions above, while still allowing multiple realizations [6], [8]. In three dimensions the only graph in this set is $K_{5,5}$, the complete bipartite graph with five vertices in each partition. This class of graphs was identified by the unusual properties of their *stress matrices*, an exploration of which will lead us to a sufficient condition for unique realizability.

2.3.1. The stress matrix. Consider a framework $p(G)$ consisting of a graph G and a generic satisfying realization p. A *stress* for $p(G)$ is an assignment of scalars $\omega_{ij} = \omega_{ji}$ to every pair of vertices of G in such a way that $\omega_{ij} = 0$ if $e_{ij} \notin E$, and

$$\sum_{j=1}^{n} \omega_{ij}(p_i - p_j) = 0 \quad \forall i, \tag{4}$$

where p_k is the location in $\mathbb{R}^d$ of vertex k. Note that these are vector equations since each p_k has d coordinates, so each of the d dimensions must satisfy an identical set of equations.

The concept of a stress comes originally from mechanical engineering, where the edges would be considered to be cables or struts under tension or compression. The framework will be in equilibrium exactly when the vector sum of all the stresses on each vertex is zero, which is the condition expressed by (4).

Equation (4) defines a stress for a particular realization p. In general, this same set of values ω_{ij} will not be a stress for a different realization. However, there is a very important exception to this general rule. Stresses are useful for our purposes because of the following result due to Connelly [7].

THEOREM 2.2. *Let p be a generic, satisfying realization of G in $\mathbb{R}^d$ in which the affine span of the locations of the vertices is d-dimensional. If ω is a stress for $p(G)$, then ω is a stress for any satisfying realization of G.*

This theorem allows us to greatly narrow down our search for alternate satisfying realizations. Once we generate a stress for p we only need to consider realizations q that satisfy the same stress equations.

Assume we have generated a stress for our initial satisfying realization p. We wish to find a q that can replace p in (4). It will be convenient to rewrite the stress equations. Let q_i^r denote coordinate r of the location of vertex i in realization q. For each $1 \leq i \leq n$ and each $1 \leq r \leq d$ we have the following equation.

$$\left(\sum_{j=1}^{n} \omega_{ij}\right) q_i^r - \sum_{j=1}^{n} \omega_{ij} q_j^r = 0. \tag{5}$$

This is just a set of n linear equations repeated for each of the d dimensions. Define the symmetric, $n \times n$ *stress matrix*, Ω, as follows.

$$\Omega_{i,j} = \begin{cases} -\omega_{ij} & \text{if } i \neq j, \\ \sum_k \omega_{ik} & \text{if } i = j. \end{cases}$$

If we denote the n-vector consisting of coordinate r of each vertex by q^r, then (5) can be succinctly expressed as

$$\Omega q^r = 0, \tag{6}$$

for each dimension r. Any satisfying realization must satisfy these equations, so our search for alternate satisfying realizations is now reduced to an investigation of the null space of Ω.

Each row of the stress matrix sums to zero, so the vector of ones is in Ω's null space. The product Ωp^r is identically zero by the construction of the stress. This is true for each of the d coordinates, so the nullity of the stress matrix is at least $d+1$. The linear combinations of these trivial null vectors are the affine linear maps of the vertices in realization p. That is, any realization in which q_i, the coordinates of vertex i, can be expressed as $Ap_i + b$ will satisfy the same stress equations as p, where A is any $d \times d$ matrix and b any d-vector. If there is nothing else in the null space of Ω then the only possible alternate satisfying realizations are these affine linear maps, which gives us the following theorem.

THEOREM 2.3. *Let p be a generic, satisfying realization of G in $\mathbb{R}^d$ in which the affine span of the locations of the vertices is d-dimensional. If ω is a stress for $p(G)$ such that Ω has nullity $d+1$, then any satisfying realization of G must be an affine linear map of p.*

Connelly has shown that these troublesome affine linear maps cannot lead to nonequivalent, satisfying realizations [7]. This gives us the following sufficient condition for a graph to have a unique realization.

THEOREM 2.4. *Let p be a generic, satisfying realization of G in $\mathbb{R}^d$ in which the affine span of the locations of the vertices is d-dimensional. If ω is a stress for $p(G)$ such that Ω has nullity $d+1$, then there is no nonequivalent, satisfying realization of G.*

Determining whether the stress matrix has the proper nullity is what we will call the *stress test* for unique realizability.

For a given realization, the stresses defined by (4) are solutions to a linear system of equations. As such they can be expressed as polynomials in the coordinates of the vertices. To determine whether or not the stress matrix has nullity 4, simply sum the squares of all the $(n-4) \times (n-4)$ subdeterminants of Ω. This polynomial will be zero if and only if the nullity of the stress matrix is greater than four. Thus we have a polynomial in terms of the coordinates of the vertices that describes our sufficiency condition. If this polynomial is nonzero for any generic realization, then it is nonzero for all generic realizations.

THEOREM 2.5. *The nullity of the stress matrix is a generic property; that is, it has the same value for for all generic realizations.*

COROLLARY 2.6. *If any generic realization passes the stress test, then all generic realizations will pass.*

In other words, the stress test is generic. Our necessary conditions were generic as well, which provides evidence that unique realizability may itself be a generic property. Whether or not this is the case is an open problem. Corollary 2.6 justifies using a random realization to generate the stresses. As we will see in the next section, a particularly convenient realization to use is the one that was utilized to generate the rigidity matrix for our redundant rigidity algorithm.

2.3.2. Forming the stress matrix. The sufficient condition for unique realizability expressed by Theorem 2.4 is not much use for us unless we can readily compute stresses. Fortunately, this is not a problem. In fact, most of the work has already been done in the QR factorization of the rigidity matrix M that was described in §2.2. Redundant edges of the graph were identified by linear dependence among the columns of M. Element $[e(i,j), di+r]$ of M is $p_i^r - p_j^r$ if the edge numbered $e(i,j)$

connects vertices i and j, and zero otherwise. Consequently, if the multipliers in a linear combination of columns of M summing to zero are denoted by $\alpha_{e(i,j)}$ for edge $e(i,j)$, then for each $1 \leq i \leq n$ and $1 \leq r \leq d$

$$\begin{aligned} 0 &= \sum_e \alpha_{e(i,j)} M_{e(i,j),di+r} \\ &= \sum_j \alpha_{e(i,j)} (p_i^r - p_j^r). \end{aligned}$$

Equating $\alpha_{e(i,j)}$ with ω_{ij} in (4) we see that the multipliers in the linear combination constitute a stress. Consequently, the solution vector to the triangular systems solved in §2.2 identifies a stress.

In the course of a full redundant rigidity calculation many stresses may be found, one for every discarded row. Each of these stresses generates its own stress matrix, and any linear combination of stresses is also a stress. Since we are interested in identifying a stress that maximizes the rank of Ω, almost any linear combination of the stresses generated in the QR factorization will suffice. In practice we use a sum of all the stresses, scaled by random multipliers.

The determination of the rank of the stress matrix can be troublesome due to numerical roundoff problems. The entries in the stress matrix are the result of a previous factorization, so they may already have modest inaccuracies. For this reason it is important to determine the rank of Ω in as numerically stable a fashion as possible, so we recommend a singular value decomposition.

3. Finding uniquely realizable subgraphs. The preceding section described two necessary conditions and a sufficient condition for a graph to have a unique realization. Step 1 of the algorithm sketched in Fig. 1 requires a further step, the identification of subgraphs that are uniquely realizable. Ideally, we would like to find subgraphs that satisfy the sufficiency condition, but it is not clear how the stress test can be used directly for this purpose. However, the necessary conditions are well suited for identifying subgraphs, which suggests using the necessary conditions to find candidate subgraphs, and then confirming their uniqueness with the sufficiency test. An outline of such an algorithm is presented in Fig. 2.

If Graph is $K_{5,5}$ **Then**
    Return (No_unique_subgraphs)
Else If not four–connected **Then**
    Recurse on four–connected components
Else If not redundantly rigid **Then**
    recurse on redundantly rigid components
Else Perform sufficiency test
    If Pass **Then**
        Return (Graph_unique)
    Else Output interesting graph

FIG. 2. ABBIE's *algorithm for finding maximal uniquely realizable subgraphs.*

The only case that is not handled with this approach is a graph that passes the necessary conditions and fails the sufficiency test. We have yet to discover such a graph, although we would be very interested in finding one. In practice this approach seems to work very well, at least on the test problems that will be described in §6.

Incidentally, our failure to find any graphs that pass the necessary conditions while failing the sufficiency test provides evidence that such graphs are uncommon, if they exist at all.

The heart of the procedure described in Fig. 2 is finding $(d+1)$-vertex connected subgraphs and redundantly rigid subgraphs. The vertex connectivity problem is well studied, and good algorithms for finding maximal subgraphs are known [1], [5], [16], [18]. However, algorithms for finding redundantly rigid subgraphs have not been previously considered. In one dimension, this requires finding biconnected components, for which there are $O(m)$ algorithms [1]. In two dimensions, an $O(n^2)$ algorithm for finding maximal redundantly rigid components is given by Hendrickson in [15]. In three and higher dimensions, an algorithm needs to rely upon the QR factorization of the transpose of the rigidity matrix. ABBIE's algorithm, summarized in Fig. 3, relies upon the observation that any edge that is not redundant in the original graph will not be redundant in any subgraph. After removing these nonredundant edges, the flexes that remain will not affect the redundantly rigid subgraphs. These subgraphs can be identified by noticing which subsets of vertices preserve their relative locations under the remaining flexes, which requires the construction of a basis for the remaining flexes.

A basis for the space of flexes can be generated by the QR factorization of the columns of the transpose of the rigidity matrix that corresponds to an independent set of redundant edges. It is helpful to exclude the trivial motions from the basis by explicitly adding them as columns at the end of the factorization. This reduces the size of the space of flexes and so speeds up the determination of subgraphs. If there are k redundant, independent columns, then the final $3n-6-k$ columns of Q form a basis for the flexes. Sets of vertices whose relative positions remain unchanged under these flexes are redundantly rigid subgraphs.

Identifying these sets of vertices requires the ability to determine whether the distance between any two vertices i and j changes under any of the allowed flexes. For each flex this involves the calculation of the inner product $(v_i - v_j) \cdot (p_i - p_j)$, where v_i is the velocity vector of vertex i under the flex, and p_i is its location. If this quantity is zero then the distance between i and j remains unchanged.

A pair of vertices whose distance doesn't change under any of the allowed flexes could just as well be connected by an edge, so we will consider such vertices to be joined by an *induced edge*. Finding sets of relatively fixed vertices corresponds to finding cliques in this graph of induced edges. A simple geometric observation simplifies this task. Let $\mathcal{S}$ be a set of at least three vertices whose relative positions don't vary. To determine whether a new vertex v should be added to $\mathcal{S}$ it is sufficient to check the change in the distance from v to any three vertices in $\mathcal{S}$. With three neighbors at fixed locations the position of v cannot vary continuously.

ABBIE's algorithm for identifying these cliques begins by looking for sets of three vertices whose relative locations are fixed. This requires $O(n^3)$ time. Once such a triangle is found, the unique clique containing it can be grown to maximal size by checking all other vertices against these three in $O(n)$ time. Although the resulting $O(n^4)$ algorithm is asymptotically the most expensive portion of the decomposition, for the problems discussed in §6, the cost of the entire component finding process is less than 1% of the cost of the QR factorizations.

4. Breaking large graphs. A maximal uniquely realizable subgraph may be large, and consequently prohibitively expensive to try to realize directly. As described in Fig. 1, ABBIE breaks such a subgraph into pieces and recurses on the pieces, before

Use QR factorization to identify independent set of redundant edges
Use QR factorization to construct basis for remaining flexes
For All independent three–cliques (x, y, z) in induced graph
    For All other vertices v
        If v has induced edges to x, y and z **Then**
            add v to subgraph containing x, y and z.

FIG. 3. *An algorithm for finding redundantly rigid subgraphs.*

trying to fit them back together. Ideally, smaller uniquely realizable subgraphs would be found directly, but we don't know how to do this. Instead, as indicated by step 3 of Fig. 1, ABBIE breaks the graph by finding a small vertex separator and recurses on the induced pieces.

In selecting a value for how large a subgraph must be before being broken, a balance must be struck between two extremes. A small value results in a large number of small optimization problems, and potentially difficult optimizations fitting many small pieces together. On the other hand, a large value leads to a smaller number of large optimizations. For the calculations described in §6 a cutoff of 15 vertices was used. The value of this parameter seemed to have a very small impact on overall computation time. The most expensive optimization problems were typically those that occurred higher up in the chain of recursion, involving many more vertices. Decisions about how to handle these relatively small components were not of critical importance.

The fundamental unit of information in the molecule problem is an edge length, so when a graph is broken into pieces, any edge that does not lie in a single component is lost to the recursive positioning procedures. For this reason we would like a decomposition technique that ensures that any two vertices joined by an edge end up in the same component. We would also like the technique to divide the graph into approximately equally sized pieces as this will speed the recursive decomposition. To accomplish these goals, ABBIE was endowed with a procedure to find a small vertex separator, and when the separator set is added to each component no edges are lost.

Forces between atoms are strongly repulsive at small distances, so each atom effectively has an exclusion zone in which no other atoms are located. In addition, distances can only be measured if two atoms aren't too far apart. These geometric constraints place the underlying graphs in the class of k-overlap graphs, which are known to have vertex separators of size $O(n^{2/3})$ [23]. For the problems described in §6, the separators found were uniformly small.

There are a number of different heuristics for finding small vertex separators, and for no compelling reason, ABBIE uses an algorithm described by Liu [22]. This algorithm uses a minimum degree ordering to generate an initial separator, which is then improved by a bipartite matching technique.

5. The optimization routines in ABBIE. Any program to solve the molecule problem must eventually assign coordinates to vertices. The combinatorial preprocessing described above merely delays this eventuality so that the actual positioning problems are smaller. The positioning problems that ABBIE needs to contend with involve fitting together two types of objects so that a set of distance constraints are satisfied: vertices, and subsets of vertices whose relative positions have already been determined which we will call *chunks*. These chunks can be treated as rigid bodies with at most six degrees of freedom in three-space.

ABBIE solves these problems using a three-phase approach: variable reduction, variable selection, and global optimization. In the first phase the program uses a combinatorial analysis to try to merge chunks and vertices together. For instance, if a vertex has four edges connecting it to a chunk then the vertex will generally have a unique location relative to the chunk, so the vertex can be merged into the chunk. This phase reduces the size of the resulting optimization problems and will be described in greater detail in §5.1.

After exhausting its bag of combinatorial tricks ABBIE resorts to a nonlinear, global optimization. All the possible locations and orientations of the vertices and chunks are expressed by a set of translational and rotational variables. Several different sets of variables are possible and ABBIE attempts to select a set that will minimize the cost of the optimization. This selection process is described in §5.2. ABBIE constructs a cost function that penalizes a realization for not satisfying an edge length constraint, so that the sum of the penalties will be zero only when all the constraints are satisfied. To find a realization where this function goes to zero, ABBIE generates random values for all the variables and uses them as a starting vector for a local minimization. This process is repeated until a zero value is found, indicating that all the constraints are satisfied, and that the locations of the vertices constitute a satisfying realization. Details of this nonlinear optimization will be given in §5.3.

Much more sophisticated techniques to solve this global optimization are possible. In fact, most previous approaches to the molecule problem have focused exclusively on this aspect, as discussed in Chapter 6 of Crippen and Havel [10]. Our goal in this work was to test the feasibility of the divide-and-conquer approach to the molecule problem, and it is our expectation that the overall approach will be successful, even though the component optimization techniques are quite simplistic. Better global optimization methods like tunneling [21] or efficient stochastic algorithms [27] could transparently replace the routines described in §5.3.

5.1. Combinatorial positioning techniques. To reduce the number of variables in the global optimization, ABBIE first tries to combine small numbers of chunks and vertices into larger chunks. ABBIE has five different heuristics for enlarging chunks, which are synopsized in Fig. 4. The success of these techniques depends upon specific sets of vertices not being coplanar, which is ensured by the assumption that the final solution is generic. In the first technique, if two chunks have at least four vertices in common then they can only be combined in one way, and ABBIE merges them. Second, if a vertex has four edges incident to a chunk then the vertex can be uniquely positioned relative to the chunk, and the chunk enlarged. ABBIE can use direct edge lengths that were given in the data, or induced lengths generated by a chunk that contains the two vertices.

The remaining three heuristics start with a *base chunk* and add pairs of objects to it. Consider a vertex with three direct or induced edges to the base chunk. We will call such a vertex *three-valent* to the base chunk. The location of the vertex relative to the chunk has only two possibilities, distinguished by a reflection of the vertex through the plane of its neighbors. If this ambiguity can be resolved then the vertex can be added to the chunk. A similar result applies to a chunk that shares three vertices with the base chunk. Such a chunk will also be called *three-valent* to the base chunk. The last three heuristics for enlarging chunks make use of this observation. The third technique allows two three-valent vertices to be added to a chunk if there is a direct or induced edge between the two vertices. The length of this edge is used to resolve the ambiguity of the reflections of the vertices. Note that this technique does

not work if the two vertices have the same three neighbors in the base chunk.

The fourth heuristic adds two three-valent chunks to the base chunk. There are several ways in which the reflection ambiguity can be resolved. The two chunks can share a vertex that is not in the base chunk, or there can be a direct or induced edge between vertices in the two chunks that does not involve a vertex in the base chunk. Again, if the two sets of three shared vertices are the same then the ambiguity cannot be resolved.

The last technique involves adding a three-valent chunk and a three-valent vertex to the base chunk by resolving the reflections with a direct or induced edge between the vertex and the chunk. Again, if the three adjacent or shared vertices are the same then the ambiguity cannot be resolved.

Until No Change
  For All Chunks X
    For All chunks Y, 4-valent to X
      Merge X and Y

    For All vertices v, 4-valent to X
      Merge v into X

    For All pairs of vertices v and w, 3-valent to X
      If valencies differ **And** reflections can be disambiguated **Then**
        Merge v and w into X

    For All pairs of chunks Y and Z, 3-valent to X
      If valencies differ **And** reflections can be disambiguated **Then**
        Merge X, Y and Z

    For All chunks Y and vertices v, 3-valent to X
      If valencies differ **And** reflections can be disambiguated **Then**
        Merge X, Y and v

FIG. 4. ABBIE's *combinatorial positioning heuristics.*

ABBIE applies these techniques to all combinations of chunks and vertices until no more merging is possible. The vast majority of positioning problems encountered in the test problems were resolved this way, without any need for the nonlinear optimizer. Additionally, more complicated heuristics are possible, and would probably be worth implementing. As the numerical results in §6 will show, the nonlinear optimizations dominate the execution time of ABBIE. Hence, it is our expectation that the additional cost of more complex techniques would be more than compensated for by the reduction in size and, consequently, cost of the optimization problems.

5.2. Selecting optimization variables. The optimization problems ABBIE must solve involve sets of chunks and vertices. Vertices have three translational degrees of freedom, and the location and orientation of a chunk can be described by three translational and three rotational variables. To describe rotations of chunks, ABBIE defines an axis of rotation using a standard (θ, ϕ) system, and an amount of rotation. This representation has fewer singularities than the more familiar Euler angles.

There are many possible ways to parameterize the motions of the vertices and chunks. For instance, any of the chunks can be held fixed while the others are allowed to move. The selection of optimization variables in ABBIE was designed to minimize the computational effort required by the global optimization. ABBIE solves the global optimization problem by a sequence of local minimizations, so there are two factors which determine the total cost of the global optimization: the cost of each local minimization and the number of local minimizations required to find the global optimum. We need to analyze these two quantities before we can explain the procedure for selecting optimization variables.

To find a local minimizer from a random starting point, ABBIE uses a trust region approach, repeatedly forming and factoring the Hessian matrix. This factorization tends to dominate the cost of each iteration, requiring $\Theta(q^3)$ floating point operations, where q is the number of variables. It is difficult to estimate the number of iterations each local minimization will require, as this depends in a complicated way on the local topography of the penalty function, so ABBIE assumes that the cost of each local minimization is simply proportional to the cube of the number of variables.

The number of local minimizations required to find the global optimum depends on the size of the region of attraction of the global minimizer relative to the size of the entire domain. Assuming this region of attraction is of average size, the number of local minimizations will be proportional to the number of local minimizers in the problem. Since this number can grow exponentially with the number of vertices, the number of local minimizers can be approximated as $2^{q/\beta}$, where β is an empirical parameter. After some experimentation, ABBIE was given a value of 8 for β for the test problems described in §6, but an appropriate value for this parameter depends on the class of problems under consideration.

There is one additional factor to consider in estimating the cost of an optimization. Note that edge lengths remain unchanged if a chunk is replaced by its mirror image, but there is no continuous rigid body motion to transform between these two realizations. ABBIE cannot know in advance which of these two *parities* is the correct one, so it has to try them both. Since only one will fit properly with the remainder of the graph, if a particular chunk is assigned an arbitrary parity then all the others must be made consistent. Since parities are selected randomly as a local optimization is started, this adds an additional factor of 2^{k-1} to the number of local minimization attempts, where k is the number of chunks in the optimization problem. In practice, there may be additional information available to the chemist that can determine the proper parities. If so, exploiting this knowledge should greatly improve performance, but the current incarnation of ABBIE assumes that only pairwise distances are available.

Combining these three factors, we can approximate the total cost of a global optimization as $\Theta(q^3 2^{k-1} 2^{q/\beta})$. ABBIE tries to select a set of optimization variables that minimizes this estimated cost function.

In performing a local optimization, one *base* chunk is held fixed at the origin to remove translational and rotational ambiguities. (If no chunks can be found, a single edge is used, and an additional vertex is constrained to lie in the x–y plane.) ABBIE tries all of the chunks in turn as candidate base chunks, and selects the one that minimizes the estimated cost of the optimization. If a second chunk shares three vertices with the base chunk then it has no continuous degrees of freedom. If a chunk shares two vertices with the base chunk its motions can be described with a single rotational variable. If a chunk has a single vertex in common with the base chunk

then it has three degrees of freedom, and if it shares none it has six. All chunks add a factor of two to the parity consideration.

In evaluating the candidate base chunks ABBIE adds the remaining chunks in a greedy manner, trying to minimize the estimated optimization cost. This process is sketched in Fig. 5. At each step ABBIE selects the remaining chunk with the largest number of vertices that are not yet contained in an accepted chunk. This chunk is accepted and the variables describing its motion included in the optimization, or rejected and ignored, depending upon which action reduces the estimated optimization cost. If the chunk is accepted it increases the number of chunks by one, and it can increase the number of variables in the optimization. If it is rejected, its vertices that are not yet in an accepted chunk are assumed to wander freely, each adding three to the number of variables. ABBIE processes all of the remaining chunks in this way, determining the cost of selecting this base chunk, as well as generating a set of variables for the optimization. ABBIE selects the base chunk that generates the lowest estimated optimization cost, and the corresponding variables are used in the global optimization.

```
Best_Cost := ∞
For All vertices v, free(v):= TRUE
For All Candidate base chunks X
    k := 1
    q := 0
    For All vertices v in X, free(v) := FALSE
    While any chunks remain
        Select next chunk Y having maximal free vertices
        t := Number of free vertices in Y
        If Y 3-valent to X Then r := 0
        Else If Y 2-valent to X Then r := 1
        Else If Y 1-valent to X Then r := 3
        Else If Y 0-valent to X Then r := 6
        Accept_Cost := (q + r)^3 2^k 2^((q+r)/β)
        Reject_Cost := (q + 3t)^3 2^(k-1) 2^((q+3t)/β)
        If (Accept_Cost < Reject_Cost) Then
            q := q + r
            k := k + 1
            For All vertices v in Y, free(v) := FALSE
        Else discard Y
    End While
    t := Number of remaining free vertices
    q := q + 3t

    Cost := q^3 2^(k-1) 2^(q/β)
    If (Cost < Best_Cost) Then
        Best_Cost := Cost
        Best_Base_Chunk := X
        Optimization_Variables := those identified above
```

FIG. 5. ABBIE's *process for selecting optimization variables.*

5.3. ABBIE's global optimization technique. To finally position the set of chunks and vertices ABBIE finds the global minimizer of a function that penalizes a realization for violating constraints. Many different penalty functions are possible and ABBIE uses a particularly simple one. For each edge e_{ij} that needs to be satisfied the program computes a value $P_{ij}^1 = (|p_i - p_j|^2 - d_{ij}^2)^2$, where p_i is the location of vertex i, and d_{ij} is the desired distance between vertices i and j. This is the simplest possible function that has continuous derivatives of all orders and is greater than zero whenever a constraint is violated. The full penalty function for edges is then $F_1 = \sum P_{ij}^1$, where the sum is taken over all edges in the graph which aren't contained in any chunks.

Positioning problems in ABBIE can involve multiple chunks, other than the base chunk, that share one or more vertices. These vertices must be forced to coincide in a satisfying realization, so ABBIE needs a penalty term to enforce this constraint. The obvious candidate would have the same functional form as that for edges but with a zero distance. Unfortunately, this function has a singular Hessian. For this reason the program uses a simpler penalty $P_{ij}^2 = |p_i - p_j|^2$. Summing all of these types of constraints together gives F_2, a second component to the cost function. The full penalty function is then $F = F_1 + F_2$. We note that the full penalty function is composed of both quartic and quadratic functions. For large deviations from satisfiability the quartic terms should dominate the quadratic ones, while near the solution the opposite should occur. In practice this seemed to cause no problems.

To find a zero of this penalty function ABBIE generates a random starting value for each of the optimization variables, including random parities for each chunk. The program then performs a local minimization, and this process is repeated until a functional value of zero (or almost zero) is found or until a limit on the number of trials is exceeded. This is an extremely simple global optimizer and much more sophisticated techniques could easily be used instead.

For local optimization ABBIE uses a modified version of the NTRUST code of Moré and Sorensen that is based upon the trust region method described in Moré and Sorensen [24]. This approach was selected for ABBIE because trust region techniques tend to be robust, and our function and its derivatives are fairly inexpensive to evaluate explicitly. NTRUST treats the Hessian as dense, which can be inefficient, but this is not a serious problem for ABBIE since the divide-and-conquer approach avoids large problems. The ability to scale variables was added to NTRUST to cope with the different ranges associated with translational and rotational variables.

6. Results. ABBIE has been tested on simulated molecular data provided by Palmer [25], [26]. The input data consisted of simulated distance constraints, corresponding to measurements that could be made in a typical NMR experiment. However, in our case the distances were given precisely, whereas true experimental data has limited precision. The molecule that generated our test problems was bovine pancreatic ribonuclease A, a typical small protein consisting of 124 amino acids and, after discarding end chains, 1849 atoms. The three-dimensional conformation of ribonuclease is known, so all pairwise distances could be determined. For our purposes, the data set consisted of all distances between pairs of atoms in the same amino acid, along with 1167 additional distances corresponding to pairs of hydrogen atoms that were within 3.5 Å of each other. The former set of values can be deduced from the chemical structure, and the latter could in principle be measured by two-dimensional NMR spectroscopy experiments. Combined, this made for a total of 15,046 edges.

Proteins are constructed of chains of amino acids. Since the shapes of the amino acids are well known, the conformation of the protein is determined by the angular

parameters where the amino acids are joined. In fact, one common approach to the analysis of protein conformation is to treat these angles as the only degrees of freedom [12]. Under this assumption, if the locations of any four noncoplanar atoms in an amino acid can be determined, the locations of the remaining atoms in that amino acid can be easily computed. This allowed us to reduce the size of the graphs that were passed to ABBIE. Within each amino acid, we discarded vertices that had no edges to vertices outside of that amino acid, until there were only four vertices left. In addition, any amino acid that had six or fewer edges to other amino acids could not be uniquely positioned. These amino acids were removed, further reducing the size of the graph.

A single problem would give only limited insight into the strengths and weakness of ABBIE, so we generated a set of related test problems of varying sizes by extracting leading subchains of amino acids from the ribonuclease. The six different problem sizes used are presented in Table 1. The second column presents the number of vertices and edges in the initial, unadulterated graphs. These graphs were reduced in size by exploiting protein structure as discussed above, resulting in the graph sizes described in the third column. These are the graphs that were passed to the unique realizability algorithms. The final column presents the size of the largest uniquely realizable subgraph that was found within each of the reduced graphs.

TABLE 1
Sizes of the test problems; vertices (edges).

Amino acids	Initial graph	Reduced graph	Largest unique subgraph
20	292 (2263)	63 (236)	57 (218)
40	604 (4902)	186 (828)	174 (786)
60	902 (7264)	310 (1392)	287 (1319)
80	1193 (9556)	405 (1804)	377 (1719)
100	1491 (12038)	504 (2272)	472 (2169)
124	1849 (15046)	698 (3292)	695 (3283)

It is worth noting that the edge density for the full molecule is greater than that for any of the leading subchains. This is a consequence of the tendency of proteins to form compact structures. Leading subchains need not be as compact, and so the number of pairs of atoms that are close together is reduced.

For the runs described below, subgraphs were considered too big to directly realize if they contained more than 15 vertices. All larger subgraphs were divided into pieces using the small vertex separator heuristic from §4. Also, the stress test to verify unique realizability was turned off. Besides the intrinsic reduction in effort, this allowed for some economy in the redundant rigidity calculation [14]. If a subgraph passed the necessary tests, but wasn't truly uniquely realizable, disabling the stress test could lead to incorrect coordinates being computed for the subgraph. However, this would be detected when the subgraph would be used in later optimizations since it would be unable to fit properly with the remainder of the full graph.

6.1. Performance of the unique realizability algorithms. ABBIE's algorithm for finding uniquely realizable subgraphs consists of alternate phases of a four-connected components routine and a redundant rigidity code. The redundant rigidity algorithm requires a QR factorization as described in §2.2. The four-connectivity algorithm in ABBIE removes two vertices at a time and checks for biconnectivity, requiring $\Theta(mn^2)$ time. Although there are asymptotically more efficient algorithms

for this step [16], [18], the QR factorization requires $O(n^3)$ time, so a more complex algorithm for four-connectivity was deemed unnecessary. The total time spent in these portions of the code as a function of the reduced graph sizes is presented in Table 2 for the six different problems. These times are all a small fraction of the optimization time. These and all subsequent timings are for CPU time on a Sparcstation 1^+. As expected, both the four-connectivity and the redundant rigidity times grow roughly as the cube of the number of vertices.

TABLE 2
Total minutes spent in unique realizability routines.

Amino acids	Redundant rigidity	Four-connectivity
20	0.16	0.11
40	4.22	5.36
60	18.82	48.78
80	45.96	57.84
100	84.47	115.04
124	333.09	323.98

Whereas the four-connectivity routines are entirely deterministic, there is a degree of randomness involved in the redundant rigidity calculations. The values in the rigidity matrix come from a random realization of the graph. For some realizations this can lead to numerical problems in the QR factorization. This was observed in practice for the largest problem, involving the factorization of a 1085×3283 matrix. In particular, the factorization had a difficult time determining when a value should be considered to be zero. After several attempts with different random number seeds a realization was generated with excellent numerical properties. For this factorization there was a gap of nearly five orders of magnitude between the smallest value that was accepted as nonzero, and the largest that was rejected. Additional runs demonstrated that as long as there was a reasonable gap between these values the redundant rigidity calculations were essentially deterministic.

6.2. The vertex separator heuristic. The algorithm for identifying small separators ran rapidly and produced good separators. For the largest problem the total time spent in the separator routines was only 1.55 minutes, a minuscule fraction of the total running time. A plot of the size of the separator set versus the size of the graph is shown in Fig. 6 for all the invocations of the algorithm in the set of test problems. Except for the smallest graphs, the vast majority of separators have between 5–10% of the total number of vertices. Note that no separator smaller than four could ever be found, for it would imply that the graph was not four-connected, and hence not uniquely realizable.

The idea of using a small separator heuristic was based on our hope that it would typically divide the graph into two halves, each of which had a good chance of being uniquely realizable. The technique succeeded in dividing the graphs into two pieces of approximately equal size, but unfortunately they were not always uniquely realizable. Often each half would contain a large uniquely realizable subgraph along with a few smaller unique subgraphs and maybe some isolated vertices. These various pieces must eventually be combined with an invocation of the global optimizer, and the cost of an optimization depends critically on the number of subgraphs and isolated vertices being combined. When this number is large the optimization problems are difficult. As the results in the next section indicate, the total cost of each of the problems was dominated by the cost of a few large optimization problems generated in this way. In

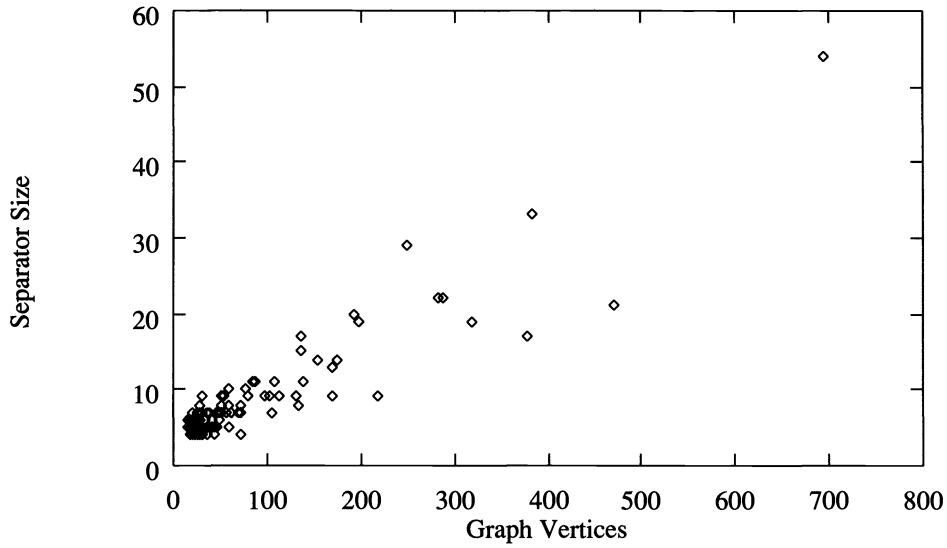


FIG. 6. *Separator size as a function of graph size.*

this sense the vertex separator approach was a disappointment. It would be preferable to have an alternate technique for dividing large problems into smaller ones that is more successful at generating a small number of uniquely realizable subproblems.

6.3. The optimization routines. As expected, the global optimization routines dominated ABBIE's running time. This is partially a consequence of the NP-hardness of the molecule problem, but it is also a reflection of the simplicity of the optimization routines encoded in ABBIE. A sophisticated optimizer should be able to reduce the running time substantially, so the times presented below should be taken as only a rough measure of the relative complexity of the optimization problems.

To determine the coordinates of the vertices ABBIE first employs a combinatoric approach to combine chunks and vertices as described in §5.1. Most of the optimization problems encountered in the set of test problems were completely solved this way. For all our problems the combinatorial operations were extremely efficient relative to the optimizations. For the largest problem all the combinatorial phases consumed a total of less than 50 CPU seconds, while the optimizations required many days.

The optimizer in ABBIE searches for a global minimizer by repeated local minimizations from random starting points. To mitigate concerns of particularly lucky or unlucky sequences of starting points, each of the six problems was run three times. The cost of the decomposition routines remained virtually unchanged, but the optimization time varied by as much as two orders of magnitude, as indicated by Table 3.

As discussed in §5, the cost of an optimization problem should grow as 2^{k-1}, where k is the number of chunks being combined. However, for our test problems we already knew the correct answer, and hence the appropriate parity for each chunk. We exploited this knowledge to reduce the actual computational effort by ensuring that all the parities for each optimization were correct. The resulting running time was then multiplied by 2^{k-1} to approximate a more realistic, unbiased run. The results in Table 3 were generated this way. Additional information may be available to the chemist that would resolve parities more directly. For instance, amino acids are

TABLE 3
Total minutes spent in global optimizer.

Amino acids	Trial 1	Trial 2	Trial 3	Average
20	1.9×10^3	9.0×10^2	4.3×10^2	1.1×10^3
40	4.5×10^4	7.5×10^5	1.2×10^6	6.6×10^5
60	6.6×10^6	2.2×10^6	4.7×10^6	4.5×10^6
80	8.8×10^5	3.6×10^5	3.5×10^3	4.1×10^5
100	1.3×10^5	3.9×10^4	2.8×10^5	1.5×10^5
124	2.5×10^5	1.1×10^5	1.8×10^5	1.8×10^5

generally found in only one of their two possible mirror images. This kind of insight could be used to greatly improve the performance of the optimizer.

The total optimization time presented in Table 3 shows an unexpected dependence on the size of the graph being realized. Except for the smallest problem, the two largest problems are the least expensive ones. This result is especially surprising since we expect larger problems to have to perform more optimizations. This expectation is borne out by the results presented in the second column of Table 4. Clearly, the optimization problems for the 40 and 60 amino acid problems are more difficult than those for the larger problems. We believe this is a consequence of using leading subchains of the protein to generate the intermediate test problems. As remarked above, unlike a full protein, a leading subchain will not generally form a compact structure. With a less dense conformation, there is less geometric data to work with.

Another way to consider the problem complexity is to look at the number of difficult optimization problems. We will consider an optimization problem to be large if it involves at least 25 variables. (Recall that if the vertices are treated individually each of them contributes three variables.) Not surprisingly, the number of large optimization problems increases with problem size, as indicated by the last column of Table 4.

TABLE 4
Number of optimizations.

Amino acids	Total optimizations	Large optimizations
20	2	1
40	7	1
60	15	1
80	21	2
100	22	3
124	32	7

For each of the test problems, the total optimization time is dominated by this subset of large problems; they always consume more than 99% of the total optimization time. A breakdown of these large problems is given in Table 5, in which the number of trials necessary to find the global optimum for the three trials is shown as a function of the number of variables and number of chunks in the optimization problem. The trials always had the parities of the chunks correct, so for a more correct measure of the difficulty of the problems the number of attempts should be multiplied by 2^{k-1}. With this dependence on parities removed, the number of trials should depend solely on the topography of the penalty function. Generally, we would expect the penalty function to become more complex as the number of variables increases, but the experimental data reveals a much more complicated situation. For example,

the problems encountered in the test problem with 100 amino acids show exactly the opposite behavior. The optimization with 30 variables is much more difficult than those with 39.

TABLE 5
Breakdown of large optimization problems.

Amino acids	Number of variables (chunks)	Number of starting attempts			
		Trial 1	Trial 2	Trial 3	Average
20	34 (7)	241	101	55	132
40	46 (8)	1425	17451	25443	14773
60	54 (7)	183092	61871	129258	124740
80	25 (5)	40	387	124	184
80	42 (7)	77470	25085	209	34255
100	39 (8)	6	6	14	9
100	30 (5)	57013	17088	101326	58476
100	39 (6)	1391	147	924	821
124	33 (7)	5395	417	2226	2679
124	39 (8)	30	1960	575	855
124	37 (5)	2745	2213	447	1802
124	28 (5)	632	364	502	499
124	39 (6)	22917	518	12261	11899
124	31 (5)	5	1	2	3
124	39 (10)	238	829	498	522

The values in Table 5 reveal why the test problems with 40 and 60 amino acids were so difficult. The optimization problems encountered were the largest of any in the test set. This led to a large number of trials, each of which involved large Hessians. In addition, these problems involved many chunks, which further increased the running time. However, the examples in the table indicate that size alone is a poor predictor of computational difficulty.

Without exception, the large, expensive problems all occurred while trying to combine a large number of chunks and vertices that were created by the small vertex separator. An alternate technique that decomposed a large graph into a small number of uniquely realizable subgraphs would reduce the incidence of such difficult optimizations with a corresponding dramatic improvement in run time.

Having said this, it is still true that the cost of an optimization problem tends to increase sharply as the problem size grows. This justifies the divide-and-conquer idea underlying ABBIE.

7. Conclusions and future work. The divide-and-conquer approach to the molecule problem exemplified by ABBIE shows promise at solving large, practically interesting instances of an NP-hard problem. This technique should work on a large class of instances of the molecule problem. Instances with many extra edges should decompose easily into manageable pieces, while those with very few edges will quickly be broken into uniquely realizable subgraphs. It is in the intermediate region where the decomposition approach may fail, when there are just enough edges for a unique solution but not enough for subgraphs to be unique.

Our recursive decomposition has several distinct advantages over other approaches to the molecule problem. First, if there is not enough information to uniquely solve the problem (the typical situation in chemical applications) ABBIE will identify and solve unique subproblems. The remaining degrees of freedom in the problem describe the range of solutions that are compatible with the data, and investigating this solution space is now reduced to a much smaller problem. Chemists are often interested in

this information for its own sake. The range of solutions may be related to the actual flexibility of the molecule, in which case the motions identified by ABBIE may have important physical significance.

Second, for many applications it is only a small portion of the molecule that is of interest, like a binding site. Even if there is not enough data to uniquely position the full molecule, ABBIE may be able to solve for the subproblem of interest. ABBIE will automatically identify the portions of the molecule that can be solved uniquely.

Third, the graph algorithms in ABBIE determine whether or not there is sufficient data to solve an instance of the problem. This knowledge can be used to direct further experiments. In this way, poorly posed problems can be readily identified and avoided.

Fourth is the problem of inconsistent data. In any physical experiment there can be some measurements that are in error. This is a difficult problem for all of the approaches to the molecule problem, and they typically find a solution that nearly, but not exactly, satisfies all the constraints. If there are a few bad values that are causing the confusion, identifying them would be extremely useful as they could then be discarded. The only previous techniques for identifying bad data involved repeated attempts to solve the full problem, each time discarding a few edges. If one of the runs produced an acceptable answer then a discarded edge must have been causing the confusion. Our recursive decomposition has the potential to simplify this task. Inconsistent data would be indicated by the inability to solve a particular subproblem, narrowing the location of the erroneous data to the values in this subproblem.

In addition to inconsistent data, we would like to be able to deal with the realistic problem in which distances are not known exactly. Much of the theory about unique realizations will no longer apply in the presence of measurement uncertainty. We believe that the underlying ideas will still be applicable in this case, but in a more heuristic way. For instance, a graph that violates the necessary conditions for uniqueness will still have multiple realizations in the presence of data uncertainty, which can still permit the decomposition into smaller subproblems. However, uniqueness is now harder (and probably impossible) to guarantee. But as long as the range of satisfying conformations of a subgraph remains relatively small, the decomposition approach is still appropriate. Instead of treating a solution to a subproblem as a rigid body, simply use it as a starting conformation for the subset of vertices, allowing their relative locations to change as the optimization proceeds. If the solution from the subproblem is near to the correct solution, then this should provide a good starting point for the succeeding optimization. This should significantly reduce the optimization effort. By treating the solution of subproblems as intelligent starting points for later optimizations, the overall difficulty of the problem should be reduced.

The algorithms in ABBIE could be improved in a number of ways. One of the asymptotically faster four-connectivity algorithms could be used, and sparse matrix algorithms could be used in the redundant rigidity calculations. Much greater savings could be realized by improving the optimization phase, by far the most time consuming portion of the code. The number of optimization variables could be reduced using more sophisticated combinatorial heuristics than those described in §5.1. A more sophisticated global optimization routine could be employed, like the stochastic technique of Rinnooy Kan and Timmer [27]. Stochastic techniques also have the advantage of being easy to parallelize [4]. Additional possible approaches to the global optimization problem would be the tunneling algorithm of Levy and Montalvo [21], or a simulated annealing approach [30]. In addition, there are various optimization tools that could improve the performance of the local optimizations. A quasi-Newton

approach could be used so that instead of refactoring the Hessian matrix at each step, the factorization would be approximated and updated in linear time. Also, sparsity within the Hessians could be exploited.

An alternate way to substantially reduce the cost of the optimizations would be to reconsider the way in which uniquely realizable subgraphs are decomposed into smaller problems. As mentioned in §6, the cost of the optimizations was dominated by problems involving many subgraphs. These problems were induced by the proliferation of smaller subgraphs generated by ABBIE's vertex separator algorithm. An alternate method that identified small uniquely realizable subgraphs more directly could have a dramatic impact on runtime.

More generally, we believe the basic ideas described in this paper have applicability beyond the molecule problem. Divide-and-conquer techniques have not been commonly used in optimization, primarily because it is difficult to figure out how they can be applied. There are three aspects to the molecule problem that make a recursive decomposition possible. First, the penalty function describing an instance of the problem expresses equality constraints, since each edge must achieve a specific distance. Second, the penalty function consists of a sum of simple subfunctions, each relating only a small number of variables; that is, it is partially separable. This allows for the identification of subproblems that completely contain a set of constraints. If, instead, the subfunctions coupled many variables, then it would be difficult to decompose the problem. Third, there is a deep combinatorial structure to the molecule problem that allows solvable subproblems to be identified. The first two of these properties are fairly common in optimization settings. Although the specific structure we have exploited is unique to the molecule problem, it is likely that other optimization problems have analogous structure that can be similarly utilized. Any problem that contains subproblems that can be solved independently should be amenable to the type of divide-and-conquer approach described here. The challenge is to identify this structure.

Acknowledgments. The ideas in this paper have been developed and refined in innumerable discussions with Tom Coleman and Bob Connelly. I am also indebted to Kate Palmer for many helpful conversations.

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] L. Asimow and B. Roth, *The rigidity of graphs*, Trans. Amer. Math. Soc., 245 (1978), pp. 279–289.

[3] ———, *The rigidity of graphs,* II, J. Math. Anal. Appl., 68 (1979), pp. 171–190.

[4] R. H. Byrd, C. L. Dert, A. H. G. Rinnooy Kan, and R. B. Schnabel, *Concurrent stochastic methods for global optimization*, Math. Programming, 46 (1990), pp. 1–30.

[5] J. Cheriyan and R. Thurimella, *On determining vertex connectivity*, Tech. Report UMIACS-TR-90-79, CS-TR-2485, Dept. of Computer Science, University of Maryland at College Park, 1990.

[6] R. Connelly, *Personal communication*, April 1989.

[7] ———, *Personal communication*, September 1989.

[8] ———, *On generic global rigidity*, in Applied Geometry and Discrete Mathematics, the Victor Klee Festschrift, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Volume 4, P. Gritzmann and B. Sturmfels, eds., AMS and ACM, 1991, pp. 147–155.

[9] H. Crapo, *Structural rigidity*, Topologie Structurale, 1 (1979), pp. 26–45.

[10] G. M. Crippen and T. F. Havel, *Distance Geometry and Molecular Conformation*, Research Studies Press Ltd., Taunton, Somerset, England, 1988.

[11] H. N. GABOW AND H. H. WESTERMANN, *Forests, frames and games: Algorithms for matroid sums and applications*, in Proc. 20th Annual Symposium on the Theory of Computing, Chicago, 1988, pp. 407–421.

[12] L. M. GIERASCH AND J. KING, EDS., *Protein folding: deciphering the second half of the genetic code*, AAAS, 1989.

[13] H. GLUCK, *Almost all simply connected closed surfaces are rigid*, in Geometric Topology, Lecture Notes in Mathematics No. 438, Springer-Verlag, Berlin, 1975, pp. 225–239.

[14] B. HENDRICKSON, *The Molecule Problem: Determining Conformation from Pairwise Distances*, Ph.D. thesis, Technical Report 90-1159, Cornell University, Dept. of Computer Science, Ithaca, NY, 1990.

[15] B. HENDRICKSON, *Conditions for unique graph realizations*, SIAM J. Comput., 21 (1992), pp. 65–84.

[16] J. E. HOPCROFT AND R. E. TARJAN, *Dividing a graph into triconnected components*, SIAM J. Comput., 2 (1973), pp. 135–158.

[17] H. IMAI, *On combinatorial structures of line drawings of polyhedra*, Discrete Appl. Math., 10 (1985), pp. 79–92.

[18] A. KANEVSKY AND V. RAMACHANDRAN, *Improved algorithms for graph four-connectivity*, in Proc. 28th IEEE Annual Symposium on Foundations of Computer Science, Los Angeles, October 1987, pp. 252–259.

[19] K. KILLIAN AND P. MEISSL, *Einige grundaufgaben der räumlichen trilateration und ihre gefährlichen örter*, Deutsche Geodätische Komm. Bayer. Akad. Wiss., A61 (1969), pp. 65–72.

[20] G. LAMAN, *On graphs and rigidity of plane skeletal structures*, J. Eng. Math., 4 (1970), pp. 331–340.

[21] A. V. LEVY AND A. MONTALVO, *The tunneling algorithms for the global minimizer of functions*, SIAM J. Sci. Stat. Comput., 6 (1985), pp. 15–29.

[22] J. W. H. LIU, *A graph partitioning algorithm by node separators*, ACM Trans. Math. Software, 15 (1989), pp. 198–219.

[23] G. L. MILLER, S.-H. TENG, AND S. A. VAVASIS, *A unified geometric approach to graph separators*, in Proc. 32nd IEEE Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, October 1991, pp. 538–547.

[24] J. MORÉ AND D. SORENSEN, *Computing a trust region step*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 553–572.

[25] K. A. PALMER, *Personal communication*, Chemistry Department, Cornell University, Ithaca, NY, March, 1990.

[26] K. A. PALMER AND H. A. SCHERAGA, *Standard-geometry chains fitted to* X*-ray derived structures: Validation of the rigid-geometry approximation.* ii. *systematic searches for short loops in proteins: applications to bovine pancreatic ribonuclease* A *and human lysozyme*, J. Comput. Chem., 13 (1992), pp. 329–350.

[27] A. H. G. RINNOOY KAN AND G. T. TIMMER, *A stochastic approach to global optimization*, in Numerical Optimization, P. Boggs, R. Byrd, and R. B. Schnabel, eds., Society for Industrial and Applied Mathematics, Philadelphia, 1984, pp. 245–262.

[28] B. ROTH, *Rigid and flexible frameworks*, Amer. Math. Monthly, 88 (1981), pp. 6–21.

[29] J. B. SAXE, *Embeddability of weighted graphs in k-space is strongly* NP*-hard*, in Proc. 17th Allerton Conference in Communications, Control and Computing, 1979, pp. 480–489.

[30] P. J. M. VAN LAARHOVEN AND E. H. L. AARTS, *Simulated Annealing: Theory and Applications*, D. Reidel Publishing Company, Boston, MA, 1987.

[31] W. WUNDERLICH, *Untersuchungen zu einem trilaterations problem mit komplaneren standpunkten*, Sitz. Osten. Akad. Wiss., 186 (1977), pp. 263–280.

[32] K. WÜTRICH, *The development of nuclear magnetic resonance spectroscopy as a technique for protein structure determination*, Accounts Chemical Res., 22 (1989), pp. 36–44.

[33] ———, *Protein structure determination in solution by nuclear magnetic resonance spectroscopy*, Science, 243 (1989), pp. 45–50.

SIAM J. OPTIMIZATION
Vol. 5, No. 4, pp. 858–870, November 1995

007

AN INFORMATION GLOBAL OPTIMIZATION ALGORITHM WITH LOCAL TUNING*

YAROSLAV D. SERGEYEV†

Abstract. We propose an algorithm using only the values of the objective function for solving unconstrained global optimization problems. This algorithm belongs to the class of the information methods introduced by Strongin [*Numerical Methods in Multiextremal Problems*, Nauka, Moskow, 1978] and differs from the other algorithms of this class by the presence of local tuning which spies on the changes of the Lipschitz constant of the objective function over different sectors of the search region. We describe two versions of the method: for solving one-dimensional problems and for solving multidimensional problems (using Peano-type space-filling curves for reduction of dimensionality). In both cases we establish sufficient conditions of convergence to the global minimum. We also report results of some numerical experiments.

Key words. global optimization, Lipschitz functions, numerical methods, convergence

AMS subject classification. 90C30

1. Introduction. Many numerical algorithms (see, e.g., the monographs and survey-scoped papers of Archetti and Schoen [1], Dixon and Szegö [2], [3], Hansen, Jaumard, and Lu [8], [9], Horst and Tuy [10], Rinnooy Kan and Timmer [17], Strongin [19], Törn and Zilinskas [23]) have been proposed to solve the one-dimensional unconstrained multiextremal problem

$$f^* = f(x^*) = \min\{f(x) : x \in [a, b]\}, \tag{1}$$

i.e., to find the global minimum f^* of a function $f(x)$ and a global minimizer x^* over an interval $[a, b]$. Consider a situation where little is known about the objective function $f(x)$.

(i) $f(x)$ is given in the form of a black box subroutine, which has a point $x \in [a, b]$ as input and $f(x)$ as output.

(ii) Lipschitz condition

$$|f(x') - f(x'')| \le L|x' - x''|, \qquad x', x'' \in [a, b], \tag{2}$$

with an unknown constant L, $0 < L < \infty$, holds for $f(x)$.

For solving the problem (1), (2) the information approach has been proposed by Strongin [19], [20]. The information algorithms are derived as optimal statistical decision functions within the framework of a stochastic model representing the function to be optimized as a sample of some random function. In this paper we present a new information algorithm based on the following idea.

Suppose that we have executed k iterations and have evaluated the objective function $f(x)$ at points $x^1, x^2, \ldots, x^k$ (we call these *trial points*). Original information algorithms produce a new trial point x^{k+1} to evaluate $f(x^{k+1})$ using an estimate μ of the Lipschitz constant L

$$\mu = \max\{|f(x_j) - f(x_{j-1})|/(x_j - x_{j-1}) : 2 \le j \le k\}, \tag{3}$$

where

$$a = x_1 < x_2 < \cdots < x_{k-1} < x_k = b \tag{4}$$

* Received by the editors February 16, 1993; accepted for publication (in revised form) July 6, 1994.

† ISI-CNR, Rende (CS), Italy and Nizhni Novgorod State University, Nizhni Novgorod, Russia (yaro@sc.deis.unical.it). This paper was written while the author was a visitor at the Department of Electronics, Informatics and Systemistics, University of Calabria, Italy.

are trial points $x^1, x^2, \ldots, x^k$ renumbered by subscripts in order of increasing coordinates. Here we construct local estimates μ_j of local Lipschitz constants L_j for every interval $[x_{j-1}, x_j], 2 \le j \le k$, where

$$(5) \qquad \begin{aligned} &|f(x') - f(x'')| \le L_j |x' - x''|, \quad x', x'' \in [x_{j-1}, x_j], \\ &\mu_j \le L_j, \quad \mu_j \le \mu \le L, \quad 2 \le j \le k. \end{aligned}$$

Thus, we try to tune the algorithm to the behaviour of the objective function over every interval $[x_{j-1}, x_j]$, $2 \le j \le k$. As will be shown hereafter, this approach permits us to accelerate the search compared to the original information methods.

The rest of the paper is constructed in the following way. In §2 the method is presented and sufficient conditions of convergence to global minima are proved. Section 3 contains a generalization on the multidimensional case based on the Peano-type space-filling curves. Section 4 describes results of some numerical experiments. Section 5 concludes the paper.

2. The one-dimensional algorithm. Let us describe the one-dimensional version (ODV) of the information algorithm with local tuning.

Starting points $x^1, x^2, \ldots, x^m, m \ge 2$, are fixed in such a way that $x^1 = a$, $x^2 = b$, and the other m-2 points are chosen arbitrarily. Values $f(x^1), \ldots, f(x^m)$ are calculated at these points. To choose the $(k+1)$th trial point $x^{k+1}, k \ge m$, we execute the steps of the following algorithm.

Step 1. Reorder the trial points $x^1, x^2, \ldots, x^k$ as shown in (4). Thus, the numeration by superscripts indicates the order of producing trial points in the course of time and the numeration by subscripts defines subintervals in which the search region is divided by the trial points (this numeration is changed after every iteration).

Step 2. Estimate Lipschitz constants $L_j, 2 \le j \le k$, from (5) by the values

$$(6) \qquad \mu_j = \max\{\lambda_j, \gamma_j\}, \qquad 2 \le j \le k,$$

where

$$(7) \qquad \lambda_j = \max\left[\frac{|z_i - z_{i-1}|}{x_i - x_{i-1}} : \begin{cases} i = 2, 3 & \text{if } j = 2, \\ i = j-1, j, j+1 & \text{if } 3 \le j \le k-1, \\ i = k-1, k & \text{if } j = k, \end{cases}\right],$$

$$(8) \qquad \gamma_j = \mu(x_j - x_{j-1})/X^{\max},$$

$$(9) \qquad X^{\max} = \max\{x_i - x_{i-1} : 2 \le i \le k\},$$

where μ is from (3) and $z_i = f(x_i)$, $1 \le i \le k$. If $\mu_j < \xi$ set $\mu_j = \xi$, where $\xi > 0$ is a parameter of the method.

Step 3. For all intervals $[x_{j-1}, x_j]$, $2 \le j \le k$, calculate characteristics

$$(10) \quad R(j) = r\mu_j(x_j - x_{j-1}) + (z_j - z_{j-1})^2/(r\mu_j(x_j - x_{j-1})) - 2(z_j + z_{j-1}), \; 2 \le j \le k,$$

where $r > 1$ is a reliability parameter.

Step 4. Execute the new trial at the point

$$(11) \qquad x^{k+1} = 0.5(x_t + x_{t-1} - (z_t - z_{t-1})/(r\mu_t)),$$

where

$$(12) \qquad t = \arg\max\{R(j) : 2 \le j \le k\}.$$

This algorithm belongs to both the class of sequential characteristical algorithms (see Grishagin [4], [5] and Grishagin, Sergeyev, Strongin [6]) and the class of partition algorithms (see Pinter [16]). In terms of the information approach (see Strongin [19], [20]) after normalization the characteristic $R(j)$ is interpreted as probability of the global minimizer location within the interval $[x_{j-1}, x_j]$ (where formula (11) estimates this location) in the course of the $(k+1)$th iteration.

In our approach we tune μ_j on the basis of local estimate λ_j and the global one γ_j, which controls authenticity of the local information in consideration. We execute this control by comparing the length of the current subinterval $[x_{j-1}, x_j]$ with the maximal (among all subintervals at the search region) length $X^{\max}$. If the interval $[x_{j-1}, x_j]$ is very wide, then we cannot trust the local information and must use global estimates.

If we know that there are d sectors within the search region where the objective function behaves differently, it is convenient to use d global estimates μ^h, $1 \leq h \leq d$, of the type (3). Thus, μ^h will estimate the Lipschitz constant over the hth sector, $1 \leq h \leq d$. In the following consideration we suppose that this additional information is absent.

The parameter ξ introduced in Step 3 reflects the following idea. If the estimate of the Lipschitz constants L_j is less than ξ we nevertheless suppose that L_j is at least equal to ξ.

Note that all values μ_j from (6) are recalculated at the $(k+1)$th iteration only if $X^{\max}$ or μ have been changed after the kth iteration. In the opposite case μ_j are calculated only for the intervals

$$[x_{t-2}, x_{t-1}], \quad [x_{t-1}, x^k], \quad [x^k, x_t], \quad [x_t, x_{t+1}],$$

where x^k is the point chosen according to (11), (12) at the kth iteration. For all other intervals the values μ_j remain the same.

Let us consider some convergence properties of the information algorithm with local tuning.

LEMMA 1.1. *If $\{x^k\}$ is a trial sequence generated by* ODV *in the course of minimizing a function $f(x)$, $x \in [a, b]$, satisfying* (2) *and x' is a limit point of $\{x^k\}$ such that $x' \neq a$, $x' \neq b$, then there exist two subsequences converging to x', one from the left and the other from the right.*

Proof. Let x' belong to an interval $[x_{s-1}, x_s]$, $s = s(k)$ after the kth iteration. Since x' is a limit point and due to (10)–(12), we have

$$\lim_{k \to \infty} (x_{s(k)} - x_{s(k)-1}) = 0.$$

Thus, in the case $x' \notin \{x^k\}$, sequence $\{x_{s(k)-1}\}$ converges to x' from the left and $\{x_{s(k)}\}$ converges from the right.

To demonstrate the lemma in the case $x' \in \{x^k\}$, we suppose that there is no sequence converging to x' from the left (the case when the right convergence is absent may be considered by analogy). Thus, there exists a number k' such that trials do not fall in the interval

$$[x_{s(k)-1}, x_{s(k)}], x_{s(k)} = x', k > k'.$$

For the characteristic $R(s)$ of this interval, the following chain of relations takes place:

$$\begin{aligned} R(s) &= r\mu_s(x' - x_{s-1}) + (z_{s-1} - f(x'))^2/(r\mu_s(x' - x_{s-1})) - 2(f(x') + z_{s-1}) \\ &= r\mu_s(x' - x_{s-1})[1 + (z_{s-1} - f(x'))^2/(r\mu_s(x' - x_{s-1}))^2 \\ &\qquad - 2(z_{s-1} - f(x'))/(r\mu_s(x' - x_{s-1}))] - 4f(x') \\ &= r\mu_s(x' - x_{s-1})[1 - (z_{s-1} - f(x'))/(r\mu_s(x' - x_{s-1}))]^2 - 4f(x') > -4f(x'). \end{aligned}$$

To obtain the last estimate we have used the inequality

$$r\mu_s > |f(x') - z_{s-1}|/(x' - x_{s-1}),$$

which holds due to (6), (7). On the other hand for the characteristic $R(s+1)$ of the interval $[x', x_{s(k)+1}]$, we obtain

$$\lim_{k\to\infty} R(s(k)+1) = -4f(x')$$

since x' is a limit point of $\{x^k\}$. Thus, for sufficiently large $k > k'$, it follows that

$$R(s(k)) > R(s(k)+1),$$

and due to (11), (12) a new trial will fall in the interval $[x_{s(k)-1}, x']$. This fact contradicts our assumption about the absence of sequence converging to x' from the left.

LEMMA 1.2. *For all trial points* x^k, $k \geq 1$, $f(x^k) \geq f(x')$.

Proof. Assume that an iteration d has produced a point x^d such that

$$z^d = f(x^d) < f(x').$$

Consider the characteristic $R(j)$ of the interval $[x^d, x_j]$

$$\begin{aligned} R(s) &= r\mu_j(x_j - x^d) + (z_j - f(x^d))^2/(r\mu_j(x_j - x^d)) - 2(f(x^d) + z_j) \\ &= r\mu_j(x_j - x^d) + (z_j - f(x^d))^2/(r\mu_j(x_j - x^d)) \\ &\quad -2(2\min\{z_j, f(x^d)\} + |z_j - f(x^d)|) \\ &= |z_j - f(x^d)|(r\mu_j(x_j - x^d)/|z_j - f(x^d)| + |z_j - f(x^d)|/(r\mu_j(x_j - x^d)) - 2) \\ &\quad - 4\min\{z_j, f(x^d)\} > -4\min\{z_j, f(x^d)\} > -4f(x'). \end{aligned}$$

Thus, new trials do not fall in the interval containing x' and, at the same time, x' is a limit point. We have contradiction that completes the proof.

LEMMA 1.3. *If there exists another limit point* $x'' \neq x'$, *then* $f(x'') = f(x')$.

Proof. Lemma 1.3 follows immediately from Lemma 1.2.

LEMMA 1.4. *If the function* $f(x)$ *has a finite number of local minima in* $[a, b]$ *then the point* x' *is locally optimal.*

Proof. If this lemma is not true then due to Lemma 1.1 a point y such that $f(y) < f(x')$ will be produced. But this is impossible because of Lemma 1.2.

Let x^* be a global minimizer of $f(x)$ over $[a, b]$ and $\{k\}$ be the sequence of all iteration numbers $\{k\} = \{1, 2, 3, \ldots\}$. The following theorem establishes sufficient conditions of global convergence of the sequence $\{x^k\}$ to x^*.

THEOREM 1.1. *If there exists an infinite subsequence* $\{h\}$ *of iteration numbers* $\{h\} \subset \{k\}$ *such that for an interval*

$$[x_{j-1}, x_j], j = j(l),\ l \in \{h\},$$

containing the point x^* *at the lth iteration, the inequality*

$$r\mu_j \geq K_j + (K_j^2 - M_j^2)^{1/2} \tag{13}$$

holds, then x^* *is a limit point of* $\{x^k\}$. *In* (13) *we have used the designations*

$$K_j = \max\{(z_{j-1} - f(x^*))/(x^* - x_{j-1}), (z_j - f(x^*))/(x_j - x^*)\}, \tag{14}$$

$$M_j = |z_{j-1} - z_j|/(x_j - x_{j-1}). \tag{15}$$

Proof. Suppose that there exists a limit point $x' \neq x^*$ of the trial sequence $\{x^k\}$. Taking into consideration (10) and Lemma 1.1 we can conclude for an interval $[x_{i-1}, x_i]$, $i = i(k)$, containing x' at the kth iteration of ODV, that

$$\lim_{k\to\infty} R(i(k)) = -4f(x'). \tag{16}$$

Consider now an interval $[x_{j-1}, x_j]$, such that

$$x^* \in [x_{j-1}, x_j], \tag{17}$$

and suppose that x^* is not a limit point of $\{x^k\}$. This signifies that there exists an iteration number m such that for all $k \geq m$

$$x^{k+1} \notin [x_{j-1}, x_j],\ j = j(k).$$

Estimate now the characteristic $R(j(k))$, $k \geq m$ of this interval. On the basis of (17) and (14) we can write

$$z_{j-1} - f(x^*) \leq K_j(x^* - x_{j-1}), \tag{18}$$

$$z_j - f(x^*) \leq K_j(x_j - x^*). \tag{19}$$

Then, summing (18) and (19) we obtain

$$z_{j-1} + z_j \leq 2f(x^*) + K_j(x_j - x_{j-1}).$$

From this inequality, using (13) and (15) we can deduce for all iteration numbers $l \in \{h\}$ that

$$\begin{aligned} R(j(l)) &= (x_j - x_{j-1})(r\mu_j + M_j^2(r\mu_j)^{-1}) - 2(z_{j-1} + z_j) \\ &\geq (x_j - x_{j-1})(r\mu_j + M_j^2(r\mu_j)^{-1} - 2K_j) - 4f(x^*) \geq -4f(x^*). \end{aligned} \tag{20}$$

Since x^* is a global minimizer and the sequence $\{h\}$ is infinite, then from (16) and (20) it follows that an iteration number l^* will exist such that

$$R(j(l^*)) \geq R(i(l^*)). \tag{21}$$

But, according to the decision rules (6)–(12) of ODV, this signifies that the l^*th trial will be executed at the interval (17). Thus, our assumption that x^* is not a limit point of $\{x^k\}$ is not true and the theorem has been proved.

COROLLARY 1.1. *Given the conditions of the theorem all limit points of* $\{x^k\}$ *are global minimizers of* $f(x)$.

Proof. The corollary is easily proved on the basis of Lemma 1.3.

Let X^* be the set of global minimizers of the function $f(x)$. Corollary 1.1 ensures that the set of limit points of $\{x^k\}$ belongs to X^*. Conditions when these sets are identical are established by Corollary 1.2.

COROLLARY 1.2. *If condition* (13) *is fulfilled for all points* $x^* \in X^*$, *then the set of limit points of* $\{x^k\}$ *coincides with* X^*.

Proof. This corollary is a straightforward consequence of the theorem and Lemma 1.3.

3. The multidimensional algorithm. Consider a generalization of the problem (1), (2) to the multidimensional case

$$F^* = F(y^*) = \min\{F(y) : y \in D\}, \tag{22}$$

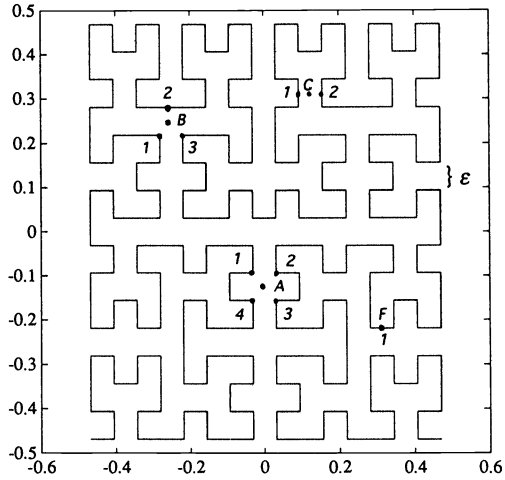


FIG. 1. *When the points on the plane are approximated by the points on the Peano curve there exist four variants of their mutual location.*

$$D = \{y \in \mathrm{R}^N : a_j \le y_j \le b_j,\ 1 \le j \le N\}, \tag{23}$$

where $F(y)$ is a Lipschitz function with a constant K, $0 < K < \infty$.

There exist at least three types of possible generalization of ODV to the multidimensional case. The first one is the well-known multistep scheme of nested optimization (see, e.g., Pijavskii [14])

$$\min_{y_1} \dots \min_{y_N} F(y_1, \dots, y_N),$$

where every one-dimensional problem may be solved by ODV. The second one is the scheme of generalization of one-dimensional algorithms proposed in Pinter [15]. In this paper the third approach, based on reduction of dimensionality using Peano (or Hilbert) curves (see Strongin [19], [21], Strongin and Sergeyev [22]), is applied.

As it has been proved in Strongin [19], solution of the problem (22), (23) may be obtained by minimizing a function

$$f(x) = F(y(x)),\ x \in [0, 1], \tag{24}$$

in the metric

$$\rho(x', x'') = |x' - x''|^{1/N}, \tag{25}$$

where N is from (23) and $y(x)$ is a space-filling Peano-type curve. In Fig. 1 we present an approximation of this curve. For the reduced function $f(x)$ the Hölder condition with the constant $L = 4K\sqrt{N}$ takes place

$$|f(x') - f(x'')| \le L|x' - x''|^{1/N}, \qquad x', x'' \in [0, 1], \tag{26}$$

where K is the Lipschitz constant of the function $F(y)$ (see Thm. 2.1 in Strongin and Sergeyev [22]).

In this section we present a multidimensional version (MDV) of the information algorithm with local tuning to solve the problem (24), (26) in the metric (25).

Suppose that k trials have already been done. To choose the $(k+1)$th trial point we proceed according to the following steps of MDV.

Step 1. Execute Step 1 of ODV.

Step 2. Evaluate the values μ_j according to (6), replacing $x_j - x_{j-1}$ by $(x_j - x_{j-1})^{1/N}$ in (3), (7), (8), and $X^{\max}$ by $(X^{\max})^{1/N}$ in (8). The values $f(x_i)$ are replaced by $F(y(x_i))$.

Step 3. For all intervals $[x_{j-1}, x_j]$, $2 \le j \le k$, calculate characteristics $R(j)$ according to (10), replacing $x_j - x_{j-1}$ by $(x_j - x_{j-1})^{1/N}$.

Step 4. Execute the new trial at the point

$$x^{k+1} = 0.5(x_t + x_{t-1}) - (2r)^{-1}(|z_t - z_{t-1}|/\mu_t)^N \operatorname{sign}(z_t - z_{t-1}),$$

where t is from (12).

The algorithm stops when

$$(x_t - x_{t-1})^{1/N} \le \varepsilon, \tag{27}$$

where $\varepsilon \ge \tau/(4\sqrt{N})$ is the given search accuracy and $\tau = 2^{-m}$ corresponds to the mth approximation of the Peano curve (see Strongin and Sergeyev [22]).

Remark. Note, that in spite of the ODV belonging to both classes of characteristical and partition algorithms in the one-dimensional case, MDV does not belong to the second of these classes in the multidimensional case. In fact, in our approach the interval partitioning takes place only after the Peano transformation of the original domain has been executed, whereas in partition algorithms the N-dimensional domain is partitioned.

LEMMA 2.1. *The results described in Lemmas* 1.1 – 1.4 *for* ODV *take place for* MDV *also.*

Proof. The corresponding results for MDV are obtained by repeating the proofs of Lemmas 1.1 – 1.4 introducing in the formulas the same changes that have been done in (6) – (11) to pass from ODV to MDV.

The following theorem generalizes Theorem 1.1 to the multidimensional case.

THEOREM 2.1. *If there exists an infinite subsequence* $\{h\}$ *of iteration numbers* $\{h\} \subset \{k\}$ *such that for an interval* $[x_{j-1}, x_j]$, $j = j(l)$, $l \in \{h\}$, *from* (17) *the inequality*

$$r\mu_j \ge 2^{1-1/N} K_j + (2^{2-2/N} K_j^2 - M_j^2)^{1/2} \tag{28}$$

holds, then x^* *is a limit point of* $\{x^k\}$. *Here*

$$K_j = \max\{(z_{j-1} - f(x^*))(x^* - x_{j-1})^{-1/N}, (z_j - f(x^*))(x_j - x^*)^{-1/N}\}, \tag{29}$$

$$M_j = |z_{j-1} - z_j|(x_j - x_{j-1})^{-1/N}. \tag{30}$$

Proof. Following the scheme of the proof of Theorem 1.1 we compare the characteristic of the interval $[x_{i-1}, x_i]$ containing a limit point $x' \ne x^*$ and the characteristic of the interval (17). For the first of these, (16) takes place. Let us estimate the characteristic of the interval (17).

From (29) and (17) we can obtain the inequalities

$$z_{j-1} - f(x^*) \le K_j(x^* - x_{j-1})^{1/N}, \tag{31}$$

$$z_j - f(x^*) \le K_j(x_j - x^*)^{1/N}. \tag{32}$$

Now, using (31), (32), and the designation

$$\alpha = (x^* - x_{j-1})/(x_j - x_{j-1}),$$

we deduce

$$\begin{aligned} z_{j-1} + z_j &\le 2f(x^*) + K_j((x^* - x_{j-1})^{1/N} + (x_j - x^*)^{1/N}) \\ &= 2f(x^*) + K_j(\alpha^{1/N} + (1-\alpha)^{1/N})(x_j - x_{j-1})^{1/N} \\ &\le 2f(x^*) + K_j(x_j - x_{j-1})^{1/N} \max\{\alpha^{1/N} + (1-\alpha)^{1/N} : 0 \le \alpha \le 1\} \\ &= 2(f(x^*) + 2^{-1/N} K_j(x_j - x_{j-1})^{1/N}). \end{aligned}$$

Using this estimate and (28), (30), we obtain

$$\begin{aligned} R(j(l)) &= r\mu_j(x_j - x_{j-1})^{1/N} + (z_{j-1} - z_j)^2 (r\mu_j)^{-1}(x_j - x_{j-1})^{-1/N} - 2(z_{j-1} + z_j) \\ &\ge (x_j - x_{j-1})^{1/N}(r\mu_j + M_j^2(r\mu_j)^{-1} - 2^{2-1/N}K_j) - 4f(x^*) \ge -4f(x^*) \end{aligned}$$

for all iteration numbers $l \in \{h\}$.

Thus, analogous to the corresponding part of the Theorem 1.1 proof, we can conclude that the inequality (21) holds and, consequently x^* is a limit point of the trial sequence $\{x^k\}$ produced by MDV.

COROLLARY 2.1. *Corollaries* 1.1 *and* 1.2 *of Theorem* 1.1 *take place for* MDV *also.*

Proof. The proof is completely analogous to the proofs of Corollaries 1.1 and 1.2.

For MDV the property of the bilateral convergence to x^* in the metric (25) has been presented (see Lemma 1.1). Let us establish a connection between convergence to a global minimizer x^* of the reduced problem (24)–(26) and convergence to the solution y^* of the original problem (22), (23). To characterize the type of convergence at the N-dimensional space, we introduce the notion of l-lateral convergence.

Let $\{y^k\}$, $y^k \in D$, be the sequence of points in D corresponding to the trial sequence $\{x^k\}$ generated by MDV, i.e.,

$$y^k = y(x^k),$$

where $y(x)$ is the Peano curve. Then, there exists a point

$$y' = (y'_1, y'_2, \dots, y'_N) \tag{33}$$

corresponding to a limit point x' of $\{x^k\}$. We partition the region D from (23) by N planes

$$y_1 = y'_1, \qquad y_2 = y'_2, \dots, y_N = y'_N$$

in 2^N sectors with the unique common vertex y' from (33). We give the following definition.

DEFINITION. *Convergence of* $\{y^k\}$ *to* y' *is* l*-lateral if there exist* l *sectors, containing subsequences of* $\{y^k\}$ *converging to* y'.

The Peano curves used for reduction of dimensionality establish a correspondence between subintervals of the curve and N-dimensional subcubes of D (for the detailed description of the Peano curves, see Strongin [19], [21] and Strongin and Sergeyev [22]). Every point on the curve approximates an ε-neighbourhood in D (see Fig. 1). Thus, the points in D may be approximated differently by the points on the curve in dependence on the mutual disposition between the curve and the point in D to be approximated. Here by "approximation" of a point $y \in D$ we mean the set of points on the curve minimizing the Euclidean distance from y.

For example (see Fig. 1), the point A has four images on the curve, B has three images, C has two, and F has only one image. It is easy to show that the number of the images ranges between 1 and 2^N. These images may be placed on the curve very far from each other in spite of vicinity in the N-dimensional space (see, for instance, images 1 and 2 of point A in Fig. 1). Thus, the point y^* from (22) may have up to 2^N images also, i.e., it is approximated by n, $1 \le n \le 2^N$, points y^{*i} such that

$$y^{*i} = y(x^{*i}), \qquad 1 \le i \le n, \quad \|y^{*i} - y^*\| \le \varepsilon, \tag{34}$$

where $\varepsilon > 0$ is defined by the Peano curve $y(x)$ and $\|\cdot\|$ is Euclidean norm. Thus, to obtain ε-approximation y^{*i} of the solution y^*, it is enough to find one of the points x^{*i} from (34). The above observation allows us to state the following result, connecting processes of solving problems (22), (23), and (24)–(26). The proof is obvious and we omit it.

PROPOSITION. *If the point y^* from* (22) *has n, $1 \le n \le 2^N$, images on the Peano curve and for m of them the conditions of Theorem* 2.1 *are fulfilled; then convergence to y^* will be l-lateral, where*

$$l \le 2^N - n + m.$$

Thus, the type of convergence inherent to MDV differs from the convergence of the other methods, which also do not use derivatives (see Pijavskii [14], Strongin [19], [20], Mladineo [13], Horst and Tuy [11], Pinter [15], and Wood [24]) in the following.

(i) All these methods have 2^N-lateral convergence.

(ii) To guarantee convergence to y^* these methods need the knowledge of the *precise* Lipschitz constant K for the *whole* region D. In contrast with these, MDV needs only the fulfillment of the condition (28) (which is considerably weaker than the Lipschitz condition) for one of the images of y^* (i.e., for a number of sectors at the neighbourhood of y^* and not for the whole region D).

4. Results of numerical experiments. We have done numerical experiments with the information algorithm with local tuning on the Mini-Supercomputer ALLIANT FX/80. Two series of experiments have been executed. In the first one we use the set of 20 test functions proposed in Hansen, Jaumard, and Lu [9] to compare performance of one-dimensional global optimization algorithms. We confront ODV with the following global optimization methods which as ODV do not use derivatives: Kushner [12], Evtushenko [7], Pijavskii [14], Strongin [19], [20]. These algorithms are denoted in Table 1 as KM, EA, PA, and SM, correspondingly. Following the schemes of these methods we have used the precise values of Lipschitz constant (see Hansen, Jaumard, and Lu [9]) executing experiments with KM, EA, and PA. We have done all experiments with the accuracy $\varepsilon = 0.0001(b-a)$, where ε is from (27) (for the one-dimensional case $N = 1$) and b, a are from (1). The parameters of the methods have been chosen as follows: for KM $\gamma = 1$, for SM $r = 2$.

Now let us discuss the choice of the ODV parameters ξ and r. The first of these is chosen as a small number greater than the number representing zero in the computer taken for ODV implementation. We have used $\xi = 10^{-6}$. To choose r we must use (13). Taking into account the fact that M_j may be equal to zero and μ_j is close to K_j we must choose $r \ge 2$. In our experiments we have used $r = 2$.

Global minima have been found by all the methods for all test functions except KM for Problem 17. Table 1 contains numbers of trials executed by the methods. In Fig. 2 we present Function 3 together with the trial points produced by ODV (the first from top to bottom line of sign + under the graph of the function) and by the original information algorithm (the second line of sign +). As it is seen from the figure, ODV provides high density of trials only in the

TABLE 1
Results of numerical experiments with one-dimensional test functions.

Problem	KM	EA	PA	SM	ODV
1	2327	4363	149	127	35
2	4693	1205	155	135	36
3	416	373	195	224	136
4	1241	2559	413	379	41
5	4153	607	151	126	45
6	4425	2146	129	112	54
7	4281	1560	153	115	39
8	355	389	185	188	132
9	3908	1068	119	125	42
10	1488	1887	203	157	40
11	6401	522	373	405	72
12	5633	1787	327	271	66
13	2289	3809	993	472	45
14	5377	347	145	108	50
15	6067	1251	629	471	63
16	1638	3953	497	557	53
17	529	951	549	470	101
18	5211	1762	303	243	41
19	2252	2054	131	117	34
20	3385	2545	493	81	42

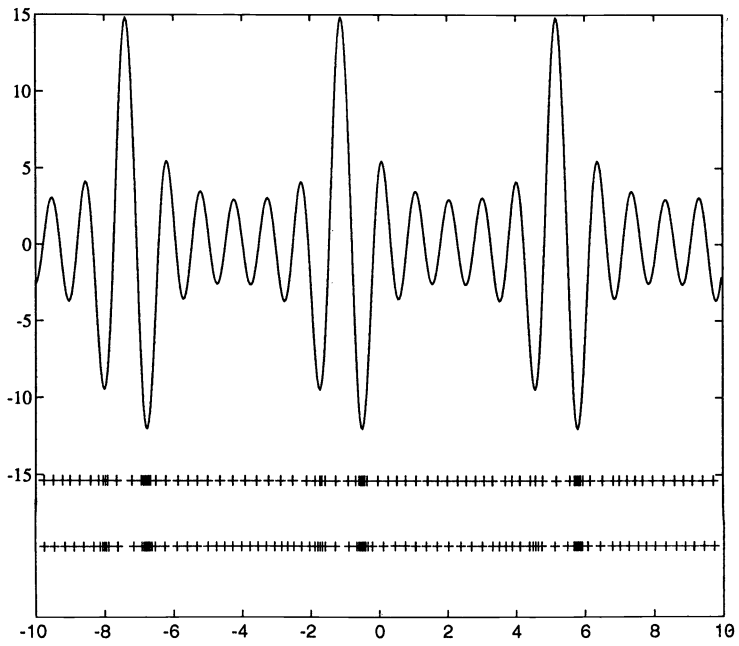


FIG. 2. *Trial points produced by* ODV *and* SM *in the course of solving Problem* 3.

neighbourhood of global minimizers. To solve this problem ODV has executed 136 trials in contrast with 224 produced by SM. The improvement (see Fig. 2) has been obtained in the regions (neighbourhoods of global and deep local minima) where local Lipschitz constants are less than the global one.

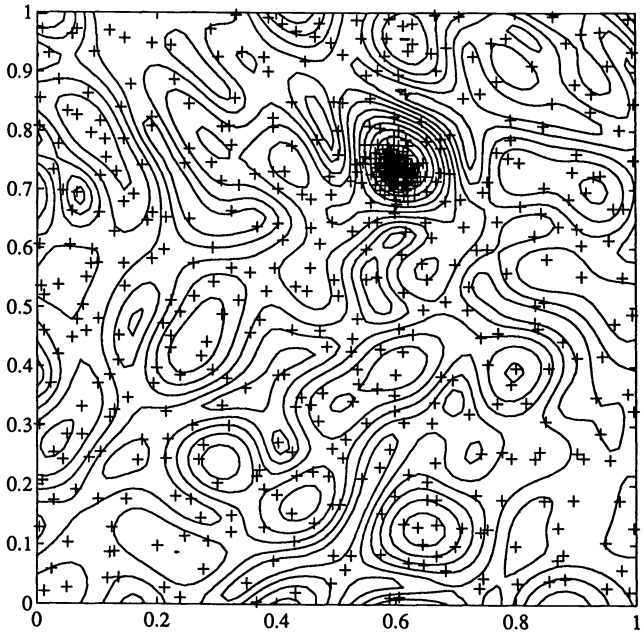


FIG. 3. *Trial points produced by* SM.

TABLE 2
Average results of numerical experiments with 100 *two-dimensional multiextremal functions.*

Method	r	%	Trials	Time	Speedup (trials)	Speedup (time)
SM	2.9	100	1575.12	70.036	—	—
MDV	2.9	98	351.37	11.150	4.49	6.28
MDV	3.1	100	425.62	15.396	3.71	4.55

In the second series of numerical experiments we have tested MDV and SM using 100 two-dimensional multiextremal functions from the following (see Grishagin [4]) class:

$$f(x) = \left\{ \left(\sum_{i=1}^{7} \sum_{j=1}^{7} [A_{ij} a_{ij}(x) + B_{ij} b_{ij}(x)] \right)^2 + \left(\sum_{i=1}^{7} \sum_{j=1}^{7} [C_{ij} a_{ij}(x) - D_{ij} b_{ij}(x)] \right)^2 \right\}^{1/2}$$

where $0 \le x_1 \le 1$, $0 \le x_2 \le 1$, and

$$a_{ij}(x) = \sin(i\pi x_1)\, \sin(j\pi x_2),$$
$$b_{ij}(x) = \cos(i\pi x_1)\, \cos(j\pi x_2),$$

and A_{ij}, B_{ij}, C_{ij}, D_{ij} are random coefficients from the interval $[-1, 1]$. Level curves of one of these functions are shown in Fig. 3.

All experiments were performed with initial points $\{0.2, 0.4, 0.6, 0.9\}$ and the search accuracy $\varepsilon = 0.001$, where ε is from (27). Both the methods have used the Peano curve approximation of the 12th order. MDV has used $\xi = 10^{-6}$. Table 2 contains the average results of the numerical experiments. We demonstrate dependence of the MDV performance on the reliability parameter r. To choose r we must use (28). Taking into account the fact

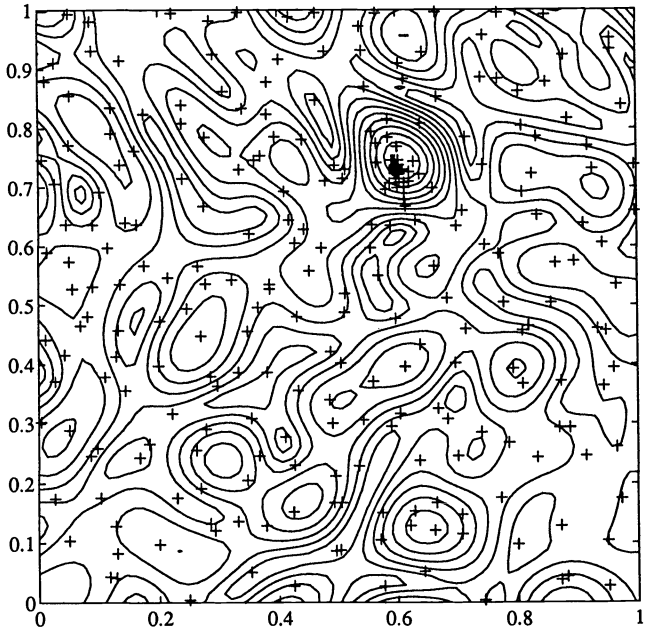


FIG. 4. *Trial points produced by* MDV.

that M_j may be equal to zero and μ_j is close to K_j, we must choose for $N = 2$ the parameter $r \geq 2^{2-1/2}$. The column % shows a quantity of experiments in which global minima have been found.

To solve the problem presented in Fig. 3, SM has produced 701 trials which are indicated by the sign $+$. The region with the most density of trial points is the global minimizer neighbourhood. In Fig. 4, we present 308 trials executed by MDV with $r = 2.9$ to solve the same problem. It is seen from the figure that MDV has unilateral convergence to the global minimizer and outside of its neighbourhood density of MDV trials is also less than SM density.

5. Conclusions. We have described an information algorithm for solving unconstrained global optimization problems. The algorithm proposed does not require the knowledge of derivatives or the Lipschitz constant and uses only the values of the objective function to achieve the global solution. This algorithm differs from the other methods belonging to the class of information algorithms by the presence of local tuning that spies on the changes of the Lipschitz constant of the objective function over different sectors of the search region.

For the one-dimensional version of the algorithm, the property of bilateral convergence and sufficient conditions of convergence to global minimizers have been established. Analogous results have been obtained for the multidimensional version of the method (using the Peano curves for reduction of dimensionality). A notion of the l-lateral convergence has been introduced and it has been demonstrated that the algorithm proposed has this type of convergence.

Numerical experiments executed with some well-known test functions confirm the theoretical results and demonstrate quite satisfactory performance of the information algorithm with local tuning compared to the other methods tested.

Acknowledgments. The author is greatly indebted to R. G. Strongin for many useful discussions and friendly support. He thanks the Department of Electronics, Informatics and Systemistics, University of Calabria, Italy, where he was a visitor.

REFERENCES

[1] F. Archetti and F. Schoen, *A survey on the global optimization problems: general theory and computational approaches*, Ann. Oper. Res., 1 (1984), pp. 87–110.

[2] L. C. W. Dixon and G. P. Szegö, *Towards Global Optimization* 1, North-Holland, New York, 1975.

[3] ———, *Towards Global Optimization* 2, North-Holland, New York, 1978.

[4] V. A. Grishagin, *Operation characteristics of some global optimization algorithms*, Prob. Stochastic Search, 7 (1978), pp. 198–206. (In Russian.)

[5] ———, *On convergence conditions for one class of global search algorithms*, Proc. All-Union Seminar Numerical Methods of Nonlinear Programming, Kharkov, 1979, pp. 82–84. (In Russian.)

[6] V. A. Grishagin, YA. D. Sergeyev, and R. G. Strongin, *An approach to the creation of parallel characteristical algorithms for global optimization*, Report No. 107, Systems Department, University of Calabria, Italy, 1991.

[7] Yu. G. Evtushenko, *Numerical methods for finding global extrema of a nonuniform mesh*, USSR Comput. Math. Math. Physics, 11 (1971), pp. 1390–1403.

[8] P. Hansen, B. Jaumard, and S.-H. Lu, *Global optimization of univariate Lipschitz functions*: 1. *Survey and properties*, Math. Programming, 55 (1992), pp. 251–272.

[9] ———, *Global optimization of univariate Lipschitz functions*: 2. *New algorithms and computational comparison*, Math. Programming, 55 (1992), pp. 273–293.

[10] R. Horst and H. Tuy, *Global Optimization—Deterministic Approaches*, Springer-Verlag, Berlin, 1993.

[11] ———, *On the convergence of global methods in multiextremal optimization*, J. Optim. Theory Appl., 54 (1987), pp. 253–371.

[12] H. Kushner, *A new method for locating the maximum point of an arbitrary multipeak curve in presence of noise*, J. Basic Engr., 86 (1964), pp. 97–106.

[13] R. H. Mladineo, *An algorithm for finding the global maximum of a multimodal, multivariate function*, Math. Programming, 34 (1986), pp. 188–200.

[14] S. A. Pijavskii, *An algorithm for finding the absolute extremum of a function*, USSR Comput. Math. Math. Physics, 12 (1972), pp. 57–67.

[15] J. Pinter, *Extended univariate algorithms for n-dimensional global optimization*, Computing, 36 (1986), pp. 91–103.

[16] ———, *Convergence qualification of adaptive partition algorithms in global optimization*, Math. Programming, 56 (1992), pp. 343–360.

[17] A. H. G. Rinnooy Kan and G. H. Timmer, *Global optimization*, Optimization, G. L. Nemhauser, A. H. G. Rinnooy Kan, and M. J. Todd, eds., North-Holland, Amsterdam, 1989.

[18] Ya. D. Sergeyev and R. G. Strongin, *A global optimization algorithm with parallel iterations*, USSR Comput. Math. Math. Physics, 29 (1989), pp. 7–15.

[19] R. G. Strongin, *Numerical Methods in Multiextremal Problems*, Nauka, Moskow, 1978. (In Russian.)

[20] ———, *The information approach to multiextremal optimization problems*, Stochastics Stochastics Rep., 27 (1989), pp. 65–82.

[21] ———, *Algorithms for multiextremal mathematical programming problems employing the set of joint space-filling curves*, J. Global Optim., 2 (1992), pp. 357–378.

[22] R. G. Strongin and Ya. D. Sergeyev, *Global Multidimensional Optimization on Parallel Computer*, Parallel Computing, 18 (1992), pp. 1259–1273.

[23] A. Törn and A. Zilinskas, *Global Optimization*, Lecture Notes in Computer Science 350, Springer-Verlag, New York, 1989.

[24] G. R. Wood, *The bisection method in higher dimensions*, Math. Programming, 55 (1992), pp. 319–337.

SIAM J. OPTIMIZATION
Vol. 5, No. 4, pp. 871–891, November 1995

008

POTENTIAL TRANSFORMATION METHODS FOR LARGE-SCALE GLOBAL OPTIMIZATION*

JACK W. ROGERS, JR.† AND ROBERT A. DONNELLY‡

Abstract. Several techniques for global optimization treat the objective function f as a force field potential. In the simplest case, trajectories of the differential equation $m\ddot{\mathbf{x}} = -\nabla f$ sample regions of low potential while retaining the energy to surmount passes that might block the way to regions of even lower local minima. A *potential transformation* is an increasing function $V:\mathbf{R} \to \mathbf{R}$. It determines a new potential $g = V(f)$, with the same minimizers as f, and new trajectories satisfying $m\ddot{\mathbf{x}} = -\nabla g = -\frac{dV}{df}\nabla f$. We discuss a class of potential transformations that greatly increase the attractiveness of low local minima and that provide, as a special case, a new approach to an equation proposed by Griewank for global optimization. Practical methods for implementing these ideas are discussed.

Key words. generalized descent, global optimization, Newtonian dynamics, potential transformation methods, PT methods, SNIFR

AMS subject classifications. 49D10, 90C30, 70D05

1. Introduction.

In this paper we present an approach to the global optimization of an unconstrained objective function $f \in C^1(\mathbf{R}^n)$ with numerous local minima in high dimensions. While the problem of local optimization has a solid mathematical theory and several highly efficient and practical algorithms, the problem of global optimization has proved to be much less tractable. Nevertheless, a number of methods have been proposed. We will give a quick summary of some of these methods and indicate some applications in the physical sciences, particularly in chemistry. Following [9, p. 3], we group the methods we discuss here into those that are *stochastic* in nature and those that are *deterministic.*

1.1. Stochastic methods.

The simplest practical stochastic method is the *multistart* method [9, p. 6], in which a random point is selected from a uniform distribution to initialize a local minimization algorithm. The process repeats until a stopping criterion is met. Of course, many initial points are selected in areas showing little promise of producing a new estimate for the global minimum. More efficient algorithms select a number of points at once and use a *clustering* method [3], [36] to place them into groups from which only one local minimization is performed. Alternatively, in the *multilevel single linkage* method [34], a *critical distance* is selected, and local minimization is initiated from each point generated unless it is within the critical distance of a point already used. These and similar ideas form the basis for several methods used by the chemical community. Investigations of small organic molecules (up to about 50 atoms—150 independent variables) were made by Chang, Guida, and Still [7], by Ferguson and Raber [14], and by Saunders [31].

A method particularly attractive to workers in the physical sciences is *simulated annealing* [22], in which the objective function is considered as the energy of a collection of atoms and an analog of temperature is used with an *annealing schedule* of melting and cooling to attempt to freeze the system at a global minimum. Such methods applicable to macromolecules were pioneered by workers in Scheraga's laboratory

* Received by the editors June 17, 1992; accepted for publication (in revised form) July 6, 1994.
† Department of Mathematics, Auburn University, Alabama 36849 (jrogers@mail.auburn.edu).
‡ Department of Chemistry, Auburn University, Alabama 36849 (donnelly@mail.auburn.edu).

[28], [24]. Donnelly [11] applied simulated annealing in a study of interacting propane molecules.

Stochastic ordinary differential equations (ODEs) can also be used to generate trajectories for global optimization. Aluffi-Pentini, Parisi, and Zirilli [1] have suggested using the Ito stochastic differential equation

$$d\mathbf{x} = -\nabla f dt + \epsilon d\mathbf{w}.$$

Solutions of this equation asymptotically approach probability densities with peaks at the global minimizers of f, and sharper peaks when ϵ is small. This suggests making ϵ a function of t converging to zero as $t \to \infty$, an idea similar to the annealing schedule used in simulated annealing. Applications in drug design have been made by C. Tosi and co-workers [15].

1.2. Deterministic methods. The theoretical advantage of stochastic methods lies in the possibility of proving that the probability of finding global minima approaches unity as the number of search steps increases without bound. For large-scale problems, however, the time required to assure a reasonably high probability makes such approaches very expensive. The deterministic methods we consider here offer no similar assurances of finding global minima but in practice may detect acceptably low values of the objective function much more quickly.

Some deterministic methods modify the objective function. Levy and Montalvo [23] use a descent algorithm, introducing singularities at local minima as they are found so that the algorithm does not return to them. Piela, Kostrowicki, and Scheraga [27] allow the objective function to "diffuse" so that most suboptimal local minima disappear; they then trace the remaining minima back to the original objective function.

ODEs have been used in a number of ways to construct deterministic trajectories to search for global minima. The simplest such ODE is that for continuous steepest descent,

$$\dot{\mathbf{x}}(t) = -\nabla f(\mathbf{x}(t)),$$

in which the search direction is always directly downhill. Trajectories commonly converge quickly to nearby local minima, making this method generally impractical for global minimization. Much more complicated behavior results if the equation for simple Newtonian dynamics

$$m\ddot{\mathbf{x}}(t) = -\nabla f(\mathbf{x}(t)), \tag{1.1}$$

is used instead, where the objective function f is considered to be the potential for the force field $\mathbf{F} = -\nabla f$, and the trajectory is that of a particle of constant mass m. Physical intuition suggests that trajectories seek out regions of low potential, while conservation of total energy gives the particle the ability to climb to passes possibly leading to basins of attraction of even lower minima [33], [4].

Molecular dynamics (MD) [25] is essentially Newtonian dynamics with a molecular potential function. It has found extensive use in the estimation of thermodynamic properties of macroscopic systems [2] and in macromolecular energy minimization. Practitioners using *constant-temperature* MD couple the system to an external temperature bath [4], which absorbs or dispenses the kinetic energy required to maintain a constant speed, as a means of avoiding potential orbiting about suboptimal local minima and also permitting more careful investigation of low potential values. In

dynamical simulated annealing (DSA) Car and Parinello [6] propose instead applying to MD, Kirkpatrick's idea of varying the temperature to alternately melt the system out of suboptimal minima and freezing it into lower energy configurations. DSA has become enormously popular in the solid-state physics community, where it has been used in structure calculations using density functional theory [20].

Incerti, Parisi, and Zirilli [21] propose incorporating a dissipative term into (1.1), yielding

$$m(t)\ddot{\mathbf{x}}(t) + c(t)\dot{\mathbf{x}}(t) = -\nabla f(\mathbf{x}(t)),$$

where $m > 0$ and $c < 0$. Instead of an annealing schedule, appropriate choices for the variable mass m and dissipative coefficient c can be used to achieve convergence to a low value of f.

In a deterministic approach very different from those just surveyed, Griewank [17] introduced the idea of a *target level*, c, as the current estimate of the value of the global minimum, and proposed some apparently very non-Newtonian dynamics for global optimization. In its final form, his equation is

$$\ddot{\mathbf{x}} = -\left[\epsilon I - (1+\epsilon)\frac{\dot{\mathbf{x}}\dot{\mathbf{x}}^T}{\|\dot{\mathbf{x}}\|^2}\right]\nabla f(\mathbf{x})[f(\mathbf{x}) - c)]. \tag{1.2}$$

Trajectories for this equation exhibit the following two characteristics, among others.

(i) Like Newtonian trajectories, they are continually deflected toward the direction of the negative gradient. Increasing ϵ increases the sensitivity to the local gradient.

(ii) The speed $v = \|\dot{\mathbf{x}}\|$ of propagation along the trajectory equals the height, $f-c$, of the objective function above the target level; contrary to Newtonian dynamics, the particle *accelerates* uphill and *decelerates* downhill. Convergence to any local minimum above the target level is impossible, since the speed is always positive for values of f above the target level.

Initial applications of this differential equation in [10] showed that it performs quite well in chemical problems which are replete with singularities, and hence often difficult to treat. However, an unacceptable amount of time was spent accurately propagating the molecular configurations. The secondary goal of solving the ODE seemed to be in conflict with the primary goal of efficiently searching the configuration space for global minima.

Consequently, Rogers and Donnelly [30] developed a discrete dynamical system modeled on the characteristics of Griewank's trajectories, using significantly larger stepsizes than are consistent with the accurate solution of an ODE. This algorithm has since become known as SNIFR. It has been applied to determine low energy molecular configurations [8], [18] and central configurations in celestial mechanics [32]. Butler and Slaminka [5] recently showed that SNIFR clearly outperformed simulated annealing on a standard set of test problems, all of dimension ≤ 10. In spite of this practical success, it is difficult to analyze theoretically the performance of SNIFR, since it is designed simply to mimic efficiently certain heuristics of Griewank's equation.

The purpose of this paper is to describe a new method based on standard Newtonian dynamics, satisfying well-understood principles such as the conservation of energy. A surprising conclusion is that the paths traversed by the trajectories of Griewank's clearly non-Newtonian equation are in fact Newtonian paths, subject to the same well-understood principles. Moreover, we propose a highly efficient implementation of this method for global optimization.

2. Potential transformation (PT) methods. There are some obvious drawbacks to the straightforward application of simple Newtonian dynamics to the problem of global optimization. Although trajectories tend to seek out regions of low potential, the kinetic energy of the particle in these regions is high. Since curvature is inversely proportional to the particle's kinetic energy, the local gradient in such regions has little influence in guiding the trajectory to local minima. To attempt to make low local minima more attractive, we introduce an increasing *potential transformation* $V : \mathbf{R} \to \mathbf{R}$, which is used to define a new function $g = V(f)$ with the same equipotential surfaces as f, but with revised values on these surfaces. The fact that V is increasing guarantees that the values of $\mathbf{x}$ that produce local and global optima are precisely the same as for the original potential function.

Subject to the new potential, the particle satisfies

$$m\ddot{\mathbf{x}} = -\nabla g = -\frac{dV}{df}\nabla f. \tag{2.1}$$

Since the trajectories are generated by a gradient field, energy is conserved. Thus if $v = \|\dot{\mathbf{x}}\|$, the kinetic energy is $T_v = \frac{1}{2}mv^2$, and the initial energy is $E_0 = T_{v_0} + V(f(\mathbf{x}_0))$, then, for all $t \geq 0$,

$$T_v + V(f) = E_0. \tag{2.2}$$

It will prove useful later to observe that, for a given energy level E_0, T_v is thus completely determined by the value of f.

Although much of what follows can be shown for more general functions, we will concentrate on potential transformations of the type

$$V(f) = -(f - c)^{-2\epsilon}, \tag{2.3}$$

using Griewank's idea of the *target level*, c, and using $\epsilon > 0$ as a parameter. Equation (2.1) implies that the effect of the potential transformation on the gradient is governed by

$$\frac{dV}{df} = -\frac{2\epsilon V}{f - c}. \tag{2.4}$$

The effect of this particular transformation is, first, to flatten the potential function for values of f sufficiently above the target level, thus lessening the effect of local perturbations; and, second, to place points at the target level at the base of an infinite potential well, making them more attractive for solutions of (2.1). Also important is the effect on the speed of the particle when searching relatively low values of the objective function. This is best shown by an example.

If the initial potential energy of a stationary particle of unit mass subject to standard Newtonian dynamics is $f_0 = 50$, then, when $f = 1$, its kinetic energy is $T(1) = 49$, and its speed is $v(1) \approx 9.9$. For $\epsilon = 1$ and $c = 0$, the corresponding transformed potential values would be $V(50) \approx -.0004$ and $V(1) = -1$, with a kinetic energy of $T(1) \approx 1$ and a speed of $v(1) \approx 1.4$. Substituting these parameters into (2.4) yields $dV/df = 2$, so it follows from (2.1) that the force field is approximately doubled when $f - c \approx 1$. Since trajectory curvature is proportional to the magnitude of the force field, and inversely proportional to kinetic energy (see (4.5)), the transformed potential results in much greater curvatures in response to the local gradient, increasing the likelihood of the particle being deflected into the enhanced basins of attraction of local minima near the target level.

3. Coarsely discretized ODE solvers. Of course, the use of ODEs to find global minima is simply a tool; the primary goal is efficient global minimization. High-order methods and small stepsizes can use up many function evaluations without yielding much progress in searching the domain of the objective function for regions of attraction of global minima. Moreover, while an ODE may efficiently find low values of the objective function, it may not be very effective at finding the actual minimizers.

For these reasons, we attack the problem of global optimization in two phases. In the first, the *global search* phase, we employ a *coarsely discretized* ODE solver, i.e., a low-order method, such as the first-order Euler method, with a relatively large stepsize, to efficiently search for low values of the objective function. In the second, the *local optimization* phase, one or more of the points just found are used to start an efficient local optimization algorithm to accurately determine the position of the minimizers. More details on the implementation will be given in later sections.

4. Reparameterization. This is not to say that the spatial stepsize during the search phase can be arbitrary. Large stepsizes can be tolerated where the objective function is high, since the goal in these regions is to sample a large area for lower values. On approach to the target level, however, the trajectories should be more accurately tracked in order to carefully examine the region for global minima. One way to accomplish this is through the use of variable stepsize ODE solvers, but our experience with these suggests that the attempt to control local truncation errors leads to unacceptably small steps near the target level, where curvatures can be extremely high. We have had more success by *reparameterizing* the solutions of (2.1) so that the speed with which they are traversed decreases as the curvature increases. This results in variable *spatial* stepsizes, even with the use of a simple fixed time-stepsize ODE solver.

We first obtain a general formula for reparameterizing the original potential function; the formula for a transformed potential can easily be obtained from it by substitution. First, recall that the arc length s of any twice-differentiable trajectory, $\mathbf{x}(t)$, satisfies $ds/dt = v \equiv \|\dot{\mathbf{x}}\|$. Defining $\mathbf{x}' \equiv d\mathbf{x}/ds$, we have

$$\dot{\mathbf{x}} = \frac{d\mathbf{x}}{dt} = \frac{d\mathbf{x}}{ds}\frac{ds}{dt} = v\mathbf{x}', \tag{4.1}$$

so that $\|\mathbf{x}'\| = \|\dot{\mathbf{x}}\|/v \equiv 1$. Differentiation of $\mathbf{x}' \cdot \mathbf{x}' \equiv 1$ yields $\mathbf{x}'' \cdot \mathbf{x}' = 0$; consequently $\mathbf{x}''$ is always orthogonal to $\mathbf{x}'$.

Assuming that $\mathbf{x}$ represents the path of a particle of constant mass m subject to a force $\mathbf{F}$ and that $T_v = \frac{1}{2}mv^2$ is its kinetic energy, (4.1) yields

$$\begin{aligned} \mathbf{F} = m\ddot{\mathbf{x}} &= \frac{d}{ds}(mv\mathbf{x}')\frac{ds}{dt} = (mv'\mathbf{x}' + mv\mathbf{x}'')v \\ &= T_v'\mathbf{x}' + 2T_v\mathbf{x}''. \end{aligned} \tag{4.2}$$

In view of the orthogonality of $\mathbf{x}'$ and $\mathbf{x}''$, we have

$$P_{\mathbf{x}'}\mathbf{F} = T_v'\mathbf{x}' \quad \text{and} \quad Q_{\mathbf{x}'}\mathbf{F} = 2T_v\mathbf{x}'', \tag{4.3}$$

where

$$P_{\mathbf{x}'}\mathbf{F} \equiv (\mathbf{x}' \cdot \mathbf{F})\mathbf{x}'$$

is the tangential component of $\mathbf{F}$ and $Q_{\mathbf{x}'}\mathbf{F} = \mathbf{F} - P_{\mathbf{x}'}\mathbf{F}$ is the normal component of $\mathbf{F}$. Note that these projections can be computed in terms of $\dot{\mathbf{x}}$, since

$$P_{\mathbf{x}'}\mathbf{F} = P_{\dot{\mathbf{x}}}\mathbf{F} \equiv \frac{\dot{\mathbf{x}} \cdot \mathbf{F}}{v^2}\dot{\mathbf{x}}.$$

Equation (4.3) can be solved for $\mathbf{x}''$ to yield

$$\mathbf{x}'' = -\frac{1}{2T_v} Q_{\mathbf{x}'} \nabla f, \tag{4.4}$$

which is a function of $\mathbf{x}$ and $\mathbf{x}'$ for a given trajectory because $T_v = E_0 - V(f(\mathbf{x}))$ by conservation of energy. This is, in fact, the ODE for the reparameterization of the original trajectory by arc length, and so it yields the definition of the *curvature* of the trajectory:

$$\kappa = \|\mathbf{x}''\| = \frac{\|Q_{\mathbf{x}'} \nabla f\|}{2T_v}. \tag{4.5}$$

To insure that another ODE produces the same trajectory, we must retain the curvature given by (4.4) at each point.

Suppose then that this trajectory is reparameterized by τ, so that the new speed satisfies $\sigma = ds/d\tau \equiv \overset{*}{s}$. Equation (4.2) then becomes

$$m \overset{**}{\mathbf{x}} = T'_\sigma \mathbf{x}' + 2T_\sigma \mathbf{x}''. \tag{4.6}$$

We require that the speed σ for a given trajectory be a function of the potential f alone, so that the kinetic energy $T_\sigma = \frac{1}{2} m \sigma^2$ also depends only on f. Then

$$T'_\sigma \mathbf{x}' = \frac{dT_\sigma}{df} (\nabla f \cdot \mathbf{x}') \mathbf{x}' = \frac{dT_\sigma}{df} P_{\mathbf{x}}^* \nabla f.$$

Substituting this and (4.4) into (4.6) then yields the reparameterized ODE for the *original* potential:

$$m \overset{**}{\mathbf{x}} = \frac{dT_\sigma}{df} P_{\mathbf{x}}^* \nabla f - \frac{T_\sigma}{T_v} Q_{\mathbf{x}}^* \nabla f. \tag{4.7}$$

For the *transformed* potential, the equation for the curvature (4.5) becomes

$$\kappa = \frac{\left\| Q_{\mathbf{x}}^* \nabla g \right\|}{2T_v} = \frac{dV/df}{2T_v} \left\| Q_{\mathbf{x}}^* \nabla f \right\|. \tag{4.8}$$

Substituting (2.2) and (2.4), we have

$$\frac{dV/df}{T_v} = \frac{-2\epsilon V}{(f-c)(E_0 - V)} = \frac{2\epsilon}{(f-c)(1 - E_0/V)}. \tag{4.9}$$

The idea is to choose the speed σ to be small wherever the curvature κ is large, e.g., when $(f-c)(1 - E_0/V) \approx 0$. This suggests choosing

$$\sigma = \frac{\mu}{m} (f - c) \left(1 - \frac{E_0}{V} \right), \tag{4.10}$$

where μ is included only for dimensional consistency: its magnitude is unity and it has units of *mass* $\times$ *speed / energy* = $(speed)^{-1}$. The equation satisfies the above condition that σ is a function of f alone along a given trajectory, since E_0 and c are constant and V is a function of f.

To obtain from (4.7) the ODE for the transformed potential, note that $dT_\sigma/dg = (dT_\sigma/df)/(dV/df)$ and $\nabla g = (dV/df)\nabla f$. Thus $(dT_\sigma/dg)P_{\overset{*}{\mathbf{x}}}\nabla g = (dT_\sigma/df)P_{\overset{*}{\mathbf{x}}}\nabla f$, and (4.7) becomes

$$m\overset{**}{\mathbf{x}} = \frac{dT_\sigma}{df}P_{\overset{*}{\mathbf{x}}}\nabla f - \frac{T_\sigma}{T_v}\frac{dV}{df}Q_{\overset{*}{\mathbf{x}}}\nabla f. \tag{4.11}$$

Equations (4.9) and (4.10) yield

$$\frac{T_\sigma}{T_v}\frac{dV}{df} = T_\sigma\frac{2\epsilon\mu}{m\sigma} = \epsilon\sigma\mu,$$

and

$$\frac{dT_\sigma}{df} = m\sigma\frac{d\sigma}{df} = \sigma\mu\left(1-(1+2\epsilon)\frac{E_0}{V}\right).$$

Thus (4.11) becomes the final equation for the PT method:

$$m\overset{**}{\mathbf{x}} = \left[\left(1-(1+2\epsilon)\frac{E_0}{V}\right)P_{\overset{*}{\mathbf{x}}}\nabla f - \epsilon Q_{\overset{*}{\mathbf{x}}}\nabla f\right]\sigma\mu. \tag{4.12}$$

Note that $\sigma\mu$ is a unitless number with magnitude of σ; if physical units are not a concern, then μ can be omitted.

It can be shown that the norm of the derivative of $P_{\overset{*}{\mathbf{x}}}$ is $\left\|\overset{*}{\mathbf{x}}\right\|^{-1}$, which is unbounded for $\overset{*}{\mathbf{x}}$ near $\mathbf{0}$. However, the norm of the derivative of $\left\|\overset{*}{\mathbf{x}}\right\|P_{\overset{*}{\mathbf{x}}} = \sigma P_{\overset{*}{\mathbf{x}}}$ is bounded (by 2, in fact), and it follows that $\partial\overset{**}{\mathbf{x}}/\partial\overset{*}{\mathbf{x}}$ is bounded. If $\nabla^2 f$ is bounded over the region accessible to a particle with energy E_0, then it is easily seen that $\partial\overset{**}{\mathbf{x}}/\partial\mathbf{x}$ is also bounded. Thus, choosing σ to compensate for the trajectory curvature in this way yields existence and uniqueness conditions for (4.12), as well as bounds on the errors of numerical approximations.

Inaccuracies in our coarsely discretized solver can cause the speed of the computed trajectory to deviate rapidly from (4.10). Consequently, we renormalize $\overset{*}{\mathbf{x}}$ at each step to enforce (4.10).

5. Relationship to Griewank's ODE. Equation (4.10) for the speed for this ODE is reminiscent of the equation for the speed for Griewank's ODE (1.2), in that the speed is approximately equal to $f-c$ for values of f near the target level. This is not an accident. If the initial energy is taken to be $E_0 = 0$ and μ is ignored, then (4.12) becomes

$$m\overset{**}{\mathbf{x}} = \left[P_{\overset{*}{\mathbf{x}}}\nabla f - \epsilon Q_{\overset{*}{\mathbf{x}}}\nabla f\right](f-c),$$

which is easily seen to be equivalent to (1.2) when $m = 1$. Since all values of the transformed potential (2.3) are negative, an energy level of $E_0 = 0$ imposes no upper limit on the value of f, and trajectories with unbounded potential are possible for Griewank's ODE. Equation (4.12), however, allows the choice of an upper bound for the potential, $f_{\max}$, resulting in $E_0 = V(f_{\max})$. Then, if $V = E_0$, all the energy has been converted to potential energy, leaving zero kinetic energy. Thus conservation of energy for (2.1) prohibits the trajectory from surpassing this limit, and $f_{\max}$ provides

control over high values of the objective function, just as the target level, c, provides control over the lower values. It can be seen from (4.10) that, like Griewank's particle, ours accelerates uphill near the target level, but, as in Newtonian dynamics, ours decelerates uphill near the upper energy limit.

It is important for this discussion to realize that the clearly non-Newtonian behavior exhibited by a solution of (4.12) results from reparameterizing, for numerical reasons, a trajectory that originates from the Newtonian equation (2.1), with potential V given by (2.3). To be more precise, if f is used as the potential function, certain Newtonian trajectories result. Introducing a potential transformation changes these trajectories to new ones, which are still Newtonian—but with a new potential function. As a simple example, consider the harmonic oscillator with potential $f(\mathbf{x}) = r^2 = \mathbf{x}\cdot\mathbf{x}$, resulting in a force field satisfying Hooke's law: $m\ddot{\mathbf{x}} = -\nabla f = -2\mathbf{x}$. Excluding degenerate cases, the resulting trajectories are ellipses centered at the origin. Now consider the potential transformation (2.3) with $c = 0$: $g(\mathbf{x}) = V(f(\mathbf{x})) = -f^{-2\epsilon} = -r^{-4\epsilon}$. The new ODE is

$$m\ddot{\mathbf{x}} = -\nabla g = -\frac{4\epsilon}{r^{4\epsilon+1}}\frac{\mathbf{x}}{r}.$$

The qualitative features of the trajectories associated with these attractive inverse power laws are well understood [16, pp. 76–82]. In particular, if $\epsilon = 1/4$, we have $m\ddot{\mathbf{x}} = -(1/r^2)(\mathbf{x}/r)$, an example of the inverse square law governing classical planetary motion, with trajectories which are conic sections with one focus at the origin [16, p. 96]: a single branch of a hyperbola for $E_0 > 0$, a parabola for $E_0 = 0$ (Griewank's equation), or an ellipse for $E_0 < 0$ (the case discussed in the preceding paragraph, with $E_0 = V(f_{\max}) = -r_{\max}^{-4\epsilon} < 0$). The trajectories for g are clearly different from those for f, but they are still Newtonian: particles attached to the origin by weightless springs have been replaced, conceptually, by celestial objects subject to the gravitational attraction of the sun. The subsequent reparameterization does not change the path of these new trajectories, only the way in which they are traversed. Thus, in the special case of Griewank's equation ($E_0 = 0$) with $c = 0$ and $\epsilon = 1/4$, the parabolic path about the sun of the planet, or, rather, what must be a comet, is traversed with speed given by (4.10):

$$\sigma = (f - c)\left(1 - \frac{E_0}{V}\right) = r^2.$$

Instead of following Kepler's law and generating equal areas in equal times, the comet, after reparameterization, slows as it approaches the sun, then speeds up as it moves farther away. While the behavior of solutions of the reparameterized ODE are clearly non-Newtonian, they allow for much better numerical approximation of the Newtonian parabolic path with a fixed time-stepsize ODE solver for orbits that approach near the sun, which represents a singularity in the force field.

Thus, in (4.12), one's intuition of Newtonian dynamics *for the transformed potential* is still applicable to the *path* of the trajectory, but not to the behavior of the particle itself.

Certain other characteristics of Griewank's equation become much more intuitive, once the Newtonian nature of the trajectory paths is recognized. For example, in [17, p. 17] it is observed that the direction of the velocity vector is a weighted average of the previous gradient vectors. We now see that this follows directly from (2.1):

$$m\ddot{\mathbf{x}} = \frac{d}{dt}m\dot{\mathbf{x}} = -\frac{dV}{df}\nabla f \Longrightarrow \dot{\mathbf{x}} = \dot{\mathbf{x}}_0 - \int_0^t \frac{1}{m}\frac{dV}{df}\nabla f\, d\tau,$$

with $\frac{1}{m}\frac{dV}{df}$ as the weighting factor.

As a second example, consider the table [17, p. 23] specifying the circumstances in which convergence or divergence can be proved for Griewank's ODE when f is homogeneous of degree δ. We assume for simplicity that the origin is the center point, from which it follows that $\nabla f(\mathbf{x}) \cdot \mathbf{x} = \delta f(\mathbf{x})$ for every $\mathbf{x} \in \mathbf{R}^n$, and we deal with $r^2 = \mathbf{x} \cdot \mathbf{x}$, rather than with r, since the former has a simpler derivative, and the latter has no derivative at all at $\mathbf{0}$. We have $dr^2/dt = 2\mathbf{x} \cdot \dot{\mathbf{x}}$ and

$$m\frac{d^2r^2}{dt^2} = 2m\dot{\mathbf{x}} \cdot \dot{\mathbf{x}} + 2m\mathbf{x} \cdot \ddot{\mathbf{x}} = 4T - 2\mathbf{x} \cdot \frac{dV}{df}\nabla f = 4T - 2\delta\frac{dV}{df}f.$$

Note that this implies that d^2r^2/dt^2 depends only on the value of f (this is not true of d^2r/dt^2—another reason for dealing with r^2 instead). Applying (2.4) and (2.2) yields

$$m\frac{f-c}{4T}\frac{d^2r^2}{dt^2} = (f-c) - 2\delta\frac{(f-c)}{4(E_0 - V)}\left(\frac{-2\epsilon V}{f-c}\right)f = \left[1 - \frac{\epsilon\delta}{1 - \dfrac{E_0}{V}}\right]f - c.$$

Since $E_0 = 0$ for Griewank's equation, we see that the sign of d^2r^2/dt^2 is the same as the sign of $(1-\epsilon\delta)f - c$ as long as neither $f - c$ nor T is zero. This is the expression in (16) of [17, p. 24] needed to argue the cases in the table in that paper. Our derivation, however, follows a more conventional line of reasoning about the qualitative behavior of trajectories subject to homogeneous potentials.

6. Setting the parameters. The purpose of this paper is to introduce a new method, and to indicate its general strengths and weaknesses. We are not concerned here with particular stopping rules or complicated automated algorithms for parameter determination, since these can be implemented in several, possibly problem-dependent, ways; the apparent success or failure of the resulting algorithm may depend more on these decisions than on the basic properties of the algorithm itself. However, a few comments about the parameters ϵ, c, and $f_{\max}$ are in order.

First, for arbitrary global optimization problems, there is the natural question of how one estimates the value of the global minimum to obtain a value for the target level, c. The determination of $f_{\max}$ presents a similar problem. In some problems, e.g., solving $f(\mathbf{x}) = \mathbf{0}$ by minimizing $\|f(\mathbf{x})\|_2^2$, the global minimum is known to be 0. Similarly, some types of problems may yield a natural upper bound on the objective function. In general, however, neither is known. One strategy is to set initial values for c and $f_{\max}$, and then, if f drops below c, to lower the value of c according to some schedule. It is also possible to raise c if a sufficiently long interval has passed with no new overall minimum occurring. As we will see later, the exact value of c is not as important as might be thought; a value a bit below the global minimum is usually better than one set exactly at it. Consequently, a better strategy may be to lower c if f is sufficiently close to it, as opposed to actually below it. Similarly, if $V(f)$ is sufficiently close to E_0, the value of E_0 might be raised, in hopes of allowing the trajectory to traverse a high pass from which it would otherwise be excluded, and which might lead to regions of even lower minima. If this is done, it is important to realize that the particle's kinetic energy has been correspondingly increased, and its speed needs to be readjusted accordingly. In this way, both c and $f_{\max}$ might be updated progressively as the algorithm progresses.

The value of ϵ, as well as the stepsize used, is another matter. We treat both as constant for the duration of the run, and determine their values from a few initial runs.

The stopping criterion obviously has a strong effect on the number of function evaluations required for a run. A number of sophisticated stopping criteria have been suggested. The most theoretically appealing satisfy some Baysian criterion. Unfortunately, these tend to require a large number of function evaluations even in relatively small dimensions. For many practical problems, the dimensions are so large, and the lack of current knowledge of a global minimum is so scarce, that it is acceptable to simply set an affordable number of function evaluations and let the algorithm exhaust them.

7. Evaluation of the PT method. In order to gauge the effectiveness of the potential transformation method we compare its performance to the performance of several other global optimization methods on two difficult test problems. The methods chosen for comparison are multistart (MS), Newtonian dynamics (ND), simulated annealing (SA), and SNIFR (SN). The first test problem we propose appears similar to, but is much more difficult than, a problem proposed originally by Griewank [17]. It has a single global minimum and a number of sub-optimal local minima growing exponentially with the dimension of the objective function. The second problem arises in molecular physics when one models the forces existing between nonbonded atoms in clusters or in protein molecules. Here the rate at which the number of local minima grows with dimensional is much faster than exponential, and the potential is highly singular. Algorithms for the chosen optimization methods are detailed in the form of pseudocode in the following subsection.

7.1. Problem I: The p-function. The first function we consider uses the p-norm of a vector,

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p},$$

where $p \geq 1$. Define the C^1-function $f : \mathbf{R}^n \to \mathbf{R}$ by

$$f(\mathbf{x}) = \frac{\|\mathbf{x}\|_p^2}{\lambda} + \|\sin \mathbf{x}\|_p^2, \tag{7.1}$$

where the sin of a vector is taken componentwise, $\lambda = 90$ and $-\frac{13\pi}{2} \leq x_i \leq \frac{13\pi}{2}$. We let the value of p vary with n, and determine it by choosing a constant ω (in this case, $\omega = 1.4$) and solving

$$\omega = n^{2/p} \tag{7.2}$$

for p. Some resulting values for p are shown in Table 1. The values of f along an axis

TABLE 1
Values of p.

n	2	5	8	10	20	50	100
p	4.12	9.57	12.36	13.69	17.81	23.25	27.37

and along a diagonal are given by

$$c_n(x) = f(x, 0, \ldots, 0) = \frac{x^2}{\lambda} + \sin^2 x \tag{7.3}$$

TABLE 2
Minimum values for $c(x)$ and $d(x)$.

Bin Number	0	1	2	3	4	5	6
Minimum of c_n	0.000	0.108	0.434	0.976	1.734	2.711	3.904
Minimum of d_n	0.000	0.152	0.607	1.367	2.429	3.796	5.465

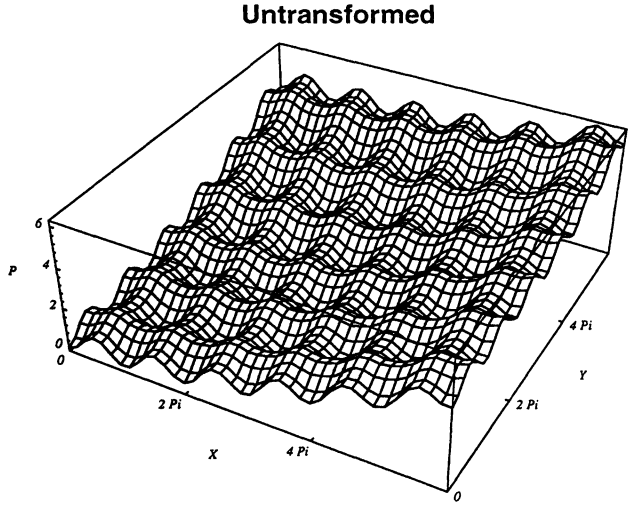


FIG. 1. *The p-function in 2-dim (untransformed).*

and

$$d_n(x) = f(x, \ldots, x) = n^{2/p}\left(\frac{x^2}{\lambda} + \sin^2 x\right) = \omega c(x). \tag{7.4}$$

These values are *independent of dimension*, which indicates that the shape of the function in high dimensions—and its difficulty as a global optimization test function—should be similar to that in low dimensions.

Except for the origin itself, the local minimizers of the p-function occur on or near the boundary of n-cubes centered at the origin and containing the minimizers of (7.3). There are thirteen such minimizers in the range $-\frac{13\pi}{2} \le x \le \frac{13\pi}{2}$, occurring near the minimizers of $\sin^2 x$, i.e., near multiples of π. Since (7.3) is symmetric about 0, there are only seven different minimum values, and six n-cubes. These values for c_n and the corresponding values for d_n are shown in Table 2. The local minima near the boundary of each of the six n-cubes are bounded by the values shown for each bin. Since the minimum value for d_n is smaller than the minimum value for c_n for the next bin, the n-cube for any minimizer can be identified from the pair of values of c_n and d_n in Table 2 that contain the value of f at the minimizer. Thus the minimizers are grouped into seven bins, $b_0, \ldots, b_6$, with only the origin in b_0. The bin number is a measure of how close the minimum is to the global minimum. A surface plot of the p-function in two dimensions is shown in Fig. 1.

7.2. Problem II: Microclusters. The second problem we consider is the determination of the ground-state of a system of unit-diameter spheres interacting via a pairwise-additive potential v. Let $\mathbf{r}_i$ denote the Cartesian coordinates of sphere i,

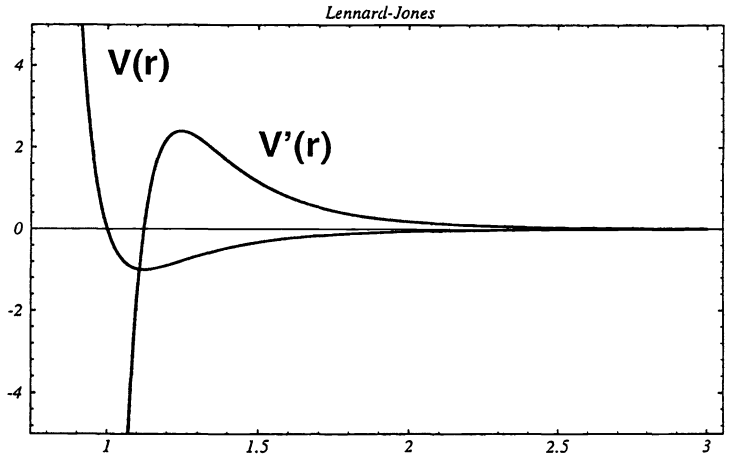


FIG. 2. *Lennard-Jones potential.*

and let $\mathbf{r}_{i,j}$ be the vector $\mathbf{r}_i - \mathbf{r}_j$. Then the Lennard-Jones potential [2] is defined by the formula

$$v(\mathbf{r}_{i,j}) = \|\mathbf{r}_{i,j}\|^{-12} - \|\mathbf{r}_{i,j}\|^{-6},$$

a sketch of which is shown in Fig. 2.

Note that the equilibrium interparticle separation occurs at $r = 2^{1/6}$, and repulsive forces dominate at shorter distances, while attractive forces dominate out to about $r = 3$. A system of N such spheres has the potential energy

$$f = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} v(\mathbf{r}_{i,j}).$$

Scientific interest in this problem stems from the fact that modern experiments reveal the existence of structural regularities in small clusters of atoms produced in supersonic beams [13]. Though the real structural problem is a many-body one, forces between neutral atoms are rather weak and short range. It is therefore expected that an additive two-body potential such as the Lennard-Jones can provide a tractable computational model for the atomic interactions.

The complexity of even so simple a model is daunting. The history of this problem and previous attempts to "solve" it have been reviewed in depth by Hoare and McInnes [19]. They exhaustively studied the problem for cluster sizes ranging from 6 to 13 spheres, for which they report the following number of potential-energy minima: 2; 4; 8; 18; 57; 145; 366; 988. One sees that the number of cluster configurations (local minima) rises much faster than linearly; in fact these authors estimate the number of distinct local minima by:

$$g(N) = \exp(-2.5176 + .3572N + .028N^2).$$

Thus for even a small cluster with 15 atoms one expects to find on the order of 10,750 local minima. A cluster containing 25 atoms is expected to have somewhere in the neighborhood of 10^{10} local minima! This explosive growth of cluster configurations

makes space covering, or other exhaustive techniques, quite impossible for all but the smallest clusters.

Finally, it is known (see [25] for a recent monograph) that the three-dimensional structure of proteins is likely largely determined by a myriad of weak forces of the van der Waals type, which can be effectively modeled (in part) using the Lennard-Jones potential. Since even a small protein can have of the order of 10^{40} potential-energy minima, the sphere-packing problem serves as a prototype for many of the structural problems of interest in the chemistry and biochemistry communities.

7.3. Pseudocode. In treating Problem I we compared the relative performance of MS, ND, and PT on the p-function in dimensions up to a maximum of N = 50. The smallest problem exhibits 3.7×10^5 local minima, while the largest has on the order of 10^{55} local minima. All algorithms were applied in a similar fashion, as shown by the pseudocode listed in Fig. 3. The algorithm to be applied is chosen by the call `SWITCH(method)`.

MS samples the objective function at a set of points randomly selected from a uniform distribution over the region of interest. Each step requires one function evaluation.

```
PROCEDURE MS
   xs = random variable
END PROCEDURE
```

ND and PT are methods based on integrating an ODE; they thus require an integration timestep h and integration method. Other system parameters are the total energy E_0 (ND and PT), gradient sensitivity ϵ (PT only), and target level c (PT only). Pseudocode for PT or MD then looks like the following:

```
PROCEDURE (ND, PT)
   SET-TARGET (for PT)
   GRAD(f, x)
   SCALE(ẋ)
   PROPAGATE(x ↦ xs)
END PROCEDURE
```

GRAD evaluates the gradient of the objective function. SCALE normalizes the velocity vector to enforce conservation of energy for ND or (4.10) for PT. PROPAGATE uses Euler's method to move the ODE one timestep forward. Target setting/resetting is required on each step; we will take this up momentarily.

Much more elaborate integration methods than Euler are clearly possible, but we have found Euler adequate to our needs as we do not pursue a goal of highly precise search trajectories.

The final procedure of importance is the target-setting routine used by PT. We prescribe a *constant target set slightly below the global minimum* in cases where this is known beforehand. In the case that the global minimum is not known in advance we have used the following procedure with success:

```
PROCEDURE SET-TARGET
   IF ((f − c) ≤ .1) THEN
   c = c − 1
```

Initialize:

```
max-trials = number of runs for a dimension
steps      = maximum iterations
method     = optimization method
wlen       = MIN(500,steps/10) = window length
wind       = steps/wlen = number of windows
c          = target level
epsilon    = gradient sensitivity
fmax       = maximum function value
h          = time-stepsize
xs         = starting       coordinate
fs         = FUNC(xs)       function value
gfs        = GRAD(f,xs)     gradient
FOR trial = 1 TO max-trials
   FOR w = 1 TO wind
      fw = fs
      xw = xs
      FOR s = 1 TO wlen
         SWITCH(method)
         fs = FUNC(xs)
         IF (fs ≤ fw) THEN
            fw = fs
            xw = xs
         ENDIF
      NEXT s
      OUTPUT(fw, xw)
   NEXT w
   fn = fw(1)
   FOR w = 1 TO wind
      LOCAL(xw, fw)
      IF (fw ≤ fn) THEN
         fn = fw
         xn = xw
      ENDIF
   NEXT w
   BIN(xn, fn)
NEXT trial
```

FIG. 3. FUNC *computes the objective function.* OUTPUT *stores coordinates for each window,* LOCAL *performs conjugate gradients,* BIN *assigns the bin number of the local minimum.*

END PROCEDURE

In Test Problem I (with a known global minimum at the origin) we chose $c = -0.2 =$ constant for each trial run. Target setting in Problem II was approached by setting the initial target level one unit lower than the initial function value. Application of the procedure above then kept the target level moving down as the algorithm discovered lower values of the objective function.

TABLE 3
MS, ND, PT *comparison.*

BIN:	0	1	2	3	4	5	6	Dim
MS:		20	30					5
ND:		15	31	4				
PT:	49	1						
MS:		1	32	17				10
ND:			3	20	23	4		
PT:	48	2						
MS:			3	27	20			20
ND:				1	19	26	4	
PT:	19	31						
MS:				4	44	2		30
ND:				2	4	34	10	
PT:	1	26	23					
MS:					18	32		50
ND:					3	28	19	
PT:			50					

In summary, each step of each of the computational methods we employ begins with a starting vector **x** and returns a new vector, **xs**, at which the objective function will be evaluated. The computational overhead of MS and SA is one function evaluation per step; that for ND and PT is one function and one gradient evaluation per step.

8. Results and discussion.

8.1. Problem I. We tested the optimization methods against Problem I in the following way. First, the dimension, N, of the problem was set and a number of steps chosen. In this study we took 32,000 steps (one function/gradient evaluation per step) in each of 50 trial runs, drawing the starting points from a uniform random distribution over the domain of interest ($|x_i| \leq \frac{13\pi}{2}$, $i = 1, 2, \ldots, N$). Each trial run was subdivided into windows: the lowest function value and associated vector in each window were stored for local minimization using conjugate gradients. Following local minimization, the lowest of all minima found on a given trial run was assigned a bin number as described above. Runs using MS, ND, and PT are summarized in Table 3.

PT used a gradient sensitivity $\epsilon = 1.$ and target level $c = -.2$ found by experiment with a few runs in dimension 10. A timestep of $h = 0.4$ was used by both PT and ND. $E_0 = f_{\max}$ was set at 30.

At first, 32,000 steps (function/gradient calculations) may seem an excessive number. However, recall Fig. 1, in which we display the p-function in two dimensions. The hilly terrain observed here is tame compared to the situation in higher dimensions. In 10 dimensions, for example, there are roughly 10^{11} local minima of the p-function over the domain we consider. The probability of finding a random point in bins $b_0, b_1, b_2, \ldots, b_j$ in d dimensions after s steps is

$$P_j(s,d) = 1 - \left(1 - \left(\frac{2j+1}{13}\right)^d\right)^s \approx s\left(\frac{2j+1}{13}\right)^d$$

or approximately $P_0 = 2 \times 10^{-7}$, $P_1 = .0137$, $P_2 = .8964$ after 32,000 steps. Thus MS performs as expected, while PT performs remarkably well. In 10 dimensions one observes that PT finds the global minimum fully 96% of the time, and is otherwise in bins 0 or 1 100% of the time. In 50 dimensions the p-function has roughly 5×10^{55}

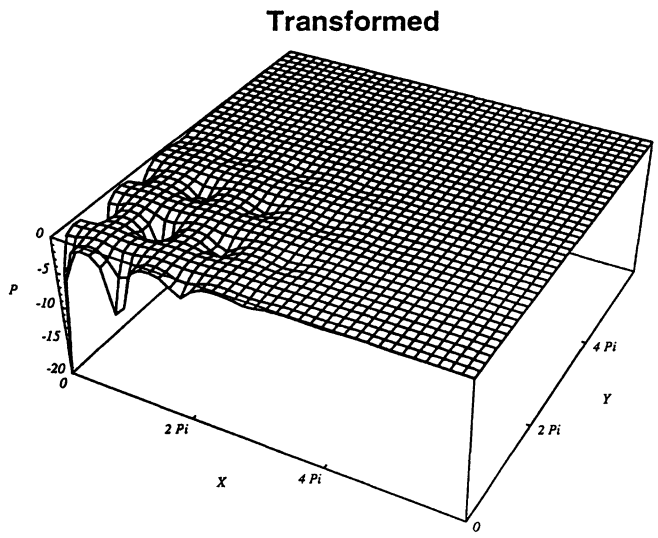


FIG. 4. *The transformed p-function.*

local minima, and the probability of finding a minimum in bins 0, 1, or 2 using a single run of 32,000 steps is roughly $P_2 = 6 \times 10^{-17}$. We observe that the PT method finds minima in bins 0, 1, or 2 100% of the time. By contrast, ordinary (i.e., *un*transformed) Newtonian dynamics performs poorly on the p-function. In fact, ND seems to be easily trapped by suboptimal minima, performing even less well as a global minimizer than MS.

Of course, PT is also based on Newtonian dynamics, and it, too, can be trapped by suboptimal minima, though with much lower frequency than one observes in Newtonian dynamics. The reasons for both the improvement in performance, and the residual difficulties, of the PT method are suggested by Fig. 4 in which we display the transformed potential in two dimensions for the current choice of system parameters ($\epsilon = 1, c = -.2, f_{\max} = 30$). Minima further out from the origin than those in bins $0, 1, 2$ have virtually disappeared under the effects of the potential transform, while those nearer the origin remain, and are deep. Thus, should the particle come near the origin, our data suggests that it remains trapped in one of the suboptimal minima for the remainder of the trajectory. Efforts to circumvent trapping by taking a target $c = 0$ were not successful in this initial study. More work in this area is needed, as will be reported in due course.

As a reference, we also applied the SA algorithm to this test problem, using the version published earlier in [11]. Initial acceptance frequencies of roughly 65% were gotten by starting at a temperature $T_0 = 1$, and a step length = 5. The annealing schedule was a customary one, $T_{s+1} = 0.95 \times T_s$, where we used 5,000 equilibration steps on sweep s, and a total of 40 sweeps overall. Thus the total number of function evaluations in the SA calculation was 200,000 per initial point, or roughly six times the number of function evaluations used in the trajectory calculations reported in Table 3. For this reason, the results are reported in a separate table, Table 4. We repeat the MS calculations of Table 3, extending them to 200,000 steps per initial point, to enable a comparison of the algorithms.

As expected, SA consistently outperforms MS in Table 4. But even with only a

TABLE 4
SA *results.*

BIN:	0	1	2	3	4	5	6	Dim
MS		43	7					5
SA	8	42						
MS		3	40	7				10
SA		40	10					
MS			2	40	8			20
SA		9	41					
MS				8	42			30
SA			47	3				
MS					26	24		50
SA				49	1			

TABLE 5
SNIFR *results.*

BIN:	0	1	2	3	4	5	6	Dim
SN	49	1						5
SN	50							10
SN	39	11						20
SN	5	45						30
SN		50						50

sixth of the number of function evaluations, PT consistently outperforms the popular annealing method. One reason for this is clear: gradient information is extremely valuable in the optimization of differentiable objective functions. Notwithstanding the strong theoretical results for the Gibb's probability distribution associated with SA, PT gets much lower, much faster. Clearly, SA is not the algorithm of choice when gradient information is available.

Finally, we include a comparison with the SNIFR algorithm discussed in §1.2. SNIFR is a discrete dynamical system, which relies on a maximum and minimum stepsize, rather than a timestep, in propagating the state vector. Parameters for SNIFR were chosen as in previous work [12], [30]: $\epsilon = 1$, maximum step length $\beta_{\max} = 1$, and scale factor $\mu = 0.25$. As in the PT calculations, we chose a fixed target, $c = -.2$, and took 32,000 steps from each of the 50 starting points. The pseudocode for the SNIFR is identical to that given for ND and PT, although the content of the subroutines is different. In Table 5 one sees that SNIFR almost always finds the global minimum for problems with dimensions ranging up to about $d = 20$. Beyond this its performance deteriorates, though by dimension 50 all minima are still in bins 0 or 1. In any case, SNIFR's performance is the best of any of the algorithms discussed here. Still, PT is not far behind, and it has the advantage of resting on a firm theoretical base, while SNIFR was devised as a discrete dynamical system mimicking the heuristics of Griewank's equation, and hence also of PT. The fact that PT is based on a modified Newtonian dynamics provides a powerful conceptual basis for understanding how it works and for seeing how to modify it in other situations, as in global optimization problems involving constraints [29]. SNIFR has not performed well in earlier tests involving constraints. It is also clear that SNIFR can be modified to impose an upper bound on its trajectories by making it mimic the general PT equation, rather than Griewank's special case.

8.2. Problem II. We next investigated the ability of PT to find minima which were at least as low as those found by previous authors. Extensive tabulations of

TABLE 6
Sphere-packing results.

14	N	ϵ	-Energy
	0		44.880
	∞		47.845
	1000	1.	47.845
16	0		55.907
	∞		56.816
	1,000	.9	53.757
	1,000	1.	52.807
	1,000	1.5	54.941
	2,000	.9	55.195
	2,000	1.	54.909
	2,000	1.5	54.941
	5,000	.9	56.816
	5,000	1.	56.816
	5,000	1.5	56.816
23	0		89.696
	∞		92.844
	5,000	.8	90.265
	5,000	.9	92.844
	5,000	1.	91.198
	5,000	1.1	90.984
24	0		91.194
	∞		97.349
	5,000	.9	96.239
	5,000	1.	96.517
	10,000	.9	97.349
	10,000	1.	96.517
25	∞		102.37
	10,000	.9	96.533
	30,000	.9	101.08
	10,000	1.	99.528
	30,000	1.	102.37
	30,000	1.5	101.08

results for the energies of ground-state microclusters of Lennard-Jones atoms have been assembled by Hoare and McInnes in [19] and by Northby in [26]. In particular, we were interested to know how the PT method performed as a function of the number of steps taken, and as a function of the gradient sensitivity ϵ.

We chose a timestep size appropriate to a typical molecular dynamics calculation, $h = .001$, and used the target setting/resetting procedure described in §7.3. For each cluster we chose initial coordinates for each sphere randomly on a cube of edge-length three units (remember that the potential is negligible beyond three units). Starting coordinates which would have produced overlapping spheres were discarded, due to the strongly repulsive behavior of v near the origin (refer to Fig. 1). A computational window of length of 500 steps was chosen; as in the previous computations the minimum found in each window was saved for polishing by conjugate gradients.

Results of our calculations are summarized in Table 6. Entries labeled $N = 0$ refer to a local minimum obtained by direct application of conjugate gradients to the starting point. Local minima reported as global minima are labeled by $N = \infty$ in the table. We report PT results as functions of the number of steps, N, and the sensitivity value ϵ.

Note that we find the global minima in all cases we studied. Interestingly, the number of function/gradient evaluations is quite small considering the complexity (i.e.,

the number of local minima) of the objective function. The largest system we studied contains 25 particles (72 independent variables), for which we obtain the global minimum with only 30,000 function/gradient calls. Results for the 16-atom cluster show clearly that slight changes in the gradient sensitivity can have a strong influence on the minima obtained. Note especially that increasing ϵ does not always produce lower minima: increasing this parameter too much can lead to undesirable "orbiting" of local minima (see [30] for interesting pictures regarding this phenomenon). Reducing ϵ too much has the effect of reducing sensitivity of the dynamics to the local gradient.

These calculations show that one should not rely too heavily on precisely chosen values of the system parameters. Rather, choose a stepsize that does not produce too drastic a variation in the objective function, begin with gradient sensitivities near $\epsilon = 1$, and employ a simple target setting routine.

9. Conclusion. In this paper, we have introduced the potential transformation algorithm for large-scale global optimization. The algorithm results in a second-order ODE which is shown to contain Griewank's ODE as a special case. However, our implementation of the algorithm is quite different, since our experience has indicated that the use of a coarsely discretized ODE solver is more efficient for the problem of global optimization than the use of accurate solvers with small stepsizes. Moreover, we have found that setting the target level too close to the global minimum results in speeds that are too low for efficient coverage of the region of interest. Consequently, we recommend maintaining a target level somewhat below the current estimate of the global minimum.

Although we have used potential transformations in this paper only in conjunction with Newtonian dynamics and the numerical technique of reparameterization, the idea of transforming the objective function in this way before applying a minimization algorithm is an independent one, which might find application in other methods.

Of course, no single trajectory or finite number of trajectories can be guaranteed to provide the global minimum of any problem in high dimensions. Indeed, as previous authors have observed, the problem of finding the global minimum of a function is inherently unsolvable. Törn and Žilinskas [35, p. 7] observe that, "From a practical point of view, the problem may be stated also in a different way: There exists a goal (e.g., to find as small a value of $f(\cdot)$ as possible), there exist resources (e.g., some number of trials), and the problem is how to use these resources in an optimal way." The PT algorithm performs well in these initial tests. The method has a strong theoretical foundation and can easily be extended to equations of motion with constraints. Applications of the constrained algorithm to problems in protein structure are now underway, and will be reported in the near future. Computer programs that perform these calculations are available on request from the second author.

REFERENCES

[1] F. Alluffi-Pentini, V. Parisi, and F. Zirilli, *Global optimization and stochastic differential equations*, J. Optim. Theory Appl., 47 (1985), pp. 1–16.

[2] J. A. Barker and D. Henderson, *What is "Liquid"? Understanding the states of matter*, Rev. Mod. Phys., 48 (1976), pp. 587–671.

[3] R. W. Becker and G. V. Lago, *A global optimization algorithm*, in Proceedings of the 8th Allerton Conference on Circuits and Systems Theory, 1970.

[4] H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. DiNola, and J. R. Haak, *Molecular dynamics with coupling to an external bath*, J. Chem. Phys., 81 (1984), pp. 3684–3690.

[5] R. A. R. BUTLER AND E. E. SLAMINKA, *An evaluation of the Sniffer global optimization algorithm using standard test functions*, J. Comput. Phys., 99 (1992), pp. 28–32.
[6] R. CAR AND M. PARINELLO, *Unified approach for molecular dynamics and density-functional theory*, Phys. Rev. Lett., 55 (1985), pp. 2471–2474.
[7] G. CHANG, W. C. GUIDA, AND W. C. STILL, *An internal coordinate Monte Carlo method for searching conformational space*, J. Amer. Chem. Soc., 111 (1989), pp. 4379–4386.
[8] R. M. COOKE, T. S. HARVEY, AND I. D. CAMPBELL, *Solution structure of human insulin-like growth factor* I. *A nuclear magnetic resonance and restrained molecular dynamics study*, Biochemistry, 30 (1991), pp. 5484–5491.
[9] L. C. W. DIXON AND G. P. SZEGÖ, EDS., *Towards Global Optimization, Volume* 2, Elsevier, New York, 1978.
[10] R. A. DONNELLY, *Generalized descent for geometry optimization*, manuscript, 1987.
[11] ———, *Geometry optimization by simulated annealing*, Chem. Phys. Lett., 136 (1987), pp. 274–278.
[12] R. A. DONNELLY AND J. W. ROGERS, JR., *A discrete search technique for global optimization*, Internat. J. Quantum Chem., S22 (1988), pp. 507–513.
[13] O. ECHT, K. SATTLER, AND E. RECTRAGEL, *Magic numbers for sphere packing: Experimental verification for free xenon clusters*, Phys. Rev. Lett., 47 (1991), pp. 1121–1128.
[14] D. M. FERGUSON AND D. J. RABER, *A new approach to probing conformational space with molecular mechanics: Random incremental pulse search*, J. Amer. Chem. Soc., 111 (1989), pp. 4371–4378.
[15] R. FUSCO, L. CACCIANOTTI, AND C. TOSI, *New methods to look for the most stable conformations of a molecule*, Il Nuovo Cimento, 8 (1986), pp. 211–218.
[16] H. GOLDSTEIN, *Classical Mechanics*, 2nd ed., Addison-Wesley, Reading, MA, 1980.
[17] A. O. GRIEWANK, *Generalized descent for global optimization*, J. Optim. Theory Appl., 34 (1981), pp. 11–39.
[18] T. S. HARVEY, A. J. WILKINSON, R. M. COOKE, AND I. D. CAMPBELL, *The solution structure of human transforming growth factor* α, Eur. J. Biochem., 198 (1991), pp. 555–562.
[19] M. HOARE AND J. MCINNES, *Morphology and statistical statics of simple microclusters*, Adv. Phys., 32 (1983), p. 791.
[20] D. HOHL, R. O. JONES, R. CAR, AND M. PARINELLO, *Energy surfaces and structure of* S_7O, J. Amer. Chem. Soc., 111 (1989), pp. 825–828.
[21] S. INCERTI, V. PARISI, AND F. ZIRILLI, *A new method for solving nonlinear simultaneous equations*, SIAM J. Numer. Anal., 16 (1979), pp. 779–789.
[22] S. KIRKPATRICK, G. D. GELATT, JR., AND M. P. VECCHI, *Optimization by simulated annealing*, Science, 220 (1983), pp. 671–680.
[23] A. V. LEVY AND A. MONTALVO, *The tunnelling algorithm for the global minimization of functions*, SIAM J. Statist. Comput., 6 (1985), pp. 15–29.
[24] Z. LI AND H. A. SCHERAGA, *Monte Carlo approach to the multiple-minima problem in protein folding*, Proc. Natl. Acad. Sci. U.S.A., 84 (1987), p. 6611.
[25] J. A. MCCAMMON AND S. C. HARVEY, *Dynamics of Proteins and Nucleic Acids*, Cambridge University Press, New York, 1987.
[26] J. NORTHBY, *Structure and binding of Lennard-Jones clusters:* $13 \leq n \leq 147$, J. Chem. Phys., 87 (1987), pp. 6166–6177.
[27] L. PIELA, J. KOSTROWICKI, AND H. A. SCHERAGA, *On the multiple-minima problem in the conformational analysis of molecules. Deformation of the potential energy hypersurface by the diffusion equation method*, J. Phys. Chem., 93 (1989), pp. 3339–3346.
[28] E. O. PURISMA AND H. A. SCHERAGA, *An approach to the multiple-minima problem in protein folding by relaxing dimensionality*, J. Molec. Biol., 697 (1987), p. 196.
[29] J. W. ROGERS, JR., *Potential transformation methods for large-scale constrained global optimization*, 1993, submitted.
[30] J. W. ROGERS, JR. AND R. A. DONNELLY, *A search technique for global optimization in a chaotic environment*, J. Optim. Theory Appl., 61 (1989), pp. 111–121.
[31] M. SAUNDERS, *Stochastic exploration of molecular mechanics energy surfaces. Hunting for the global minimum*, J. Amer. Chem. Soc., 109 (1987), pp. 3150–3152.
[32] E. E. SLAMINKA AND K. D. WOERNER, *Central configurations and a theorem of Palmore*, Celest. Mech. Dynam. Astron., 48 (1990), pp. 347–355.
[33] J. A. SYNMAN AND L. P. FATTI, *A multi-start global minimization algorithm with dynamic search trajectories*, J. Optim. Theory Appl., 54 (1987), pp. 121–141.
[34] G. T. TIMMER, *Global Optimization: A Stochastic Approach*, Ph.D. thesis, Erasmus University Rotterdam, Centrum voor Wiskunde en Informatica, Amsterdam, 1984.
[35] A. TÖRN AND A. ŽILINSKAS, *Global Optimization*, Springer-Verlag, Berlin, 1989.

[36] A. A. TÖRN, *A search clustering approach to global optimization*, in Towards Global Optimization, Volume 2, L. C. W. Dixon and G. P. Szegö, eds., Elsevier, New York, 1978.

SIAM J. OPTIMIZATION
Vol. 5, No. 4, pp. 892–913, November 1995

009

EXISTENCE AND REGULARITY OF SOLUTIONS TO A VARIATIONAL PROBLEM OF MUMFORD AND SHAH: A CONSTRUCTIVE APPROACH*

YANG WANG†

Abstract. We study a variational problem arising in the approach of Mumford and Shah to the image segmentation problem of computer vision. Given $f \in L^\infty(D)$ for a domain D in $\mathbf{R}^2$, the simplified Mumford–Shah energy associated to a decomposition $D = \Omega_1 \cup \cdots \cup \Omega_N$ is

$$\mathbf{E_0}[\Gamma, \alpha] = \sum_{i=1}^{N} \int_{\Omega_i} (f(x) - c_{\Omega_i})^2 \, dx + \alpha|\Gamma|,$$

where $\alpha > 0$ is a constant, c_{Ω_i}, is the average of $f(x)$ on Ω_i, and where $|\Gamma|$ is the length of the boundary of the regions Ω_i not in ∂D. Mumford and Shah showed, using geometric measure theory, that for a continuous f a minimizing Γ^* exists that is piecewise C^2. We prove this result constructively, and also extend it to show for general bounded measurable f that a minimizer exists. Furthermore, we prove that every minimizer must be piecewise $C^{1,1}$. Our approach is to study $\mathbf{E_0}[\Gamma, \alpha]$ on the class of piecewise linear Γ.

Key words. image segmentaion, Mumford–Shah energy, minimizer, Hausdorff metric

AMS subject classification. 49JXX

1. Introduction.

The segmentation problem in computer vision is the problem of subdividing an image into regions in such a way that in each region, the image is relatively uniform. Mumford and Shah [10] proposed to do this by minimizing energy functionals that encode penalty measures for properties of a good segmentation. Let $f \in L^\infty(D)$, where D is a domain in $\mathbf{R}^2$, represent the light intensity of the image. A decomposition of D is

$$D = \Omega_1 \cup \Omega_2 \cup \cdots \cup \Omega_N,$$

where each region Ω_i is closed and has a boundary $\partial\Omega_i$ that is piecewise C^1. Let $\Gamma = \cup_i \overline{\partial\Omega_i \setminus \partial D}$ be the boundary of the segmentation. Mumford and Shah propose to find such a decomposition and an approximating function $u(x)$ by minimizing the *Mumford–Shah energy*

$$\mathbf{E}[u, \Gamma, \mu, \nu] = \mu^2 \int_D \big(u(x) - f(x)\big)^2 \, dx + \sum_i \int_{\Omega_i} |\nabla u|^2 \, dx + \nu^2|\Gamma|, \tag{1}$$

where $\mu, \nu > 0$ are constants (weight parameters) and $|\Gamma|$ is the length of Γ. In addition to this energy functional, Mumford and Shah introduced a simplified functional obtained by letting μ, $\nu \to 0$ and $\mu^2/\nu^2 \to \alpha > 0$. Then $\nabla u \equiv 0$, and the *simplified Mumford–Shah energy* is

$$\mathbf{E_0}[\Gamma, \alpha] = \sum_{i=1}^{N} \int_{\Omega_i} \big(f(x) - c_{\Omega_i}\big)^2 \, dx + \alpha|\Gamma|, \tag{2}$$

where c_{Ω_i} is the average of $f(x)$ over Ω_i. It is this simplified Mumford–Shah energy functional that is the subject of this paper.

* Received by the editors July 7, 1993; accepted for publication (in revised form) June 3, 1994.
† School of Mathematics, Georgia Institute of Technology, Atlanta, Georgia 30332 (wang@math.gatech.edu).

Mumford and Shah [10] proved, using methods from geometric measure theory, that if $f(x)$ is continuous on D then there exists a solution Γ^* to

$$\mathbf{E_0}[\Gamma^*, \alpha] = \inf_{\Gamma} \mathbf{E_0}[\Gamma, \alpha],$$

where Γ^* is piecewise C^2. In this paper, we present a constructive approach to finding minimizers of $\mathbf{E_0}[\Gamma, \alpha]$, that is based on studying $\mathbf{E_0}[\Gamma, \alpha]$ over piecewise linear boundaries Γ. Using it we rederive the existence result of Mumford and Shah, and we show, more generally, that for any bounded measurable $f \in L^\infty(D)$ there exists a minimizer Γ^* that is piecewise C^1. Furthermore we show that for *every* minimizer Γ^* it must satisfy a *weak curvature bound*: the unit tangent vector of γ (parametrized by arc length) of any C^1-segment of Γ^* satisfies a Lipschitz condition where the Lipschitz constant depends only on D, α, and $\max_D |f(x)|$. The constructive nature of our approach is in obtaining such Γ^* as a suitable limit of piecewise linear Γ_n's, and constraints on the behavior of the Γ_n's, e.g., angles between segments. In principle it is possible to develop a computer implementation of this approach.

Now reconsider the general Mumford–Shah problem (1). It is much harder. Mumford and Shah conjectured that there is a minimizing solution to

$$\mathbf{E}[u^*, \Gamma^*, \mu, \nu] = \inf_{u,\Gamma} \mathbf{E}[u, \Gamma, \mu, \nu],$$

where Γ^* is piecewise C^1 and $u^* \in W^{1,2}(D \setminus \Gamma^*)$, but this conjecture has never been proved. Existence results have been achieved for a weaker problem where Γ is only required to be a relatively closed set and $|\Gamma|$ is replaced by Hausdorff one-dimensional measure; cf. [8], [7]. Such an existence result has recently been obtained [1], [5]. Shah [12] obtained some results on a one–dimensional simplification of the problem, and Richardson [11] obtains asymptotic information on solutions as $\mu \to \infty$. Both of these authors use a geometric measure theory approach relying heavily on existence theorems. The elementary constructive approach of this paper offers a potentially promising approach to some of the questions.

2. Basic results. Let $\mathcal{C} = \{A \subset \mathbf{R}^2 \mid A \text{ is compact }\}$. The *Hausdorff metric* on $\mathcal{C}$ is defined as

$$d_H(A, B) = \sup_{x \in A} \inf_{y \in B} |x - y| + \sup_{x \in B} \inf_{y \in A} |x - y|$$

for A, $B \in \mathcal{C}$. It is easy to show that d_H is indeed a metric on $\mathcal{C}$. The following are well-established facts (see [6]).

PROPOSITION 2.1. 1. *$(\mathcal{C}, d_H)$ is a complete metric space.*

2. *Let $\{A_i\} \subset \mathcal{C}$ and $A_1 \supseteq A_2 \supseteq A_3 \ldots$. Then*

$$\lim_{i \to \infty} A_i = \bigcap_{i=1}^{\infty} A_i$$

in the metric space $(\mathcal{C}, d_H)$.

3. *Suppose $\{A_i\}$ is a sequence in $\mathcal{C}$ and $\lim_{n \to \infty} A_n = A$ in $(\mathcal{C}, d_H)$. Then*

$$A = \bigcap_{n=1}^{\infty} \overline{\left(\bigcup_{i=n}^{\infty} A_i \right)}.$$

4. *Let D be any compact subset of $\mathbf{R}^2$ and $\mathcal{C}_D = \{A \subseteq D \mid A \text{ is closed in } D\}$. Then $\mathcal{C}_D$ is a compact subset of $\mathcal{C}$ in $(\mathcal{C}, d_H)$.*

For simplicity we shall write $\mathbf{E_0}[\Gamma]$ in place of $\mathbf{E_0}[\Gamma, \alpha]$ from now on. Given any Γ and $D \setminus \Gamma = \bigcup_i \Omega_i$, where each Ω_i is a connected component of $D \setminus \Gamma$, we can separate the energy

$$\mathbf{E_0}[\Gamma] = \sum_i \int_{\Omega_i} (f - c_{\Omega_i})^2 dx + \alpha|\Gamma|$$

into two parts:

$$\text{square energy:} \quad \mathbf{E_S}[\Gamma] = \sum_i \int_{\Omega_i} (f - c_{\Omega_i})^2 dx;$$

$$\text{length energy:} \quad \mathbf{E_L}[\Gamma] = \alpha|\Gamma|.$$

Recall that c_{Ω_i} denotes the average of $f(x)$ over Ω_i. Notice that the square energy $\mathbf{E_S}[.]$ can be defined for any closed subset $A \subseteq D$.

LEMMA 2.2. *Let $\{A_i\}_{i>0} \subset \mathcal{C}_D$. If $A_1 \supseteq A_2 \supseteq A_3 \supseteq \cdots, \{A_i\}_{i>0} \subset \mathcal{C}_D$ and $A = \cap_{i=1}^{\infty} A_i$, then*

$$\lim_{i \to \infty} \mathbf{E_S}[A_i] = \mathbf{E_S}[A].$$

Proof. For any compact subset $B \subseteq D$ and $(x, y) \in D \times D$, define

$$\chi_B(x, y) = \begin{cases} 0 & \text{if } x \in B \text{ or } y \in B, \\ 0 & \text{if } x \text{ and } y \text{ do not belong to the same} \\ & \text{connected component of } D \setminus B, \\ 1 & \text{otherwise.} \end{cases}$$

$\chi_B : D \times D \longrightarrow \mathbf{R}$ is measurable and $|\chi_B| \leq 1$. It is easy to see that for any $(x, y) \in D \times D$, $\lim_{i \to \infty} \chi_{A_i}(x, y) = \chi_A(x, y)$.

Note that for any $B \subset \mathcal{C}_D$,

$$\mathbf{E_S}[B] = \int_D (f(x) - g_B(x))^2 dx,$$

where

$$g_B(x) = \begin{cases} f(x) & \text{if } x \in B, \\ \dfrac{\displaystyle\int_D \chi_B(x, y) f(y) dy}{\displaystyle\int_D \chi_B(x, y) dy} & \text{otherwise.} \end{cases}$$

Since $\lim_{i \to \infty} \chi_{A_i} = \chi_A$, by the Lebesgue Dominated Theorem,

$$\lim_{i \to \infty} \int_D \chi_{A_i}(x, y)\, dy = \int_D \chi_A(x, y)\, dy, \quad \text{and}$$

$$\lim_{i \to \infty} \int_D \chi_{A_i}(x, y) f(y)\, dy = \int_D \chi_A(x, y) f(y)\, dy$$

for any $x \in D$; hence

$$\lim_{i\to\infty} g_{A_i}(x) = g_A(x).$$

Applying the Lebesgue Dominated Theorem again we obtain

$$\begin{aligned}\lim_{i\to\infty} \mathbf{E_S}[A_i] &= \lim_{i\to\infty} \int_D (f(x) - g_{A_i}(x))^2 dx \\ &= \int_D (f(x) - g_A(x))^2 dx \\ &= \mathbf{E_S}[A]. \qquad \square\end{aligned}$$

PROPOSITION 2.3 (Lower semicontinuity). *If* $\{A_i\} \subset \mathcal{C}_D$ *and* $\lim_{i\to\infty} A_i = A \in \mathcal{C}_D$ *in* $(\mathcal{C}_D, d_H)$, *then*

$$\liminf_{i\to\infty} \mathbf{E_S}[A_i] \geq \mathbf{E_S}[A].$$

Proof. For any $B_1, B_2 \in \mathcal{C}_D$ such that $B_1 \subset B_2$, $\mathbf{E_S}[B_1] \geq \mathbf{E_S}[B_2]$. Let $B_n = \overline{\bigcup_{i=n}^{\infty} A_i}$. Then $B_1 \supseteq B_2 \supseteq B_3 \supseteq \cdots$. Hence

$$A = \bigcap_{i=1}^{\infty} B_i = \lim_{i\to\infty} B_i.$$

By Lemma 2.2,

$$\lim_{n\to\infty} \mathbf{E_S}[B_n] = \mathbf{E_S}[A].$$

Since $B_n \supseteq A_m$ for $m \geq n$, $\mathbf{E_S}[B_n] \leq \mathbf{E_S}[A_n]$; thus

$$\liminf_{n\to\infty} \mathbf{E_S}[A_n] \geq \liminf_{n\to\infty} \mathbf{E_S}[B_n] = \mathbf{E_S}[A]. \qquad \square$$

3. Properties of the segmentation. In this section, we examine the segmentations of the domain D by using *piecewise linear* line segments. For simplicity we restrict our discussions to the domain $D = [0, L] \times [0, L]$. When a piecewise linear Γ gives a *locally minimal* $\mathbf{E_0}[\Gamma]$, meaning that $\mathbf{E_0}[\Gamma]$ cannot be reduced through a small perturbation of Γ (in the topology induced by the Hausdorff metric), there are some restrictions as to the number of edges, regions, etc.

We devote essentially the entire section to prove the following two facts: if a piecewise linear Γ is a local minimum of the energy $\mathbf{E_0}[.]$ then (i) the number of connected components in $D \setminus \Gamma$ must be bounded by some constant depending only on D, α and $\max_D |f|$, and so are the numbers of edges and junctions (see Definition 3.2) in Γ (Proposition 3.8); (ii) the angle between any two adjacent linear segments must be sufficiently close to π (Proposition 3.12).

These two facts are central to our main existence and regularity results, which we prove by obtaining a piecewise C^1 global minimizer from a sequence of locally minimizing piecewise linear Γ_j's.

DEFINITION 3.1. $\Gamma \in \mathcal{C}_D$ *is called* piecewise linear *if*

$$\Gamma = \bigcup_{i=1}^{N} l_i$$

for some finite collection $\{l_i\}_{i=1}^N$, *where each* $l_i \subset D$ *is a linear segment, no* l_i *is part of the boundary* ∂D, *and for any two different* $i, j \leq N$, l_i *and* l_j *intersect at most a common endpoint of* l_i *and* l_j. $(l_1, l_2, \ldots, l_N)$ *is called a* piecewise linear representation *of* Γ.

Note that the angle between any two adjacent segments is allowed to be π. For each piecewise linear Γ there are infinitely many ways of choosing l_i's, and hence there are infinitely many different piecewise linear representations. Denote

$$\mathbf{SL}(D) = \left\{\Gamma \in \mathcal{C}_D \;\middle|\; \Gamma \text{ is piecewise linear}\right\}.$$

Before any further discussion we define the following terms related to Γ.

DEFINITION 3.2. *Let* $\Gamma \in \mathbf{SL}(D)$ *and let* $(l_i, l_2, \ldots l_N)$ *be a piecewise linear representation of* Γ. *Define the following terms related to* Γ *and its piecewise linear representation.*

region. *A region defined by* Γ *is a connected component of* $D \setminus \Gamma$.

node. *A node of* Γ *is an endpoint of any linear segment* l_i.

junction. *A junction of* Γ *is a node that satisfies any of the following:*

1. *It is on* ∂D; *or*
2. *it connects to at least* 3 *linear segments*; *or*
3. *it connects to only one linear segment (tip of a crack).*

edge. *An edge of* Γ *is defined as the closure of any connected component of* $\Gamma \setminus \{$*the junctions of* $\Gamma\}$.

edge-element. *An edge-element of* Γ *is a linear segment* l_i.

boundary-component. *A boundary-component defined by* Γ *is the closure of any connected component of* $\partial D \setminus \{$*the junctions of* $\Gamma\}$.

For any representation of Γ, we also define

$$\begin{aligned}
R(\Gamma) &= \text{Number of regions defined by } \Gamma,\\
n(\Gamma) &= \text{Number of nodes in the representation of } \Gamma,\\
J(\Gamma) &= \text{Number of junctions in } \Gamma,\\
E(\Gamma) &= \text{Number of edges in } \Gamma,\\
e(\Gamma) &= \text{Number of edge-elements in the representation of } \Gamma,\\
B(\Gamma) &= \text{Number of boundary-components defined by } \Gamma.
\end{aligned}$$

Note that among all the terms defined, only nodes, edge-elements, $n(\Gamma)$, and $e(\Gamma)$ actually depend on the piecewise linear representation of Γ.

DEFINITION 3.3. *Let* $\Gamma \in \mathbf{SL}(D)$. *An edge* γ *in* Γ *is called a* crack *if the two sides of* γ *belong to the same connected component of* $D \setminus \Gamma$.

PROPOSITION 3.4. *Let* $\Gamma \in \mathbf{SL}(D)$ *and* Γ^* *be derived from* Γ *by removing all cracks in* Γ. *Then* $\mathbf{E_0}[\Gamma^*] \leq \mathbf{E_0}[\Gamma]$. *The equality* $\mathbf{E_0}[\Gamma^*] = \mathbf{E_0}[\Gamma]$ *holds only when* $\Gamma^* = \Gamma$, *i.e.,* Γ *is crack-free. Hence*

$$\inf_{\Gamma \in \mathbf{SL}(D)} \mathbf{E_0}[\Gamma] = \inf_{\substack{\Gamma \in \mathbf{SL}(D)\\ \Gamma \text{ is crack-free}}} \mathbf{E_0}[\Gamma].$$

Proof. Notice that $\mathbf{E_S}(\Gamma^*) = \mathbf{E_S}(\Gamma)$, and obviously $\mathbf{E_L}(\Gamma^*) \leq \mathbf{E_L}(\Gamma)$ with the equality being true only when $\Gamma^* = \Gamma$. □

LEMMA 3.5 (Euler). *For any crack-free* $\Gamma \in \mathbf{SL}(D)$, *an edge* γ *of* Γ *is called a* simple loop *if it contains no junction. Let* $l(\Gamma)$ *be the number of edges of* Γ *that are*

simple loops and $c(\Gamma)$ *be the number of connected components of* $\Gamma \cup \partial D$. *Then*

$$E(\Gamma) + B(\Gamma) - R(\Gamma) - J(\Gamma) = -c(\Gamma) + l(\Gamma) + \delta(\Gamma),$$
$$e(\Gamma) + B(\Gamma) - R(\Gamma) - n(\Gamma) = -c(\Gamma) + \delta(\Gamma),$$

where $\delta(\Gamma) = 1$ *if there is no junction on* ∂D, *and* $\delta(\Gamma) = 0$ *otherwise.*

Proof. The theorem of Euler states that if a simply connected domain is divided into some simply connected subdomains, then

$$e - r - n = -1,$$

where e is the number of edges, r is the number of regions, and n is the number of nodes.

Notice that in our case, the definition of junctions is slightly different from the definition of nodes in the theorem of Euler . In our definition, whenever an edge forms a closed loop, we do not consider that there is a junction on the edge. In the theorem of Euler, however, such an edge is considered to contain one node. Thus, if $c(\Gamma) = 1$, then

$$e = E(\Gamma) + B(\Gamma),$$
$$r = R(\Gamma),$$
$$n = J(\Gamma) + \delta(\Gamma).$$

Therefore,

$$E(\Gamma) + B(\Gamma) - R(\Gamma) - J(\Gamma) = -1 + \delta(\Gamma).$$

If $c(\Gamma) > 1$, applying the theorem of Euler to each connected component of $\Gamma \cup \partial D$. Summing the equalities up, we have

$$E(\Gamma) + B(\Gamma) - R(\Gamma) - J(\Gamma) = -c(\Gamma) + l(\Gamma) + \delta(\Gamma).$$

For the second part of our lemma, compensating for $l(\Gamma)$ is not needed because a line segment cannot form a closed loop. Hence

$$e(\Gamma) + B(\Gamma) - R(\Gamma) - n(\Gamma) = -c(\Gamma) + \delta(\Gamma). \qquad \square$$

Let $A(\Omega)$ denote the area of Ω for any region Ω.

LEMMA 3.6. *Let* $\Omega \subset D$ *be any connected piecewise* C^1 *domain. Let* $\partial\Omega = E \cup B$, *where* $B = \partial\Omega \cap \partial D$ *and* $E = \overline{\partial\Omega \setminus B}$. *Suppose* $A(\Omega) \leq A(D)/2$. *Then*

$$|E| \geq \frac{1}{3}|B|.$$

Proof. Recall that $D = [0, L] \times [0, L]$. We consider two cases. If $|E| \geq L$, then it follows from the isoperimetric inequality that

$$|E| + 4L - |B| = |\partial(D \setminus \Omega)| \geq \sqrt{4\pi A(D \setminus \Omega)} \geq \sqrt{2\pi}L.$$

Hence $3|E| > (4 - \sqrt{2\pi})L + |E| \geq B$.

Suppose $|E| < L$. Then each connected component of E either does not intersect ∂D, or the two intersecting points lie on the same side of ∂D or two adjacent sides

of ∂D. It is easy to see that whatever happens, we always have $|E| \geq |B|/\sqrt{2} \geq |B|/3$. □

LEMMA 3.7. *Let $\Gamma \in \mathbf{SL}(D)$ and Ω be a region defined by Γ. If $\gamma \subseteq \partial\Omega$ is an edge of Γ such that $|\gamma| > a_0 A(\Omega)$, where $a_0 = 4\max_D |f|^2/\alpha$, then*

$$\mathbf{E_0}[\Gamma] > \mathbf{E_0}[\overline{\Gamma \setminus \gamma}].$$

Proof. If γ is a crack, then obviously

$$\mathbf{E_0}[\Gamma] > \mathbf{E_0}[\overline{\Gamma \setminus \gamma}].$$

So we assume that γ is not a crack. Thus γ separates two different regions Ω and Ω^*. Let c_Ω, c_{Ω^*}, and $c_{\Omega\cup\Omega^*}$ be the average of $f(x)$ over Ω, Ω^*, and $\Omega \cup \Omega^*$, respectively. Then,

$$\begin{aligned}
&\mathbf{E_0}[\Gamma] - \mathbf{E_0}[\overline{\Gamma \setminus \gamma}] \\
&= \int_\Omega (f - c_\Omega)^2 dx + \int_{\Omega^*} (f - c_{\Omega^*})^2 dx - \int_{\Omega\cup\Omega^*} (f - c_{\Omega\cup\Omega^*})^2 dx + \alpha|\gamma| \\
&\geq \int_\Omega (f - c_\Omega)^2 dx + \int_{\Omega^*} (f - c_{\Omega^*})^2 dx - \int_{\Omega\cup\Omega^*} (f - c_{\Omega^*})^2 dx + \alpha|\gamma| \\
&= \int_\Omega \left\{(f - c_\Omega)^2 - (f - c_{\Omega^*})^2\right\} dx + \alpha|\gamma| \\
&\geq -4\max_\Omega |f|^2 A(\Omega) + \alpha|\gamma| \;>\; 0.
\end{aligned}$$

Therefore, $\mathbf{E_0}[\Gamma] > \mathbf{E_0}[\overline{\Gamma \setminus \gamma}]$. □

PROPOSITION 3.8. *There exists a constant $K_0 = K_0(D, \alpha, \max|f|) > 0$ such that for any $\Gamma \in \mathbf{SL}(D)$, if $R(\Gamma) + E(\Gamma) + B(\Gamma) + J(\Gamma) > K_0$, then $\mathbf{E_0}[\overline{\Gamma \setminus \gamma}] < \mathbf{E_0}[\Gamma]$ for some edge γ of Γ.*

Proof. Let $\Gamma \in \mathbf{SL}(D)$ and Ω be a region defined by Γ with $A(\Omega) < A(D)/2$. Then $|E| \geq |B|/3$ where $\partial\Omega = E \cup B$ as in Lemma 3.6. Hence

$$|E| \geq \frac{1}{4}(|E| + |B|) = \frac{1}{4}|\partial\Omega| \geq \frac{1}{4}\sqrt{4\pi A(\Omega)} = \frac{1}{2}\sqrt{\pi A(\Omega)}.$$

Choose $0 < b < A(D)/2$ so that for all $0 < \varepsilon \leq b$,

$$\frac{1}{10}\left(\frac{1}{2}\sqrt{\pi\varepsilon}\right) > a_0\,\varepsilon,$$

where $a_0 = 4\max_D |f|^2/\alpha$. It follows from Lemma 3.7 that if $A(\Omega) = \varepsilon \leq b$ and γ is an edge of Γ, then $\mathbf{E_0}[\overline{\Gamma \setminus \gamma}] < \mathbf{E_0}[\Gamma]$ whenever $|\gamma| \geq \sqrt{\pi\varepsilon}/20$.

Let $\mathcal{A} \subset \mathbf{SL}(D)$ denote the set of Γ's such that $\mathbf{E_0}[\overline{\Gamma \setminus \gamma}] \geq \mathbf{E_0}[\Gamma]$ for any edge γ of Γ. For any $\Gamma \in \mathcal{A}$ the above implies that given any region Ω defined by Γ with $A(\Omega) \leq b$ and $\partial\Omega = E \cup B$, the edge part E comprises at least 10 edges of Γ. Let R_+ be the number of regions defined by Γ which have area $> b$. Obviously $R_+ < A(D)/b$. Since each edge corresponds to only two regions while except for those with area $> b$ each region corresponds to at least ten edges, we have

$$E(\Gamma) \geq \frac{10}{2}(R(\Gamma) - R_+) \geq 5R(\Gamma) - \frac{5A(D)}{b}.$$

Applying the same argument to junctions, namely, each junction corresponds to at least three edges or boundaries, while each edge or boundary corresponds to at most two junctions, we have

$$E(\Gamma) + B(\Gamma) \geq \frac{3}{2} J(\Gamma).$$

Combining the two inequalities, we obtain

$$\begin{aligned} & E(\Gamma) + B(\Gamma) - R(\Gamma) - J(\Gamma) \\ &= \frac{1}{3} E(\Gamma) + \frac{1}{3} B(\Gamma) - R(\Gamma) + \frac{2}{3} \left(E(\Gamma) + B(\Gamma) - \frac{3}{2} J(\Gamma) \right) \\ &\geq \frac{1}{3} E(\Gamma) - R(\Gamma) \\ &\geq \frac{2}{3} R(\Gamma) - \frac{5A(D)}{3b}. \end{aligned}$$

On the other hand, Lemma 3.5 gives us

$$E(\Gamma) + B(\Gamma) - R(\Gamma) - J(\Gamma) = -c(\Gamma) + l(\Gamma) + \delta(\Gamma),$$

where $l(\Gamma)$ is the number of edges that are simple loops in Γ. Since any region defined by Γ cannot be enclosed by such a simple loop if the area of the region is $\leq b$, it implies $l(\Gamma) \leq A(D)/b$ and therefore

$$E(\Gamma) + B(\Gamma) - R(\Gamma) - J(\Gamma) \leq 1 + A(D)/b.$$

Hence $R(\Gamma) \leq C_R$, where $C_R = 8A(D)/b + 3/2$.

The rest follows easily. For any $\Gamma \in \mathcal{A}$, since $E(\Gamma) + B(\Gamma) \geq \frac{3}{2} J(\Gamma)$ we have

$$\begin{aligned} \frac{1}{3} \left(E(\Gamma) + B(\Gamma) \right) &\leq E(\Gamma) + B(\Gamma) - J(\Gamma) \\ &= \left(E(\Gamma) + B(\Gamma) - J(\Gamma) - R(\Gamma) \right) + R(\Gamma) \\ &\leq -1 + \frac{A(D)}{b} + R(\Gamma). \end{aligned}$$

Hence both $E(\Gamma)$ and $B(\Gamma)$ are uniformly bounded. So $J(\Gamma) \leq \frac{2}{3}(E(\Gamma) + B(\Gamma))$ must also be uniformly bounded. This proves the proposition. □

COROLLARY 3.9. *Let*

$$\mathbf{SL}_0(D) = \left\{ \Gamma \in \mathbf{SL}(D) \;\middle|\; R(\Gamma) + E(\Gamma) + B(\Gamma) + J(\Gamma) \leq K_0 \right\}.$$

Then

$$\inf_{\Gamma \in \mathbf{SL}(D)} \mathbf{E_0}[\Gamma] = \inf_{\Gamma \in \mathbf{SL}_0(D)} \mathbf{E_0}[\Gamma].$$

We now introduce a new subset of $\mathbf{SL}(D)$. Let $\mathbf{SL}(D, m, \varepsilon) \subset \mathbf{SL}(D)$ denote the set of Γ's, which have a piecewise linear representation such that $n(\Gamma) \leq m$ and $|e| \leq \varepsilon$ for any edge-element e of Γ. We have the following lemma.

LEMMA 3.10. $\mathbf{SL}(D, m, \varepsilon) \subset \mathcal{C}_D$ *is a compact subset in* $\{\mathcal{C}, d_H\}$ *for any* $m > 0$ *and* $\varepsilon > 0$.

Proof. We establish a bound for $e(\Gamma) + B(\Gamma)$ for $\Gamma \in \mathbf{SL}(D, m, \varepsilon)$. Since except for those that contain at least one corner of D, each region corresponds to at least three edge-elements or boundaries while each edge-element or boundary corresponds to no more than two regions, so

$$e(\Gamma) + B(\Gamma) \geq \frac{3}{2}\Big(R(\Gamma) - 4\Big).$$

Hence,

$$\frac{1}{3}\Big(e(\Gamma) + B(\Gamma)\Big) \leq e(\Gamma) + B(\Gamma) - R(\Gamma) + 4 = n(\Gamma) - c(\Gamma) + \delta(\Gamma) + 4 \leq n(\Gamma) + 4.$$

Therefore $e(\Gamma)+B(\Gamma) \leq 3m+12$. We now conclude the compactness of $\mathbf{SL}(D, m, \varepsilon)$ by showing that the limit of $\{\Gamma_i\}$, where $\Gamma_i \in \mathbf{SL}(D, m, \varepsilon)$, must also be in $\mathbf{SL}(D, m, \varepsilon)$. Let

$$\Gamma_i = \bigcup_{j=1}^{n_i} e_j^i,$$

where e_j^i is an edge-element and $e_j^i \cap e_k^i$ for any $j \neq k$ is either empty or a node in Γ_i. Because n_i are bounded for all i, we may without loss of generality assume that $n_i = n^*$ for all i, or we may replace the sequence Γ_i by a subsequence.

Choose a subsequence Γ_{k_1}, Γ_{k_2}, $\Gamma_{k_3}, \ldots$ of $\{\Gamma_i\}_{i>0}$ such that for all $j \leq n^*$

$$\lim_{i\to\infty} e_j^{k_i} = e_j^*.$$

Then for any $j \leq n^*$ either e_j^* is a linear segment with $|e_j^*| \leq \varepsilon$ or a single point.

Now let Γ^* be the limit of $\{\Gamma_i\}$. Then $\Gamma^* = \bigcup_{j=1}^{n^*} e_j^*$. It is clear that $\Gamma^* \in \mathbf{SL}(D)$. Notice that if $e_{j_0}^*$ is a single point for some j_0 then we still have $\Gamma^* = \bigcup_{j\neq j_0} e_j^*$. So we may without loss of generality assume that all e_j^* are line segments. If (e_j^*) is a piecewise linear representation of Γ^* then we have $\Gamma^* \in \mathbf{SL}(D, m, \varepsilon)$. Suppose (e_j^*) is not a piecewise linear representation of Γ. Then the following must occur: for some $i \neq j$, a node of e_j^* may lie in the interior of some e_i^*, or $e_j^* \cap e_i^*$ may be a linear segment itself. However, given either of the above cases we can always subdivide $e_i^* \cup e_j^*$ into smaller edge-elements without adding any *new nodes.* So by this procedure we obtain a piecewise linear representation of Γ. Since no new nodes are added, all nodes in this representation are limit points of nodes of in Γ_i, so $n(\Gamma) \leq m$ and $\Gamma \in \mathbf{SL}(D, m, \varepsilon)$. This implies the compactness of $\mathbf{SL}(D, m, \varepsilon)$. □

PROPOSITION 3.11. *For any $m > 0$ and $\varepsilon > 0$, there exists a $\Gamma^* \in \mathbf{SL}(D, m, \varepsilon) \cap \mathbf{SL}_0(D)$ such that*

$$\mathbf{E_0}[\Gamma^*] = \inf_{\Gamma\in\mathbf{SL}(D,m,\varepsilon)} \mathbf{E_0}[\Gamma].$$

Proof. Notice that for any sequence $\{\Gamma_n\} \subset \mathbf{SL}(D, m, \varepsilon)$ such that $\Gamma_n \to \Gamma$, we have $\mathbf{E_L}(\Gamma) \leq \liminf_n \mathbf{E_L}(\Gamma_n)$. The existence of $\Gamma^* \in \mathbf{SL}(D, m, \varepsilon)$ follows immediately from the compactness of $\mathbf{SL}(D, m, \varepsilon)$ and the lower semicontinuity of the energy $\mathbf{E_S}(\Gamma)$. $\Gamma^* \in \mathbf{SL}_0(D)$ follows from Proposition 3.8. □

PROPOSITION 3.12. *Suppose $\Gamma_0 \in \mathbf{SL}(D, m, \varepsilon)$ and e_2, e_2 are any two adjacent edge-elements of Γ_0 that intersect at a nonjunction node. Let $0 \leq \theta \leq \pi$ be the angle between e_1 and e_2. If*

$$|\pi - \theta| > M_0(|e_1| + |e_2|),$$

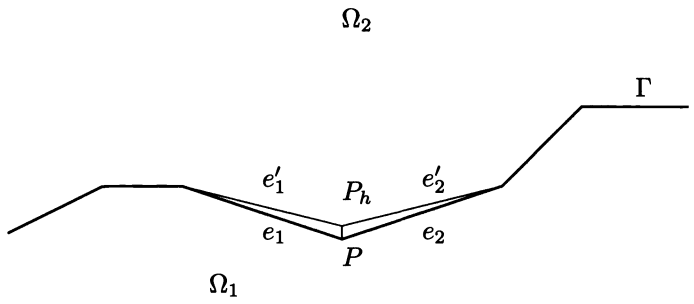


FIG. 1. *A small perturbation of* Γ.

where $M_0 = 8 \max_D |f|^2/\alpha$, *then*

$$\mathbf{E_0}[\Gamma_0] > \inf_{\Gamma \in \mathbf{SL}(D,m,\varepsilon)} \mathbf{E_0}[\Gamma].$$

Proof. Suppose that e_1 and e_2 intersect at the node P. Let Ω_1 and Ω_2 be the regions separated by the edge containing e_1 and e_2, as illustrated in Fig. 1.

Consider a new $\Gamma^h \in \mathbf{SL}(D,m,\varepsilon)$, which is obtained from Γ_0 by slightly perturbing e_1 and e_2, also shown in Fig. 1. As Γ_0 becomes Γ^h, the node P becomes P_h so that the line segment PP_h satisfies $|PP_h| = h$ and it bisects the angle θ. Let the domain formed by the polygon $e_1 e_2 e_2' e_1'$ be Ω_h. Then

$$\begin{aligned}
&\mathbf{E_0}[\Gamma^h] - \mathbf{E_0}[\Gamma_0] \\
&= \int_{\Omega_1 \cup \Omega_h} (f - c_{\Omega_1 \cup \Omega_h})^2 dx + \int_{\Omega_2 \setminus \Omega_h} (f - c_{\Omega_2 \setminus \Omega_h})^2 dx + \alpha\Big(|e_1'| + |e_2'|\Big) \\
&\quad - \int_{\Omega_1} (f - c_{\Omega_1})^2 dx - \int_{\Omega_2} (f - c_{\Omega_2})^2 dx - \alpha\Big(|e_1| + |e_2|\Big).
\end{aligned}$$

(As usual, for any domain $\Omega \subset D$, c_Ω is the mean value of $f(x)$ over Ω.) Elementary trigonometry gives

$$A(\Omega_h) = \Big(|e_1| + |e_2|\Big) h \sin\frac{\theta}{2} \quad \text{and} \quad |e_i| = |e_i| - h\cos\frac{\theta}{2} + o(h), \qquad i = 1,2.$$

Because

$$A(\Omega_1 \cup \Omega_h) c_{\Omega_1 \cup \Omega_h} - A(\Omega_1) c_{\Omega_1} = \int_{\Omega_1 \cup \Omega_h} f dx - \int_{\Omega_1} f dx = O(A(\Omega_h)),$$

it is immediate that $c_{\Omega_1 \cup \Omega_h} - c_{\Omega_1} = O(A(\Omega_h))$. Similarly, $c_{\Omega_2 \setminus \Omega_h} - c_{\Omega_2} = O(A(\Omega_h))$. We have therefore

$$\begin{aligned}
&\int_{\Omega_1 \cup \Omega_h} (f - c_{\Omega_1 \cup \Omega_h})^2 dx - \int_{\Omega_1} (f - c_{\Omega_1})^2 dx \\
&\qquad = \int_{\Omega_1 \cup \Omega_h} (f - c_{\Omega_1} + c_{\Omega_1} - c_{\Omega_1 \cup \Omega_h})^2 dx - \int_{\Omega_1} (f - c_{\Omega_1})^2 dx \\
&\qquad = \int_{\Omega_h} (f - c_{\Omega_1 \cup \Omega_h})^2 dx + \Big(c_{\Omega_1} - c_{\Omega_1 \cup \Omega_h}\Big)^2 A(\Omega_1)
\end{aligned}$$

$$\leq 4 \sup_D |f|^2 A(\Omega_h) + o(h),$$

$$\int_{\Omega_2 \setminus \Omega_h} (f - c_{\Omega_2 \setminus \Omega_h})^2 dx - \int_{\Omega_2} (f - c_{\Omega_2})^2 dx$$
$$\leq 4 \sup_D |f|^2 A(\Omega_h) + o(h).$$

Hence,

$$\begin{aligned}
\mathbf{E_0}[\Gamma^h] - \mathbf{E_0}[\Gamma_0] &\leq 8 \sup_D |f|^2 A(\Omega_h) - 2\alpha h \cos\frac{\theta}{2} + o(h) \\
&= 8 \sup_D |f|^2 \Big(|e_1| + |e_2|\Big) h \sin\frac{\theta}{2} - 2\alpha h \cos\frac{\theta}{2} + o(h) \\
&= \alpha M_0 \Big(|e_1| + |e_2|\Big) h \sin\frac{\theta}{2} - 2\alpha h \cos\frac{\theta}{2} + o(h).
\end{aligned}$$

Let $M_0 = 8 \sup_D |f|^2/\alpha$. If $|\pi - \theta| > M_0(|e_1| + |e_2|)$, then

$$\begin{aligned}
&\limsup_{h \to 0^+} \frac{\mathbf{E_0}[\Gamma^h] - \mathbf{E_0}[\Gamma_0]}{h} \\
&\leq \alpha M_0 \Big(|e_1| + |e_2|\Big) \sin\frac{\theta}{2} - 2\alpha \cos\frac{\theta}{2} \\
&= \alpha \sin\frac{\theta}{2} \left\{ M_0 \alpha \Big(|e_1| + |e_2|\Big) - 2\cot\frac{\theta}{2} \right\} \\
&< \alpha \sin\frac{\theta}{2} \left\{ |\pi - \theta| - 2\tan\frac{\pi - \theta}{2} \right\} \\
&\leq \alpha \sin\frac{\theta}{2} \left\{ |\pi - \theta| - 2\left|\frac{\pi - \theta}{2}\right| \right\} \\
&= 0.
\end{aligned}$$

Therefore $\mathbf{E_0}[\Gamma^h] < \mathbf{E_0}[\Gamma_0]$ for sufficiently small $h > 0$. □

4. Approximation. In this section we will be looking at more general segmentations of D, namely, those formed by *piecewise* C^1 Γ. The key idea in this section is to show that for any locally minimizing piecewise linear Γ, the restriction on the angle between any two adjacent linear segments of Γ stated in Proposition 3.12 implies that each edge of Γ can be approximated well by a $C^{1,1}$ curve. Using this fact we prove our existence result (Proposition 4.6).

We call a curve γ a *simple* C^k curve if there exists a C^k map $c : [0,1] \longrightarrow \mathbf{R}^2$ with $c'(t) \neq 0$ for all $t \in [0,1]$ such that $c(t_1) \neq c(t_2)$ for any $t_1 \in [0,1]$, $t_2 \in (0,1)$ and $t_1 \neq t_2$.

DEFINITION 4.1. $\Gamma \in \mathcal{C}_D$ *is called* piecewise C^1 *if*

$$\Gamma = \bigcup_{i=1}^{N} \gamma_i$$

for some finite collection $\{\gamma_i\}_{i=1}^N$, *where each* $\gamma_i \subset D$ *is a simple* C^1 *curve and for any* i *and* $j \neq i$, *both* $\gamma_i \cap \partial D$ *and* $\gamma_j \cap \gamma_j$ *are either empty or contain one or two endpoints of* γ_i. $(\gamma_1, \gamma_2, \ldots, \gamma_N)$ *is called a* piecewise C^1 *representation of* Γ.

Denote

$$\mathbf{S}^1(D) = \left\{\Gamma \in \mathcal{C}_D \;\middle|\; \Gamma \text{ is piecewise } C^1\right\}.$$

It is obvious that $\mathbf{SL}(D) \subset \mathbf{S}^1(D)$. We have the following generalization of Definition 3.2.

DEFINITION 4.2. *Let $\Gamma \in \mathbf{S}^1(D)$. We define the following terms related to Γ.*

region. *A region defined by Γ is a connected component of $D \setminus \Gamma$.*

junction. *A junction of Γ is a point in D where some γ_i and ∂D meet, or where at least three different γ_i's meet.*

edge. *An edge of Γ is defined as the closure of any connected component of $\Gamma \setminus \{$junctions of $\Gamma\}$.*

boundary. *A boundary defined by Γ is the closure of a connected component of $\partial D \setminus \{$junctions of $\Gamma\}$.*

Notice that none of the terms defined above depend on the representation of Γ. The following are also independent of the representation of Γ:

$$R(\Gamma) = \text{Number of regions defined by } \Gamma;$$
$$J(\Gamma) = \text{Number of junctions in } \Gamma;$$
$$E(\Gamma) = \text{Number of edges in } \Gamma;$$
$$B(\Gamma) = \text{Number of boundaries defined by } \Gamma.$$

LEMMA 4.3. *It holds that*

$$\inf_{\Gamma \in \mathbf{S}^1(D)} \mathbf{E_0}[\Gamma] = \inf_{\Gamma \in \mathbf{SL}(D)} \mathbf{E_0}[\Gamma].$$

Proof. It is obvious that

$$\inf_{\Gamma \in \mathbf{S}^1(D)} \mathbf{E_0}[\Gamma] \leq \inf_{\Gamma \in \mathbf{SL}(D)} \mathbf{E_0}[\Gamma].$$

But since any $\Gamma \in \mathbf{S}^1(D)$ can be approximated to arbitrary degree of accuracy, it is easy to see that $\inf_{\Gamma \in \mathbf{S}^1(D)} \mathbf{E_0}[\Gamma] \geq \inf_{\Gamma \in \mathbf{SL}(D)} \mathbf{E_0}[\Gamma]$. □

LEMMA 4.4. *For any $\varepsilon > 0$,*

$$\inf_{\Gamma \in \mathbf{SL}(D)} \mathbf{E_0}[\Gamma] = \lim_{m \to \infty} \inf_{\Gamma \in \mathbf{SL}(D,m,\varepsilon)} \mathbf{E_0}[\Gamma].$$

Proof. Notice that any linear segment can be broken up and viewed as the union of linear segments of length $\leq \varepsilon$. Therefore,

$$\mathbf{SL}(D) = \bigcup_{m=1}^{\infty} \mathbf{SL}(D, m, \varepsilon) = \lim_{m \to \infty} \mathbf{SL}(D, m, \varepsilon).$$

The lemma follows immediately. □

Lemmas 4.3 and 4.4 indicate that in order to minimize $\mathbf{E_0}[\Gamma]$ for $\Gamma \in \mathbf{S}^1(D)$, we can first minimize $\mathbf{E_0}[\Gamma]$ over $\mathbf{SL}(D, m, \varepsilon) \cap \mathbf{SL}_0(D)$ and consider the limit of $\mathbf{E_0}[\Gamma]$ as $m \to \infty$.

LEMMA 4.5. *Let $f(t) : [a, b) \longrightarrow \mathbf{R}^2$ be a piecewise constant map such that $f(t) = X_i$ for $t \in [t_i, t_{i+1})$, where $a = t_0 < t_1 < \cdots < t_n = b$. Suppose $|t_{i+1} - t_i| \leq \varepsilon$ for any $0 \leq i < n$ and there exists a constant M such that $|X_{i+1} - X_i| \leq M(|t_{i+2} - t_i|)$ for any $0 \leq i < n - 1$. Then there is a $F(t) : [a, b) \longrightarrow \mathbf{R}^2$, which has the following properties:*

1. $F(t)$ *is* C^1.
2. $|F'(t)| \le 6M$.
3. $|F(t) - f(t)| \le 8M\varepsilon$.
4. $\int_a^b F(t)dt = \int_a^b f(t)dt$.

Proof. Consider

$$a = \bar{t}_0 < \bar{t}_1 < \cdots < \bar{t}_n = b,$$

where for any $0 < i < n$, $\bar{t}_i = (t_i + t_{i+1})/2$. Let $F_1(t)$ be a cubic spline approximation of $f(t)$ on $[a, b]$ defined as follows: for $t \in [\bar{t}_i, \bar{t}_{i+1})$, where $0 \le i < n-1$,

$$F_1(t) = \frac{X_i - X_{i+1}}{(\bar{t}_{i+1} - \bar{t}_i)^3}\Big(2(t - \bar{t}_i)^3 - 3(\bar{t}_{i+1} - \bar{t}_i)(t - \bar{t}_i)^2\Big) + X_i,$$

and for $t \in [\bar{t}_{n-1}, \bar{t}_n]$

$$F_1(t) = X_{n-1}.$$

Clearly, $F_1(\bar{t}_i) = X_i$ and $F_1'(\bar{t}_i) = 0$ for any $i < n$; hence $F_1(t)$ is C^1.

For any $t \in [a, b]$, if $t \in [t_i, t_{i+1})$, then either $f(t) = X_i$ or $f(t) = X_{i+1}$. Notice that if $t \in [t_i, t_{i+1})$ then

$$0 \le 3(\bar{t}_{i+1} - \bar{t}_i)(t - \bar{t}_i)^2 - 2(t - \bar{t}_i)^3 \le (\bar{t}_{i+1} - \bar{t}_i)^3.$$

Hence we have

$$\begin{aligned}
|F_1(t) - f(t)| &\le |F_1(t) - X_i| + |X_i - f(t)| \\
&\le \frac{|X_i - X_{i+1}|}{(\bar{t}_{i+1} - \bar{t}_i)^3} \cdot \Big|2(t - \bar{t}_i)^3 - 3(\bar{t}_{i+1} - \bar{t}_i)(t - \bar{t}_i)^2\Big| \\
&\quad + |X_i - X_{i+1}| \\
&\le |X_i - X_{i+1}| + |X_i - X_{i+1}| \\
&\le 2|X_i - X_{i+1}| \\
&\le 2M|t_i - t_{i+2}| \\
&\le 4M\varepsilon,
\end{aligned}$$

and

$$\begin{aligned}
|F_1'(t)| &= \frac{|X_i - X_{i+1}|}{(\bar{t}_{i+1} - \bar{t}_i)^3} \cdot \Big|6(t - \bar{t}_i)^2 - 6(\bar{t}_{i+1} - \bar{t}_i)(t - \bar{t}_i)\Big| \\
&\le \frac{|X_i - X_{i+1}|}{(\bar{t}_{i+1} - \bar{t}_i)^3} \cdot |(t - \bar{t}_i)(\bar{t}_{i+1} - t)| \\
&\le \frac{2 \cdot 3M|t_i - t_{i+2}|}{|t_{i+2} + t_{i+1} - t_{i+1} + t_i|} \\
&= 6M.
\end{aligned}$$

Let $F(t) = F_1(t) + \delta$ where

$$\delta = \frac{\int_a^b \big(f(t) - F_1(t)\big)dt}{b - a};$$

then $F'(t) = F_1'(t)$ and

$$\begin{aligned} |F(t) - f(t)| &\le |F(t) - F_1(t)| + |F_1(t) - f(t)| \\ &\le |\delta| + 4M\varepsilon \\ &\le \frac{\int_a^b |f(t) - F_1(t)|dt}{b-a} + 4M\varepsilon \\ &\le 8M\varepsilon. \end{aligned}$$

Hence $F(t)$ satisfies the listed properties. □

PROPOSITION 4.6. *Let $\{\Gamma_i\}_{i>0}$ be a sequence in $\mathcal{C}_D$ such that $\Gamma_i \in \mathbf{SL}(D, m_i, \varepsilon_i)$, where $\lim_{i\to\infty} m_i = \infty$ and $\lim_{i\to\infty} \varepsilon_i = 0$. Suppose*

$$\mathbf{E_0}[\Gamma_i] = \inf_{\Gamma\in\mathbf{SL}(D,m_i,\varepsilon_i)} \mathbf{E_0}[\Gamma].$$

Then there exists a $\Gamma^* \in \mathcal{C}_D$, which is limit point of $\{\Gamma_i\}_{i>0}$ such that Γ^* satisfies the following properties.

1. *It holds that*

$$\Gamma^* = \bigcup_{j=1}^{N^*} \gamma_j,$$

with $N^* \le 2K_0$, where $K_0 = K_0(D, \alpha, \max|f|)$ is defined in Proposition 3.8 and each γ_j is a simple C^1 curve.

2. *For any $j \le N^*$ let $T_j(s)$ denote the unit tangent vector of γ_j parametrized by the arc length s of γ_j. Then,*

$$|T_j(s_1) - T_j(s_2)| \le C_0|s_1 - s_2|,$$

where $C_0 = C_0(D, \alpha, \max|f|)$ is a constant.

3. *For any i and $j \ne i$, both $\gamma_i \cap \partial D$ and $\gamma_i \cap \gamma_j^*$ are either empty or contain some endpoints of γ_i. Hence $\Gamma^* \in \mathbf{S}^1(D)$.*

4.

$$\mathbf{E_0}[\Gamma^*] = \inf_{\Gamma\in\mathbf{S}^1(D)} \mathbf{E_0}[\Gamma].$$

Proof. Let $\Gamma_i = \bigcup_{j=1}^{m_i} \gamma_j^i$ where γ_j^i are the edges of Γ_i. According to Corollary 3.9, since $m_i \le \mathbf{K_0}$, we may without loss of generality assume that $m_i = N_0$ because we can always find an $N_0 \le K_0$ such that there are infinitely many i's for which $m_i = N_0$.

Now, fix a j and consider the family $\{\gamma_j^i\}_{i>0}$. Define f_i by

$$\gamma_j^i(s) = \gamma_j^i(0) + \int_0^s f_i(t)dt,$$

where s is the arc length parameter of γ_j^i and $0 \le s \le l_i^j = |\gamma_j^i|$. Since γ_j^i is piecewise linear, f_i is piecewise constant. Because Γ_i minimizes $\mathbf{E_0}[\Gamma]$ in $\mathbf{SL}(D, m_i, \varepsilon_i)$, Proposition 3.12 implies that $f_i(t)$ satisfies the conditions stated in Lemma 4.5 for some constant $M = M(D, \alpha, \max|f|)$. Hence there is a C^1 function $F_i(t)$ defined on $[0, l_i^j]$ such that

$$|F_i(t) - f_i(t)| \le 8M\varepsilon_i, \qquad |F_i'(t)| \le 6M, \quad \text{and}$$

$$\int_0^{l_i^j} f_i(t)dt = \int_0^{l_i^j} F_i(t)dt.$$

For a given i let $\hat{\gamma}_j^i$ be the parametrized curve

$$\hat{\gamma}_j^i(s) = \gamma_j^i(0) + \int_0^s F_i(t)dt$$

for $0 \le s \le l_i^j$. Since $|F_i'(t)|$ is a uniformly bounded sequence of functions, there exists a subsequence $\{\hat{\gamma}_j^{n_i}\}_{i>0}$ of $\{\hat{\gamma}_j^i\}_{i>0}$ that converges uniformly to some $\gamma_j^*(s)$ which is either C^1 or in the degenerate case, a single point. Let $\Gamma^* = \cup_{j=1}^{N_0} \gamma_j^*$. If $\gamma_{j_0}^*$ is a single point for some j_0, then we still have $\Gamma^* = \cup_{j \neq j_0} \gamma_j^*$. So without loss of generality we may assume that γ_j^* is not a single point for all j, and that $\lim_{i\to\infty} \gamma_j^i = \gamma_j^*$.

We now prove that $\Gamma^* \in \mathbf{S}^1(D)$ by showing that it has a piecewise C^1 representation. If (γ_j^*) is a piecewise C^1 representation of Γ^* then we are done. Suppose it is not a piecewise C^1 representation of Γ^*. Then there must be some $n \neq m$ such that the set $\gamma_n^* \cap \gamma_m^*$ contains a point x that is not an endpoint of both γ_n^* and γ_m^*, or there is an $x \in \gamma_n^* \cap \partial D$ such that x is not an endpoint of γ_n^* for some n. We show that in the former case x must be an endpoint of either γ_n^* or γ_m^*. If not, since $\gamma_n^i \cap \gamma_m^i$ for any i contains only endpoints of both γ_n^i and γ_m^i, γ_n^* and γ_m^* must be tangent to each other at x (they cannot cross each other at x, otherwise γ_n^i and γ_m^i will intersect for sufficiently large i). For any $a > 0$, let $\delta = \delta(a) = d_H(\gamma_n^* \cap B_a(x), \gamma_m^* \cap B_a(x))$.

Since γ_n^* and γ_m^* are C^1 and tangent to each other at x, we can make δ/a arbitrarily small by choosing a sufficiently small $a > 0$.

Let p_1, p_2 be the endpoints of $B_a(x) \cap \gamma_n^*$ and q_1, q_2 of $B_a(x) \cap \gamma_m^*$. Since p_1, p_2, q_1, and q_2 are all on $\partial B_a(x)$, we assume that on $\partial B_a(x)$, q_1 is in between p_1 and q_2, while q_2 is in between p_2 and q_1. This is shown in Fig. 2. Consider p_1^i, $p_2^i \in \gamma_n^i$, and q_1^i, $q_2^i \in \gamma_m^i$ which satisfy

$$\lim_{i\to\infty} p_1^i = p_1, \qquad \lim_{i\to\infty} p_2^i = p_2$$

and

$$\lim_{i\to\infty} q_1^i = q_1, \qquad \lim_{i\to\infty} q_2^i = q_2.$$

Let

$$\hat{\Gamma}_i = \Gamma_i \cup \{\text{line segments } p_1^i q_1^i \text{ and } p_2^i q_2^i\};$$

$\hat{\Gamma}_i \in \mathbf{SL}(D, m_i + n_i, \varepsilon_i)$ for some $n_i > 0$. Denote the portion of γ_n^i between p_1^i and p_2^i by ψ_n^i and the portion of γ_m^i between q_1^i and q_2^i by ψ_m^i. Let Ω be the region enclosed by ψ_n^i, $p_2^i q_2^i$, ψ_m^i, and $p_1^i q_1^i$. Since for sufficiently large i,

$$d_H(\psi_n^i, \psi_m^i) \le 2\delta\,, \quad |\psi_n^i| \le 3a, \quad \text{and} \quad |\psi_m^i| \le 3a,$$

we have

$$A(\Omega_i) \le 2\delta \max\{\psi_n^i, \psi_m^i\} \le 6\delta a.$$

Therefore,

$$\left|\mathbf{E}_{\mathbf{S}}[\hat{\Gamma}_i \setminus \psi_n^i] - \mathbf{E}_{\mathbf{S}}[\Gamma_i]\right| \le C_1 A(\Omega_i) \le 6C_1\delta a$$

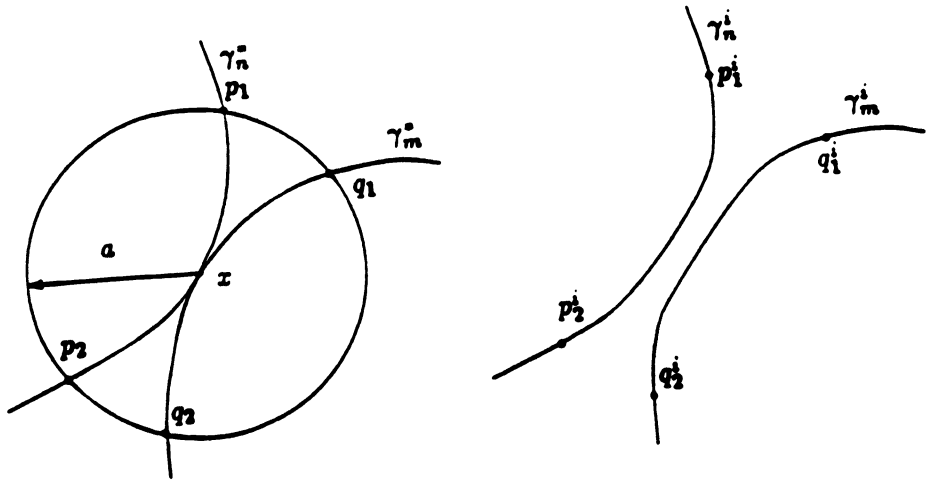


FIG. 2. *A case in which γ_n^* is tangent to γ_m^*.*

and

$$\mathbf{E_0}[\hat{\Gamma}_i \setminus \psi_n^i] - \mathbf{E_0}[\Gamma_i] \leq 6C_1\delta a + 2\delta - |\psi_n^i| \leq 6C_1\delta a + 2\delta - a.$$

Since δ/a can be made arbitrarily small by choosing a sufficiently small $a > 0$, we can choose an $a > 0$ such that

$$6C_1\delta a + 2\delta - a \leq -\frac{a}{2}.$$

Thus for i sufficiently large,

$$\mathbf{E_0}[\hat{\Gamma}_i \setminus \psi_n^i] \leq \mathbf{E_0}[\Gamma_i] - \frac{a}{2}.$$

But

$$\lim_{i\to\infty} \mathbf{E_0}[\Gamma_i] = \inf_{\Gamma\in\mathbf{S}^1(D)} \mathbf{E_0}[\Gamma] \leq \inf_{\Gamma\in\mathbf{SL}(D,m,\varepsilon)} \mathbf{E_0}[\Gamma]$$

for any $N_0 > 0$ and $\varepsilon > 0$. This is a contradiction. Hence any $x \in \gamma_n^* \cap \gamma_m^*$ must be an endpoint of either γ_n^* or γ_m^*. The same argument shows that any $x \in \gamma_n^* \cap \partial D$ must be an endpoint of γ_n^*, and that if γ_n^* has a self-intersection at x then x must be an endpoint of γ_n^*.

So we may now refine each γ_n^* into $\gamma_n^* = \cup\gamma_{n,k}^*$ such that each endpoint of $\gamma_{n,k}^*$ is and endpoint of some γ_m^*, and that every $x \in \gamma_{n,k}^* \cap \gamma_{m,l}^*$ must be an endpoint of both $\gamma_{n,k}^*$ and $\gamma_{m,l}^*$. Each $\gamma_{n,k}^*$ is simple. Furthermore, since the total number of endpoints of all γ_n^* is bounded by K_0, the number of $\gamma_{n,k}^*$ is bounded by $2K_0$. So $(\gamma_{n,k}^*)$ is a piecewise C^1 representation of Γ^*.

It is clear that $\Gamma^* = \cup\gamma_{n,k}^*$ satisfies properties 1, 3 of Proposition 4.6. Because each $\gamma_{n,k}^*$ is simple, property 2 follows immediately from Lemma 4.5. We now prove property 4. Since $\lim_{i\to\infty} \Gamma_i = \Gamma^*$, we have

$$\liminf_{i\to\infty} \mathbf{E_S}[\Gamma_i] \geq \mathbf{E_S}[\Gamma^*].$$

The proof of property 1 also shows that

$$\lim_{i\to\infty} \mathbf{E_L}[\Gamma_i] \geq \mathbf{E_L}[\Gamma^*].$$

Therefore,

$$\inf_{\Gamma\in\mathbf{S}^1(D)} \mathbf{E_0}[\Gamma] \leq \mathbf{E_0}[\Gamma^*] \leq \liminf_{i\to\infty} \mathbf{E_0}[\Gamma_i] = \inf_{\Gamma\in\mathbf{S}^1(D)} \mathbf{E_0}[\Gamma]. \qquad \square$$

5. Conclusion.

THEOREM 5.1. *Let $f(x) \in L^\infty(D)$ where $D = [0, L] \times [0, L]$. Then there exists a $\Gamma^* \in \mathbf{S}^1(D)$ such that $\mathbf{E_0}[\Gamma^*] = \inf_{\Gamma\in\mathbf{S}^1(D)} \mathbf{E_0}[\Gamma]$. Moreover, every such Γ^* satisfies the following properties.*

1. *Γ^* is crack-free and there exists a constant $K = K(D, \alpha, \max_D |f|)$ such that*

$$R(\Gamma^*) + E(\Gamma^*) + J(\Gamma^*) + B(\Gamma^*) \leq K.$$

2. *Every edge in Γ^* is C^1.*
3. *Suppose $\gamma = \gamma(s)$ is an edge of Γ^* parametrized by the arc length. Let $T(s) = \gamma'(s)$ be the unit tangent vector of γ. Then*

$$|T(s_1) - T(s_2)| \leq C_0|s_1 - s_2|,$$

where $C_0 = 18 \max_D |f|^2/\alpha$.

4. *Every junction in D° connects to exactly three edges such that the angle between any two edges is $2\pi/3$. Every junction on ∂D connects one edge to ∂D such that the edge meets ∂D perpendicularly.*

Before proving Theorem 5.1, we first examine the effect a small perturbation of Γ will have on $\mathbf{E_0}[\Gamma]$. Let $\Gamma^* \in \mathbf{S}^1(D)$ and $\gamma \subset \Gamma^*$ be a piece of C^1 curve parametrized by its arc length s,

$$\gamma(s) : [0, l] \longrightarrow \mathbf{R}^2.$$

Consider a perturbation of $\gamma(s)$ with a sufficiently small $h > 0$:

$$\gamma_h(s) = \gamma(s) - ha(s)S_0,$$

where $a(s) \in C_0^\infty[0, l]$ and S_0 is a unit vector pointing to a fixed side of γ on the support of $a(s)$. This can be achieved if the support of $a(s)$ is sufficiently small. We have

$$|\gamma_h'(x)|^2 = |T(s) - ha'(s)S_0|^2 = 1 - 2ha'(s)g_1(s) + o(h),$$

where $g_1(s) = \langle S_0, T(s)\rangle$ with $\langle .,. \rangle$ being the inner product in $\mathbf{R}^2$. Hence

$$|\gamma_h'(x)| = 1 - ha'(s)g_1(s) + o(h).$$

Denote the region on the left side of γ (with respect to the orientation of $\gamma(s)$) by Ω_L and the region on the right side of γ by Ω_R. Let Ω_h be the domain sandwiched by γ and γ_h,

$$\Omega_h = \left\{\gamma(s) - ta(s)S_0 \;\middle|\; 0 \leq s \leq l,\ 0 \leq t \leq h\right\}.$$

Then

$$A(\Omega_h) = \int_0^l \int_0^h |J(\gamma_t(s)|dtds = O(h),$$

where $J(\gamma_t(s))$ is the Jacobian of $\gamma_t(s)$:

$$|J(\gamma_t(s))| = \left|\det \begin{pmatrix} \gamma'(s) - ta'(s)S_0 \\ -a(s)S_0 \end{pmatrix}\right| = \left|\det \begin{pmatrix} T(s) \\ -a(s)S_0 \end{pmatrix}\right| = a(s)g_2(s).$$

LEMMA 5.2. *Let* $\Gamma^h = (\Gamma^* \setminus \gamma) \cup \gamma_h$. *Then*

$$\mathbf{E_0}[\Gamma^h] - \mathbf{E_0}[\Gamma^*] = \int_0^l \int_0^h F(\gamma_t(s))g_2(s)a(s)\,dt\,ds - \alpha h \int_0^l g_1(s)a'(s)\,ds + o(h),$$

where $F(x) = (c_{\Omega_R} - c_{\Omega_L})(2f(x) - c_{\Omega_L} - c_{\Omega_R})$.

Proof. Without loss of generality, we assume that $\Omega_h \subset \Omega_R$. Calculations in the proof of Proposition 3.12 have shown that

$$c_{\Omega_L \cup \Omega_h} = c_{\Omega_L} + \varepsilon_L, \qquad c_{\Omega_R \setminus \Omega_h} = c_{\Omega_R} + \varepsilon_R,$$

where $\varepsilon_L,\ \varepsilon_R = O(A(\Omega_h)) = O(h)$.

$$\begin{aligned}
&\int_{\Omega_L \cup \Omega_h} (f - c_{\Omega_L \cup \Omega_h})^2 dx - \int_{\Omega_L} (f - c_{\Omega_L})^2 dx \\
&= \int_{\Omega_L} (f - c_{\Omega_L} - \varepsilon_L)^2 dx + \int_{\Omega_h} (f - c_{\Omega_L} - \varepsilon_L)^2 dx - \int_{\Omega_L} (f - c_{\Omega_L})^2 dx \\
&= \int_{\Omega_L} -2\varepsilon_L (f - c_{\Omega_L}) dx + \int_{\Omega_L} \varepsilon_L^2 dx + \int_{\Omega_h} (f - c_{\Omega_L} - \varepsilon_L)^2 dx \\
&= \int_{\Omega_h} (f - c_{\Omega_L})^2 dx + o(h).
\end{aligned}$$

Similarly,

$$\int_{\Omega_R \setminus \Omega_h} (f - c_{\Omega_R \setminus \Omega_h})^2 dx - \int_{\Omega_R} (f - c_{\Omega_R})^2 dx = -\int_{\Omega_h} (f - c_{\Omega_R})^2 dx + o(h).$$

Therefore

$$\begin{aligned}
&\mathbf{E_S}[\Gamma^h] - \mathbf{E_S}[\Gamma^*] \\
&= \int_{\Omega_h} \left\{ (f - c_{\Omega_L})^2 - (f - c_{\Omega_R})^2 \right\} dx + o(h) \\
&= \int_0^l \int_0^h \left\{ \left(f(\gamma_t(s)) - c_{\Omega_L} \right)^2 - \left(f(\gamma_t(s)) - c_{\Omega_R} \right)^2 \right\} |J(\gamma_t(s))| dtds + o(h) \\
&= \int_0^l \int_0^h \left\{ \left(f(\gamma_t(s)) - c_{\Omega_L} \right)^2 - \left(f(\gamma_t(s)) - c_{\Omega_R} \right)^2 \right\} a(s)g_2(s) dtds + o(h) \\
&= h \int_0^l F(\gamma_t(s))g_2(s)a(s)ds + o(h).
\end{aligned}$$

$$\mathbf{E_L}[\Gamma^h] - \mathbf{E_L}[\Gamma^*] = \alpha \int_0^l (|\gamma_h'(s)| - |\gamma'(s)|)\,ds = -\alpha h \int_0^l g_1(s)a'(s)ds + o(h).$$

This proves the lemma. □

Proof of Theorem 5.1. It is clear from Proposition 4.6 that there exists a $\Gamma^* \in \mathbf{S}^1(D)$ such that $E(\Gamma^*) = \inf_{\Gamma \in \mathbf{S}^1(D)} E(\Gamma)$, and that for each such Γ^* it must satisfy property 1. We show that Γ^* satisfies properties 2–4.

First we prove that the unit tangent vector of every C^1 curve in Γ^* must satisfy Lipschitz condition. Property 3 will follow easily from property 2, which we will prove later. Let $\gamma(s)\colon [0,l] \longrightarrow \mathbf{R}^2$ be an edge of Γ^* parametrized by its arc length. Using the same notations as in Lemma 5.2, we have

$$\mathbf{E_0}[\Gamma^h] - \mathbf{E_0}[\Gamma^*] = \int_0^l \int_0^h F(\gamma_t(s)) g_2(s) a(s)\, dt\, ds - \alpha h \int_0^l g_1(s) a'(s)\, ds + o(h).$$

Let $N(s)$ be the unit normal vector of $\gamma(s)$ pointing to the region Ω_L. Suppose $1 > |T(s_1) - T(s_0)| > C|s_1 - s_0|$ where $s_0, s_1 \in [0,l]$ are sufficiently close. Choose $S_0 = N(s_0)$. Then $g_1(s_0) = \langle S_0, T(s_0) \rangle = 0$ and

$$|g_1(s_1)| = |\langle S_0, T(s_1) \rangle| > \frac{C}{2} |s_1 - s_0|.$$

Let $\operatorname{supp}(a(s)) \subseteq [s_0, s_1]$. So

$$\begin{aligned} \mathbf{E_0}[\Gamma^h] - \mathbf{E_0}[\Gamma^*] &= \int_0^l \int_0^h F(\gamma_t(s)) g_2(s) a(s)\, dt\, ds - \alpha h \int_0^l g_1(s) a'(s)\, ds + o(h) \\ &= h \int_{s_0}^{s_1} \Big(\int_{s_0}^{s} G(t) g_2(t)\, dt - \alpha g_1(s) \Big) a'(s)\, ds + o(h), \end{aligned}$$

where $G(s) = \frac{1}{h} \int_0^l \int_0^h F(\gamma_t(s))\, dt$. Note that

$$\left| \int_{s_0}^{s} G(t) g_2(t)\, dt \right| \leq \int_{s_0}^{s} |G(t) g_2(t)|\, dt < 8 \max_D |f|^2 |s - s_0|.$$

So if $C > 18 \max_D |f|^2 / \alpha$ then

$$\begin{aligned} &\max_{s \in [s_0, s_1]} \Big(\int_{s_0}^{s} G(t) g_2(t)\, dt - \alpha g_1(s) \Big) - \min_{s \in [s_0, s_1]} \Big(\int_{s_0}^{s} G(t) g_2(t)\, dt \Big) \\ &\geq \left| \Big(\int_{s_0}^{s_1} G(t) g_2(t)\, dt - \alpha g_1(s_1) \Big) - \Big(\int_{s_0}^{s_0} G(t) g_2(t)\, dt - \alpha g_1(s_0) \Big) \right| \\ &\geq \frac{18 \max_D |f|^2}{2\alpha} \alpha |s_1 - s_0| - 8 \max_D |f|^2 |s_1 - s_0| \\ &= \max_D |f|^2 |s_1 - s_0| \ > \ 0. \end{aligned}$$

Therefore we can find an $a(s) \in C_0^\infty([0,l])$ such that $\mathbf{E_0}[\Gamma^h] < \mathbf{E_0}[\Gamma^*]$ by choosing a sufficiently small h. This is impossible. Hence $|T(s_1) - T(s_0)| < (18 \max_D |f|^2 / \alpha)\, |s_1 - s_0|$.

Next we prove that property 4 must be satisfied by Γ^*. Let P be any junction in Γ^* such that $P \notin \partial D$; assume that two edges γ_1 and γ_2 meet at P at angle $0 < \theta < \pi$. Consider $P' \in D$ such that $|PP'| = h$ and PP' bisect angle θ, as illustrated in Fig. 3. Let $A \in \gamma_1$ and $B \in \gamma_2$ be sufficiently close to P and $|AP| = |BP| = a$. We first assume that both γ_1 and γ_2 are locally linear around P. Denote the domain enclosed by the polygon $APBP'$ by Ω. Elementary trigonometry shows that

$$|AP'| = |BP'| = a - h \cos \frac{\theta}{2} + o(h),$$

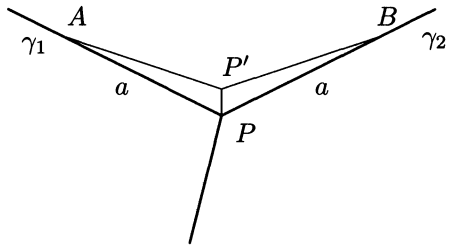


FIG. 3. *Close-up of a junction.*

$$A(\Omega) = ah \sin \frac{\theta}{2}.$$

Let

$$\Gamma^h = \Big(\Gamma^* \setminus \{\text{line segment } AP,\ BP\}\Big) \cup \{\text{line segments } PP',\ AP',\ BP'\}.$$

Calculations in the proof of Proposition 3.12 have shown that

$$\mathbf{E_S}[\Gamma^h] - \mathbf{E_S}[\Gamma^*] \leq C_1 A(\Omega) \ = \ C_1 ah \sin \frac{\theta}{2},$$
$$\mathbf{E_L}[\Gamma^h] - \mathbf{E_L}[\Gamma^*] = \alpha h \left(1 - 2\cos \frac{\theta}{2}\right) + o(h).$$

Thus,

$$\mathbf{E_0}[\Gamma^h] - \mathbf{E_0}[\Gamma^*] \leq \left(C_1 a \sin \frac{\theta}{2} + \alpha\left(1 - 2\cos \frac{\theta}{2}\right)\right) h + o(h).$$

If $0 < \theta < 2\pi/3$ then we can choose sufficiently small a and h so that $\mathbf{E_0}[\Gamma^h] - \mathbf{E_0}[\Gamma^*] < 0$. This contradicts the fact that Γ^* minimizes $\mathbf{E_0}[\Gamma]$. Thus, the angle at which γ_1 and γ_2 meet must be $2\pi/3$ or more.

In general, γ_1 and γ_2 are not locally linear. Let the length of γ_1 between A and P be s_1 and let the length of γ_2 between B and P be s_2. Because both γ_1 and γ_2 are C^1 and the unit tangent vectors parametrized by their respected arc length satisfy the Lipschitz condition, for $i = 1, 2$ we have

$$\begin{aligned} a &= \left| \int_0^{s_i} \gamma_i'(s)\, ds \right| \\ &= \left| \int_0^{s_i} \gamma_i'(0)\, ds + \int_0^{s_i} (\gamma_i'(s) - \gamma_i'(0))\, ds \right| \\ &\geq s - \int_0^{s_i} C_0 s\, ds \\ &= s_i - \frac{C_0 s_i^2}{2}; \end{aligned}$$

hence $s_i - a = O(a^2)$. Similarly, we can show that the area enclosed by γ_1 and AP and that by γ_2 and BP are both $O(a^3)$. Thus all arguments used in the locally linear case will not be affected. So $\theta \geq 2\pi/3$.

Therefore at any junction $P \in D$, the angle at which every two edges meet should be no less than $2\pi/3$. Consequently, the junction must connect exactly three edges and the angle at which every two edges meet must be $2\pi/3$.

Suppose $P \in \partial D$ is a junction. Let $0 < \theta \le \pi/2$ be the angle at which an edge γ meets ∂D. As in the $P \notin \partial D$ case, we may assume that γ is locally linear at P. Consider $A \in \gamma$ and $B \in \partial D$ such that the line segment AB is perpendicular to the boundary. Assume that $|PA| = h$; thus $|AB| = h\sin\theta$. Let

$$\Gamma^h = \Big(\Gamma^* \setminus \{\text{line segment } PA\}\Big) \cup \{\text{line segments } AB\}.$$

Then

$$\begin{aligned}\mathbf{E_0}[\Gamma^h] - \mathbf{E_0}[\Gamma^*] &\le \frac{C_1}{2}h^2 \sin\theta\cos\theta - (1-\sin\theta)h + o(h)\\ &\le -(1-\sin\theta)h + o(h).\end{aligned}$$

Hence $\theta = \pi/2$, i.e., γ must meet ∂D perpendicularly.

Property 2 can be proved by using essentially the same idea. If two C^1 curves meet at a nonjunction point such that they form a corner at that point, then we can decrease $\mathbf{E_0}$ by cutting the corner. We omit the details of the proof here. □

THEOREM 5.3. *Let $f(x)$ be continuous on $D = [0,L]\times[0,L]$. Suppose $\Gamma^* \in \mathbf{S}^1(D)$ and*

$$\mathbf{E_0}[\Gamma^*] = \inf_{\Gamma\in\mathbf{S}^1(D)} \mathbf{E_0}[\Gamma].$$

Then $\Gamma^* \in \mathbf{S}^2(D)$. Moreover, let γ be any edge of Γ^* and $x \in \gamma$ be a nonjuction point. Then

$$\alpha\kappa(x) = (c_{\Omega_R} - c_{\Omega_L})\big(c_{\Omega_R} + c_{\Omega_L} - 2f(x)\big),$$

where γ is oriented with Ω_L and Ω_R being the region on its left and right, respectively, and $\kappa(x)$ is the curvature of γ at x.

Proof. Again, we use the same notations as in Lemma 5.2. Let $\gamma(s)\colon [0,l] \longrightarrow \mathbf{R}^2$ be any edge of Γ^* parametrized by its arc length s. Then according to Lemma 5.2,

$$\mathbf{E_0}[\Gamma^h] - \mathbf{E_0}[\Gamma^*] = \int_0^l \int_0^h F(\gamma_t(s))g_2(s)a(s)\,dt\,ds - \alpha h\int_0^l g_1(s)a'(s)\,ds + o(h).$$

Denote $F_0(s) = (c_{\Omega_R} - c_{\Omega_L})(2f(\gamma(s)) - c_{\Omega_L} - c_{\Omega_R})$. Then because $f(x)$ is continuous, $F(\gamma_t(s)) - F_0(s) \to 0$ as $h \to 0$. Thus

$$\mathbf{E_0}[\Gamma^h] - \mathbf{E_0}[\Gamma^*] = h\int_0^l \Big(F_0(s)g_2(s)a(s) - \alpha g_1(s)a'(s)\Big)\,ds + o(h).$$

Hence

$$\int_0^l \Big(F(s)g_2(s)a(s) - \alpha g_1(s)a'(s)\Big)ds = 0,$$

$$\int_0^l \left\{-\Big(\int_0^s F(t)g_2(t)dt\Big)a'(s) - \alpha g_1(s)a'(s)\right\}ds = 0.$$

Because $a(s)$ can be any function in $C_0^\infty([0,l])$ as long as S_0 points to a fixed side of γ on the support of $a(s)$, it implies

$$\int_0^s F(t)g_2(t)dt + \alpha g_1(s) = \text{constant}$$

on any interval $[b,c] \subseteq (0,l)$ in which S_0 points to a fixed side of γ. Thus on $[b,c]$, $g_1(s) = \langle S_0, T(s)\rangle$ is C^1. Since S_0 is arbitrary, $T(s)$ is C^1 and hence $\gamma(s)$ is C^2. Let $x = \gamma(s_0)$ where $s_0 \in [0,l]$ and choose $S_0 = N(s_0)$ where $N(s)$ is the unit normal vector of $\gamma(s)$ pointing to the region Ω_L. Then

$$\alpha\langle S_0, -\kappa(s)N(s)\rangle = \alpha\langle S_0, T'(s)\rangle = \alpha g_1'(s) = -F(s)g_2(s).$$

It is clear that $S_0 = N(s_0)$ implies $g_2(s_0) = 1$; hence

$$\begin{aligned}\alpha\kappa(x) &= -\Big(f\big(\gamma(s)\big) - c_{\Omega_L}\Big)^2 + \Big(f\big(\gamma(s)\big) - c_{\Omega_R}\Big)^2 \\ &= (c_{\Omega_R} - c_{\Omega_L})\Big(c_{\Omega_R} + c_{\Omega_L} - 2f(x)\Big). \qquad \square\end{aligned}$$

Acknowledgments. This paper is part of the author's dissertation under Professor David Mumford at Harvard University. The author is extremely indebted to Professor Mumford, without whose supervision as well as his kindness and inspiration this work would not have been accomplished. The author is also greatly indebted to Taka Shiota for his immense help. Finally, the author would like to thank the anonymous referees for their very constructive comments.

REFERENCES

[1] L. Ambrosio, *Variational problems in* SBV, Acta Appl. Math., 17 (1989), pp. 1–40.
[2] A. Blake and A. Zisserman, *Using Weak Continuity Constraints*, Report CSR-186-85, Dept. of Computer Science, Edinburgh University, 1985.
[3] ———, *Visual Reconstruction*, MIT Press, Cambridge, 1987.
[4] G. Congredo and I. Tamiani, *On the existence of a problem in image segmentation*, preprint, 1989.
[5] E. Di Giorgi, M. Carriero, and A. Leaci, *Existence theorem for a minimum problem with free discontinuity set*, Arch. Rational Math. Anal., 108 (1989), pp. 195–218.
[6] K. Falconer, *The geometry of fractal sets,* Cambridge University Press, Cambridge, 1985.
[7] S. Kulkarni, S. Mitter, and T. Richardson, *An existence result and lattice approximations for a variational problem arising in computer vision*, Proc. Signal Processing Workshop, Instit. Math and Appl., Minneapolis, MN, 1989.
[8] G. Maso, J. Morel, and S. Solimini, *A variational method in image segmentation: existence and approximation results*, Acta Math., 168 (1992), pp. 89–151.
[9] D. Mumford and J. Shah, *Boundary detection by minimizing functionals*, I, Proc. IEEE CVPR, 1985.
[10] ———, *Optimal approximations by piecewise smooth functions and associated variational problems*, Comm. Pure Appl. Math., 42 (1989), pp. 577–685.
[11] T. Richardson, *Limit theorems for a variational problem arising in computer vision,* Ann. Sch. Normale Sup. Pisa (4), 19 (1992), pp. 1–49.
[12] J. Shah, *Properties of energy-minimizing segmentations*, Siam J. Control Optim., 30 (1992), pp. 99–111.