SIAM J. OPTIM.
Vol. 12, No. 1, pp. 1–17

ERROR BOUNDS FOR LOWER SEMICONTINUOUS FUNCTIONS IN NORMED SPACES*

KUNG FU NG† AND XI YIN ZHENG‡

Abstract. Without the convexity or analyticity assumption, we study error bounds for an inequality system defined by a general lower semicontinuous function and establish sufficient/necessary conditions on the existence of error bounds in infinite dimensional normed spaces. Some characterizations for a convex inequality system to possess an error bound in a reflexive Banach space are also given. As applications, in dealing with the Hoffman error bound result in normed spaces, we give a computable Lipschitz bound constant, which is better than previous Lipschitz bound constants in some examples; we also consider error bounds for quadratic functions on R^n.

Key words. error bound, lower semicontinuous function, convex function, normed space, Lipschitz bound constant

AMS subject classifications. 90C31, 90C25, 49J52

PII. S1052623499358884

1. Introduction. Let X be a normed space, $f : X \to R\bigcup\{+\infty\}$ a proper lower semicontinuous function, and let $S = \{x \in X | f(x) \leqslant 0\}$. We always assume $S \neq \emptyset$. We say that an error bound for f holds if there is a positive constant τ such that for each $x \in X$

$$dist(x, S) \leqslant \tau[f(x)]_+,$$

where $dist(x, S) = \inf\{\|x - y\| | y \in S\}$ and $[f(x)]_+ = \max\{f(x), 0\}$. Error bounds have important applications in sensitivity analysis of mathematical programming and in convergence analysis of some algorithms. In recent years, the study of error bounds has received increasing attention in the mathematical programming literature. See [1, 5, 6, 8, 9, 10, 11, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27] and especially the excellent survey papers [16, 26] for details. However, most of the previous error bound results are concerned with either convex or analytic [20] inequality systems, and to the best of our knowledge an error bound for a general lower semicontinuous function has not been studied. One of our aims is to study an error bound for a lower semicontinuous function on X. Using the Dini-directional derivatives, a main result (Theorem 2.5) in section 2 implies that if X is a Banach space and f is lower semicontinuous, then an error bound for f holds provided that

$$\sup_{x \in X \setminus S} \inf_{\|h\|=1} \underline{d}^+ f(x)(h) < 0 \tag{1.1}$$

(a partial converse is given in Theorem 2.7). If f is convex and X is a reflexive Banach space, then one has that an error bound for f holds if and only if (1.1) holds, and

*Received by the editors July 7, 1999; accepted for publication (in revised form) November 8, 2000; published electronically May 22, 2001. This research was supported by an earmarked grant from the Research Grant Council of Hong Kong.

http://www.siam.org/journals/siopt/12-1/35888.html

†Department of Mathematics, The Chinese University of Hong Kong, Shatin, New Territory, Hong Kong (kfng@math.cuhk.edu.hk).

‡Department of Mathematics, Yunnan University, Kunming 650091, People's Republic of China (xyzheng@ynu.edu.cn). This author's work was supported by the National Natural Science Foundation of the People's Republic of China and by ABSF of Yunnan Province, China.

this is the case if and only if there exists a positive constant δ such that

$$\inf\{d^+f(x)(h)|h \in N_S^1(x)\} \geqslant \delta \tag{1.2}$$

whenever x is a boundary point of S with

$$\emptyset \neq N_S^1(x) := \{h \in X | \|h\| = 1 \text{ and } dist(x + th, S) = t \text{ for some } t > 0\}.$$

(The above characterization in terms of $N_S^1(x)$ is due to Lewis and Pang [16] for the case when $X = R^n$.) As applications, we deal with, in Theorem 4.4, the Hoffman error bound result in normed spaces (our proof is very different from that of Hoffman [14] and others [10, 3]) and give a computable Lipschitz bound constant; we also give a systematic treatment for the existence (or nonexistence) of error bounds for quadratic functions.

2. An error bound for a proper lower semicontinuous function on a normed space. We will need the following notation. For $x \in X$ and $h \in X$ with $\|h\| = 1$, we denote the upper and lower Dini-directional derivatives, respectively, by

$$\bar{d}^+f(x)(h) := \limsup_{t\to 0^+} \frac{f(x + th) - f(x)}{t}$$

and

$$\underline{d}^+f(x)(h) := \liminf_{t\to 0^+} \frac{f(x + th) - f(x)}{t}.$$

(In the case when $f(x) = +\infty$, we define $\underline{d}^+f(x)(h) = -\infty$.) Clearly, $\underline{d}^+f(x)(h) \leqslant \bar{d}^+f(x)(h)$. If $\underline{d}^+f(x)(h) = \bar{d}^+f(x)(h)$, f is said to be right differentiable at x in the direction h. For $x \in \partial S$ (the boundary of S), define

$$N_S^1(x) = \{h \in X | \|h\| = 1 \text{ and } dist(x + th, S) = t \text{ for some } t > 0\}$$

and

$$\partial_N S = \{x \in \partial S | N_S^1(x) \neq \emptyset\}.$$

If X is a Hilbert space and S is convex, then it is easy to verify that

$$N_S^1(x) = \{h \in N_S(x) | \|h\| = 1\}, \text{ where } N_S(x) = \{v \in X | \langle v, y - x\rangle \leqslant 0, y \in S\}$$

is the normal cone of S at x.

The following result of the mean-valued theorem type will help us establish error bound results in terms of the Dini-directional derivatives. It is known (cf. [4] and [6]) when f is assumed continuous.

LEMMA 2.1. *Let X be a normed space and f a proper lower semicontinuous function on X; let $x \in \text{dom}(f) := \{x \in X | f(x) < +\infty\}$, and let $h \in X$ with $\|h\| = 1$ and $t > 0$. Assume that there exists $\delta \in R$ such that for each $\alpha \in [0,\ t)$, $\underline{d}^+f(x + \alpha h)(h) \leqslant \delta$. Then*

$$f(x + th) - f(x) \leqslant t\delta.$$

Proof. For any $\varepsilon > 0$, let $t_\varepsilon = \sup\{0 \leqslant s \leqslant t | f(x + sh) - f(x) \leqslant s(\delta + \varepsilon)\}$. By the lower semicontinuity of f, one has

$$f(x + t_\varepsilon h) - f(x) \leqslant t_\varepsilon(\delta + \varepsilon). \tag{2.1}$$

We claim that $t_\varepsilon = t$. Indeed, if $t_\varepsilon < t$, then $x+t_\varepsilon h \in dom(f)$ and so $\underline{d}^+ f(x+t_\varepsilon h)(h) \leqslant \delta$ by assumption. Consequently there exists $t' \in (t_\varepsilon,\ t)$ such that

$$f(x + t'h) - f(x + t_\varepsilon h) \leqslant (t' - t_\varepsilon)(\delta + \varepsilon).$$

This and (2.1) imply that $f(x + t'h) - f(x) \leqslant t'(\delta + \varepsilon)$. It follows from the definition of t_ε that $t' \leqslant t_\varepsilon$, a contradiction. This shows that $f(x+th)-f(x) \leqslant t(\delta+\varepsilon)$. Letting $\varepsilon \to 0$, one has that $f(x + th)(h) - f(x) \leqslant t\delta$.

The following lemma itself is an interesting result on an error bound for a general function (without the continuity assumption).

LEMMA 2.2. *Let (X, d) be a metric space and $f : X \to R\bigcup\{+\infty\}$ a proper function; let $\tau > 0$ and $0 \leqslant \rho < 1$ be constants. Suppose that for each $x \in f^{-1}(0,\ +\infty) = \{y \in X | 0 < f(y) < +\infty\}$ there is $x' \in f^{-1}[0,\ +\infty)$ such that*

$$dist(x', S) \leqslant \rho dist(x, S) \tag{2.2}$$

and

$$d(x, x') \leqslant \tau[f(x) - f(x')].$$

Then, for each $x \in X$, $dist(x, S) \leqslant \tau[f(x)]_+$.

Proof. It suffices to show that for each $x \in f^{-1}(0,\ +\infty)$, $dist(x, S) \leqslant \tau f(x)$. For each $x \in f^{-1}(0,\ +\infty)$, let $x_0 = x$ and suppose that $x_0, \ldots, x_k$ have been chosen in $f^{-1}(0,\ +\infty)$ such that for each $1 \leqslant i \leqslant k$

$$dist(x_i, S) \leqslant \rho^i dist(x, S) \quad \text{and} \quad d(x_{i-1}, x_i) \leqslant \tau[f(x_{i-1}) - f(x_i)].$$

By the given condition, there is $x_{k+1} \in f^{-1}[0,\ +\infty)$ such that

$$dist(x_{k+1}, S) \leqslant \rho dist(x_k, S) \leqslant \rho^{k+1} dist(x, S) \text{ and } d(x_k, x_{k+1}) \leqslant \tau[f(x_k) - f(x_{k+1})].$$

If $f(x_{k+1}) = 0$, then

$$dist(x, S) \leqslant d(x_0, x_{k+1}) \leqslant \sum_{i=1}^{k+1} d(x_{i-1}, x_i) \leqslant \tau \sum_{i=1}^{k+1} [f(x_{i-1}) - f(x_i)] = \tau f(x).$$

Thus we may assume that $f(x_{k+1}) \neq 0$, and hence inductively we obtain a sequence $\{x_n\}$ in $f^{-1}(0,\ +\infty)$ such that $x_0 = x$,

$$dist(x_n, S) \leqslant \rho^n dist(x, S) \text{ and } d(x_{n-1}, x_n) \leqslant \tau[f(x_{n-1}) - f(x_n)]$$

for all $n \in N$, where N is the natural number set. Hence for $n \in N$,

$$\begin{aligned} dist(x, S) &\leqslant d(x, x_n) + dist(x_n, S) \\ &\leqslant \sum_{i=1}^{n} d(x_{i-1}, x_i) + \rho^n dist(x, S) \\ &= \tau \sum_{i=1}^{n} [f(x_{i-1}) - f(x_i)] + \rho^n dist(x, S) \\ &\leqslant \tau f(x) + \rho^n dist(x, S) \end{aligned}$$

as $f(x_n) \geqslant 0$. It follows that

$$dist(x, S) \leqslant \frac{\tau}{1-\rho^n} f(x) \text{ for each } n \in N.$$

This implies that

$$dist(x, S) \leqslant \tau f(x).$$

The following lemma is a straightforward consequence of Theorem 2(ii) in Hamel [11]; it shows that in Lemma 2.2, if we add the conditions that X is complete and f is lower semicontinuous, then the conclusion is still true when (2.2) is removed.

LEMMA 2.3. *Let (X, d) be a complete metric space and $f : X \to R\bigcup\{+\infty\}$ a proper lower semicontinuous function; let $\tau > 0$ be a constant. Suppose that for each $x \in f^{-1}(0, +\infty)$ there is $x' \in f^{-1}[0, +\infty)$ such that*

$$0 \neq d(x, x') \leqslant \tau[f(x) - f(x')].$$

Then, for each $x \in X$, $dist(x, S) \leqslant \tau[f(x)]_+$.

Proof. We may assume that $X \neq S$. Let $g(x) = \max\{f(x), 0\}$ for each $x \in X$. Then $g_{\min} := \inf\{g(x) | x \in X\} = 0$ and $S = \{x \in X | g(x) = g_{\min}\}$. For each $x \in dom(g)$ with $g(x) > g_{\min}$ (i.e., $x \in f^{-1}(0, +\infty)$), by the given condition there exists $x' \in f^{-1}[0, +\infty)$ such that $0 \neq d(x, x') \leqslant \tau[f(x) - f(x')]$, that is, $g(x') + \frac{1}{\tau} d(x, x') \leqslant g(x)$; this last inequality is trivially true if $x \notin dom(g)$. By Theorem 2(ii) in Hamel [11], for each $x \in X$, one has

$$dist(x, S) \leqslant \tau(g(x) - g_{\min}),$$

that is,

$$dist(x, S) \leqslant \tau[f(x)]_+.$$

THEOREM 2.4. *Let X be a normed space and f a proper lower semicontinuous function on X; let $0 < \delta < +\infty$ and $0 \leqslant \rho < 1$. Suppose that for each $x \in f^{-1}(0, +\infty)$ there exist $h_x \in X$ with $\|h_x\| = 1$ and $t_x > 0$ such that for $t \in [0, t_x)$,*

$$dist(x + t_x h_x, S) \leqslant \rho dist(x, S) \quad \text{and} \quad \underline{d}^+ f(x + t h_x)(h_x) \leqslant -\delta. \tag{2.3}$$

Then for each $x \in X$, $dist(x, S) \leqslant \frac{1}{\delta}[f(x)]_+$.

Proof. Given $x \in f^{-1}(0, +\infty)$, we will check that there exists $x' \in f^{-1}([0, +\infty))$ satisfying the properties stated in Lemma 2.2 with $\tau = \frac{1}{\delta}$. If $f(x + t_x h_x) \geqslant 0$, then we can take $x' := x + t_x h_x$ because, by Lemma 2.1 (applied to $-\delta$ instead of δ) and assumption (2.3), one has $\|x - x'\| = t_x \leqslant \frac{1}{\delta}[f(x) - f(x + t_x h_x)]$. If $f(x + t_x h_x) < 0$, let $s_x = \sup\{0 \leqslant t \leqslant t_x | f(x + s h_x) > 0 \text{ for each } s \in [0, t]\}$. Then $f(x + s_x h_x) \leqslant 0$ and $f(x + t h_x) > 0$ for all $t \in [0, s_x)$. By the lower semicontinuity of f, S is a closed set. It follows from $x \notin S$ that $dist(x, S) > 0$. Thus, there exists $s'_x \in (0, s_x)$ such that

$$dist(x + s'_x h_x, S) \leqslant \|x + s'_x h_x - (x + s_x h_x)\| = s_x - s'_x \leqslant \rho dist(x, S)$$

and $x + s'_x h_x \in f^{-1}([0, +\infty))$. Note that $\underline{d}^+ f(x + t h_x)(h_x) \leqslant -\delta$ for each $t \in [0, s'_x]$. As in the first part of our proof, it follows that $x + s'_x h_x$ has the desired properties for x'.

If X is assumed to be a Banach space, conditions in Theorem 2.4 can be simplified (by considering $\underline{d}^+ f(x)$ in place of $\underline{d}^+ f(x + th_x)$).

THEOREM 2.5. *Let X be a Banach space, $f : X \to R\bigcup\{+\infty\}$ a proper lower semicontinuous function, and let $0 < \delta < +\infty$. Suppose that for each $x \in f^{-1}(0, +\infty)$ there exists $h_x \in X$ with $\|h_x\| = 1$ such that $\underline{d}^+ f(x)(h_x) \leqslant -\delta$. Then for each $x \in X$, $dist(x, S) \leqslant \frac{1}{\delta}[f(x)]_+$.*

Proof. Let $x \in X$ with $0 < f(x) < +\infty$. Then $\underline{d}^+ f(x)(h_x) \leqslant -\delta < -(\delta - \varepsilon)$ for any $\varepsilon \in (0, \delta)$. It follows from the lower semicontinuity of f and $f(x) > 0$ that there exists $t > 0$ small enough such that $x + th_x \in f^{-1}[0, +\infty)$ and $\frac{1}{t}[f(x + th_x) - f(x)] < -(\delta - \varepsilon)$. Thus

$$\|x - (x + th_x)\| = t \leqslant \frac{1}{\delta - \varepsilon}[f(x) - f(x + th_x)].$$

By Lemma 2.3, one has that for each $x \in X$, $dist(x, S) \leqslant \frac{1}{\delta - \varepsilon}[f(x)]_+$. Letting $\varepsilon \to 0$, the proof is completed.

Recall that the cone of feasible directions of a convex set $C \subset X$ at a point $x \in C$ is, by definition, the set

$$\mathcal{F}_C(x) = \{v \in X | x + tv \in C; \text{ for some } t > 0\}.$$

The following is a general constraint error bound result.

COROLLARY 2.6. *Let X be a Banach space, $f : X \to R\bigcup\{+\infty\}$ a proper lower semicontinuous function; let $0 < \delta < +\infty$ and C be a closed convex subset of X such that $S_C = S\bigcap C \neq \emptyset$. Suppose that for each $x \in f^{-1}(0, +\infty)\bigcap C$ there exists $h_x \in \mathcal{F}_C(x)$ with $\|h_x\| = 1$ such that $\underline{d}^+ f(x)(h_x) \leqslant -\delta$. Then for each $x \in C$, $dist(x, S_C) \leqslant \frac{1}{\delta}[f(x)]_+$.*

Proof. Define $g(x) := f(x) + \delta_C(x)$ for each $x \in X$, where δ_C is the indicator function of C. Then g is a proper lower semicontinuous function, $g^{-1}(0, +\infty) = f^{-1}(0, +\infty)\bigcap C$, $S_C = \{x \in X | g(x) \leqslant 0\}$, and for each $x \in g^{-1}(0, +\infty)$ and $h \in \mathcal{F}_C(x)$, $\underline{d}^+ g(x)(h) = \underline{d}^+ f(x)(h)$. It follows from the given conditions and Theorem 2.5 that for each $x \in X$, $dist(x, S_C) \leqslant \frac{1}{\delta}[g(x)]_+$. This implies that $dist(x, S_C) \leqslant \frac{1}{\delta}[f(x)]_+$ for all $x \in C$.

The following result gives a necessary condition for f to have a local error bound.

THEOREM 2.7. *Let X be a normed space, f a proper lower semicontinuous function on X, and $\tau > 0$. Suppose that for each $x \in \partial S$, there is $\delta(x) > 0$ such that whenever $y \in X$ with $\|y - x\| < \delta(x)$,*

$$dist(y, S) \leqslant \tau[f(y)]_+. \tag{2.4}$$

Then for each $x \in \partial_N S$,

$$\inf\{\underline{d}^+ f(x)(h) | h \in N_S^1(x)\} \geqslant \frac{1}{\tau}.$$

Proof. For each $x \in \partial_N S$ and each $h \in N_S^1(x)$, one has, by definition, that there exists $t > 0$ such that $dist(x + th, S) = t$. It is easy to verify that for each $s \in (0, t)$, $dist(x + sh, S) = s > 0$, and so $x + sh \notin S$; that is, $f(x + sh) > 0$. Hence, for $0 < s < \min\{\delta(x), t\}$, (2.4) gives

$$s \leqslant \tau f(x + sh) = \tau[f(x + sh) - f(x)].$$

This implies that $\underline{d}^+ f(x)(h) \geqslant \frac{1}{\tau}$, and hence that

$$\inf\{\underline{d}^+ f(x)(h) | h \in N_S^1(x)\} \geqslant \frac{1}{\tau}.$$

COROLLARY 2.8. *Let X be a normed space and f a proper lower semicontinuous function on X. Suppose $\tau > 0$ such that for each $x \in X$, $dist(x, S) \leqslant \tau[f(x)]_+$. Then, for each $x \in \partial_N S$,*

$$\inf\{\underline{d}^+ f(x)(h) | h \in N_S^1(x)\} \geqslant \frac{1}{\tau}. \tag{2.5}$$

In section 3, we shall show that (2.5) is also sufficient for f to have an error bound when f is assumed convex.

COROLLARY 2.9. *Let X be a finite dimensional normed space and f a differentiable function on X. Suppose that an error bound for f holds. Then f satisfies the Slater condition (i.e., there exists $x_0 \in X$ such that $f(x_0) < 0$).*

Proof. Suppose to the contrary that f does not satisfy the Slater condition. Then for each $x \in S$, one has $f(x) = \inf\{f(y) | y \in X\}$, and so $\bigtriangledown f(x) = 0$. It follows from Corollary 2.8 that $N_S^1(x)$ must be empty for each $x \in \partial S$. Pick a point z in $X \backslash S$. Since X is a finite dimensional space, there exists $x_0 \in \partial S$ such that $dist(z, S) = \|z - x_0\|$. This implies that

$$\frac{z - x_0}{\|z - x_0\|} \in N_S^1(x_0),$$

contradicting an earlier remark.

For convex f, the finite dimension assumption of X can be relaxed.

COROLLARY 2.10. *Let X be a reflexive Banach space and f a differentiable convex function on X. Suppose that an error bound for f holds. Then f satisfies the Slater condition.*

Proof. Since f is a differentiable convex function, S is closed convex in X. Pick a point $z \in X \setminus S$; by the reflexivity of X, there exists $x_0 \in \partial S$ such that $dist(z, S) = \|z - x_0\|$. The proof follows as in the proof of Corollary 2.9.

3. Error bounds for lower semicontinuous convex functions on reflexive spaces. The aim of this section is to show that, when f is assumed to be convex, the results presented in the preceding section lead to conditions which are both necessary and sufficient for f to have an error bound. Throughout this section we let f be a proper lower semicontinuous convex function on X. Recall that the right directional derivative and the left directional derivative of f are defined by

$$d^+ f(x)(h) = \lim_{t \to 0^+} \frac{f(x + th) - f(x)}{t}$$

and

$$d^- f(x)(h) = \lim_{t \to 0^-} \frac{f(x + th) - f(x)}{t}.$$

They always exist and $d^- f(x)(h) = -d^+ f(x)(-h)$ for each $x \in dom(f)$ and $h \in X \setminus \{0\}$. When $f(x) = +\infty$, we define $d^+ f(x)(h) = -\infty$ for all $h \in X$. It is known [7] that for $0 \leqslant t_1 < t_2$ with $x + t_1 h$ and $x + t_2 h$ in $dom(f)$,

$$d^+ f(x + t_1 h)(h) \leqslant d^- f(x + t_2 h)(h) \leqslant d^+ f(x + t_2 h)(h). \tag{3.1}$$

THEOREM 3.1. *Let X be a reflexive Banach space and f a proper lower semicontinuous convex function on X. Let $\tau > 0$. Then the following statements are equivalent.*

(i) *τ is an error bound: $dist(x, S) \leqslant \tau[f(x)]_+$ for each $x \in X$.*

(ii) *τ is a local error bound for f at each boundary point of S: for each $x \in \partial S$, there exists $\delta(x) > 0$ such that whenever $y \in X$ with $\|y - x\| < \delta(x)$, one has $dist(y, S) \leqslant \tau[f(y)]_+$.*

(iii) *For each $x \in \partial_N S$,*

$$\inf\{d^+f(x)(h) | h \in N_S^1(x)\} \geqslant \frac{1}{\tau}. \tag{3.2}$$

(iv) *For each $x \in X \setminus S$ there exist $t_x > 0$ and $h_x \in X$ with $\|h_x\| = 1$ satisfying*

$$x + t_x h_x \in S \text{ and } d^+f(x + th_x)(h_x) \leqslant -\frac{1}{\tau} \text{ for all } t \in [0,\ t_x).$$

(v) *For each $x \in X \setminus S$, there exists $h_x \in X$ with $\|h_x\| = 1$ satisfying*

$$d^+f(x)(h_x) \leqslant -\frac{1}{\tau}.$$

Proof. (i)⇒(ii) and (iv)⇒(v) are trivial; (ii)⇒(iii) and (v)⇒(i) follow directly from Theorems 2.7 and 2.5.

(iii)⇒(iv): Note that for each $x \in X$ with $f(x) = +\infty$, $d^+f(x)(h) = -\infty$ for all $h \in X$ with $\|h\| = 1$. Given an $x \in f^{-1}(0,\ +\infty)$, by the reflexivity of X and the convexity of S, there exists $x_0 \in \partial S$ such that $\|x - x_0\| = dist(x, S)$. Let

$$t_x = \|x - x_0\| \text{ and } h_x = \frac{1}{t_x}(x_0 - x).$$

Then $x_0 = x + t_x h_x \in S$ and $-h_x \in N_S^1(x_0)$. By the convexity of f and $x, x + t_x h_x \in dom(f)$, it is clear that $x + th_x \in dom(f)$ for each $t \in [0,\ t_x]$. By (iii), one has that $d^+f(x_0)(-h_x) \geqslant \frac{1}{\tau}$, that is, $d^-f(x_0)(h_x) \leqslant -\frac{1}{\tau}$. Consequently, by (3.1), we have that for each $t \in [0,\ t_x)$,

$$d^+f(x + th_x)(h_x) \leqslant d^-f(x + t_x h_x)(h_x) = d^-f(x_0)(h_x) \leqslant -\frac{1}{\tau}.$$

Let $\beta = \inf\{\tau > 0 | dist(x, S) \leqslant \tau[f(x)]_+ \text{ for each } x \in X\}$. Note that $\beta = 0$ if and only if $X = S$ and that $\beta = +\infty$ if and only if f does not have any error bound. The following result comes directly from Theorem 3.1.

COROLLARY 3.2. *Let X be a reflexive Banach space and f a proper lower semicontinuous convex function. Then*

$$\frac{1}{\beta} = -\sup_{x \in X \setminus S} \inf_{h \in X, \|h\| = 1} d^+f(x)(h) = \inf_{x \in \partial_N S} \inf_{h \in N_S^1(x)} d^+f(x)(h).$$

If the continuity is assumed in place of the lower semicontinuity, we have the following.

THEOREM 3.3. *Let X be a reflexive Banach space and f a continuous convex function. Then f has an error bound if and only if*

$$\delta := \inf_{x \in X \setminus S} \inf\{\|x^*\| | x^* \in \partial f(x)\} > 0. \tag{3.3}$$

In this case, $dist(x,S) \leqslant \frac{1}{\delta}[f(x)]_+$ for each $x \in X$.

Proof. Suppose that (3.3) holds. Then, for each $x \in X\backslash S$, $\mathrm{int}(\delta B(X^*)) \bigcap \partial f(x) = \emptyset$, and so by the separation theorem, there exists $h_x \in X$ with $\|h_x\| = 1$ such that

$$-\delta = \inf\{\langle x^*, h_x\rangle | x^* \in \delta B(X^*)\} \geqslant \sup\{\langle x^*, h_x\rangle | x^* \in \partial f(x)\}, \tag{3.4}$$

where $B(X^*)$ denotes the unit ball of the dual space X^* of X. Since f is continuous at x, $d^+f(x)(h_x) = \sup\{\langle x^*, h_x\rangle | x^* \in \partial f(x)\}$, and so, $d^+f(x)(h_x) \leqslant -\delta$. By Theorem 3.1, one obtains that $dist(x,S) \leqslant \frac{1}{\delta}[f(x)]_+$ for each $x \in X$. This shows the sufficiency part. It is clear that the necessity part is directly from (v) of Theorem 3.1 and $d^+f(x)(h_x) = \sup\{\langle x^*, h_x\rangle | x^* \in \partial f(x)\}$.

In [12], R. Y. He informed us that he obtained the result of Theorem 3.3 for general Banach spaces by using suitable subdifferentials and Ekeland's variation principles.

COROLLARY 3.4. *Let X be a reflexive Banach space and f a continuous convex function. Suppose that*

$$\sup_{x\in\partial S} \inf_{h\in X, \|h\|=1} d^+f(x)(h) < 0. \tag{3.5}$$

Then an error bound for f holds.

Proof. We need only show the local version: (ii) of Theorem 3.1 holds. Take $\delta_0 > 0$ such that for each $x \in \partial S$ there exists $h_x \in X$ with $\|h_x\| = 1$ satisfying $d^+f(x)(h_x) < -\delta_0$. Since $d^+f(\cdot)(h_x)$ is upper semicontinuous (see [7]), there exists $r > 0$ such that

$$d^+f(\cdot)(h_x) < -\delta_0 \text{ on } B(x,r).$$

By the continuity of f, for $\varepsilon := \frac{r\delta_0}{2}$, there exists $r_1 \in (0,\ \frac{r}{2})$ such that

$$f(\cdot) = f(\cdot) - f(x) < \varepsilon \text{ on } B(x,r_1). \tag{3.6}$$

Let $y \in B(x,r_1) \setminus S$. Then, it follows from Lemma 2.1 that

$$f\left(y + \frac{r}{2}h_x\right) - f(y) \leqslant -\frac{r}{2}\delta_0.$$

It follows from (3.6) that $f(y + \frac{r}{2}h_x) < 0 < f(y)$, and hence $f(y + t_y h_x) = 0$ for some $t_y \in (0,\ \frac{r}{2})$. Again by Lemma 2.1, $f(y + t_y h_x) - f(y) \leqslant -t_y\delta_0$, and so

$$f(y) \geqslant \delta_0 t_y = \delta_0\|y - (y + t_y h_y)\| \geqslant \delta_0 dist(y,S).$$

Therefore, for each $y \in B(x,r_1)$,

$$dist(y,S) \leqslant \frac{1}{\delta_0}[f(y)]_+.$$

Remark. For the case $X = R^n$, the equivalence of (i) and (iii) in Theorem 3.1 is due to Lewis and Pang [16] (also see Klatte and Li [15]), while the other parts of Theorem 3.1 are new even for the finite dimensional case. Since it is not always easy to compute $N_S^1(x)$ and $\inf\{d^+f(x)(h) | h \in N_S^1(x)\}$, it is sometimes more convenient to check an error bound for f by using (v) of Theorem 3.1 and Theorem 3.3. We will take advantage of this in section 4 in the proof of an extension of Hoffman's error bound result to the setting of normed spaces (our proof is quite different from the approach of Hoffman [14] and that of others [3, 10, 15]). It may be of interest to note

that (3.3) in Theorem 3.3 is a weaker condition than (ACQ9) in [15] (the condition first considered in [16]):

$$\inf_{x\in\partial S}\inf_{x^*\in\partial f(x)}\|x^*\|>0.$$

For example, let $X=R$ and $f(x)=|x|$ for each $x\in X$. Then $S=\{0\}$, and

$$0=\inf_{x\in\partial S}\inf_{x^*\in\partial f(x)}\|x^*\|<\inf_{x\in X\setminus S}\inf_{x^*\in\partial f(x)}\|x^*\|=1.$$

However (3.5) in Corollary 3.4 is equivalent to (AQC9) in Klatte and Li [15]. In what follows, we give a proof of this equivalence. Since $d^+f(x)(h)=\sup\{\langle x^*,h\rangle|x^*\in\partial f(x)\}$, it is clear that (3.5) in Corollary 3.4 implies (ACQ9) in [15]. Conversely, suppose that (ACQ9) holds. Then there exists $\delta>0$ such that for each $x\in\partial S$, $\delta B(X^*)\bigcap\partial f(x)=\emptyset$. As in (3.4) it follows that $\inf_{h\in X,\|h\|=1}d^+f(x)(h)\leqslant-\delta$ for each $x\in\partial S$.

Theorem 2.7, Corollary 2.8, and (iii) of Theorem 3.1 all concern the condition (3.2) for points in $\partial_N S$. By the definitions, it is obvious that $\partial_N S\subset\partial S$. The following proposition further explains the relationship between $\partial_N S$ and ∂S.

PROPOSITION 3.5. *Let X be a reflexive Banach space and f a continuous convex function. Then $\overline{\partial_N S}=\partial S$. In addition, if f also satisfies the Slater condition, $\partial_N S=\partial S$.*

Proof. For $x\in\partial S$ and $\varepsilon>0$, pick a point $y\in X\setminus S$ and a point $y_0\in S$ such that

$$0<\|y-x\|<\frac{\varepsilon}{2}\ \text{ and }\ dist(y,S)=\|y-y_0\|.$$

Then it is easy to verify that $y_0\in\partial_N S$; also $\|y-y_0\|=dist(y,S)\leqslant\|y-x\|$ and so

$$\|y_0-x\|\leqslant\|y-y_0\|+\|y-x\|<\varepsilon.$$

This shows that $x\in\overline{\partial_N S}$, and hence that

$$\overline{\partial_N S}=\partial S.$$

If f satisfies the Slater condition, S is a closed convex set with a nonempty interior. By the separation theorem, for each $x\in\partial S$, there is an $x^*\in X^*$ with $x^*\neq0$ such that

$$\langle x^*,x\rangle=\sup\{\langle x^*,y\rangle|y\in S\}. \tag{3.7}$$

By the reflexivity of X, there is an $h\in X$ with $\|h\|=1$ such that $\langle x^*,h\rangle=\|x^*\|$. By (3.7), for any $y\in S$, we have

$$\|x^*\|=\langle x^*,h\rangle\leqslant\langle x^*,x+h-y\rangle\leqslant\|x^*\|\|x+h-y\|,$$

and so

$$\|x+h-x\|=1\leqslant\|x+h-y\|.$$

This shows that $dist(x+h,S)=1$. Hence $h\in N_S^1(x)$ and so $x\in\partial_N S$. Therefore, $\partial_N S=\partial S$.

If the reflexivity of X in Proposition 3.5 is dropped, $\partial_N S$ may be empty even though f is a continuous convex function satisfying the Slater condition.

Example 3.6. Let X be a nonreflexive Banach space. Then, by the well-known James theorem there exists $x^* \in X^*$ with $x^* \neq 0$ such that for each $h \in X$ with $\|h\| = 1$

$$\langle x^*, h\rangle < \|x^*\|. \tag{3.8}$$

It is clear that x^* satisfies the Slater condition. Next we show that $\partial_N S_* = \emptyset$, where $S_* = \{x \in X | \langle x^*, x\rangle \leqslant 0\}$. Indeed, if there exists $x_0 \in \partial_N S_*$, then $\langle x^*, x_0\rangle = 0$, and for some $h_0 \in X$ with $\|h_0\| = 1$ and some $t_0 > 0$, one has

$$dist(x_0 + t_0 h_0, S_*) = t_0.$$

This and (3.8) imply that $0 < \langle x^*, h_0\rangle < \|x^*\|$. By the definition of $\|x^*\|$, there are $h_1 \in X$ with $\|h_1\| = 1$ and $0 < r < 1$ such that $\langle x^*, rh_1\rangle = \langle x^*, h_0\rangle$. It follows from $\langle x^*, x_0\rangle = 0$ that

$$\langle x^*, x_0 + t_0 h_0 - t_0 r h_1\rangle = 0.$$

This implies that

$$dist(x_0 + t_0 h_0, S_*) \leqslant \|x_0 + t_0 h_0 - (x_0 + t_0 h_0 - t_0 r h_1)\| = r t_0,$$

contradicting $dist(x_0 + t_0 h_0, S_*) = t_0$.

4. Application to linear inequality system. Let X be a normed space and X^* denote the dual space of X; let $x_1^*, \ldots, x_n^* \in X^*$, $c_1, \ldots, c_n \in R$, and $S = \{x \in X | \langle x_i^*, x\rangle - c_i \leqslant 0,\ i = 1, \ldots, n\}$. Define $f(x) = \max\{\langle x_i^*, x\rangle - c_i | i = 1, \ldots, n\}$ for each $x \in X$. It is clear that $S = \{x \in X | f(x) \leqslant 0\}$. Assuming that $S \neq \emptyset$, the fundamental result of Hoffman [13] asserts that if $X = R^n$, then there exists a constant $\tau > 0$ such that $dist(x, S) \leqslant \tau[f(x)]_+$ for each $x \in X$. This Lipschitz error bound constant τ is related to the convergence rate of algorithms appearing in some applications. Several authors considered the constants of this type; see Mangsarian and Shiau [24], Bergthaller and Singer [2], Güler, Hoffman, and Rothblum [10], Burke and Tseng [3], and references therein. In this section, we will explicitly give a new Lipschitz error bound constant, which is better than the previous ones in some examples. We first introduce some notation. Let I denote the set $\{1, \ldots, n\}$ and, for each $i \in I$, g_i denotes the affine function defined by $g_i(x) = \langle x_i^*, x\rangle - c_i$. For each $x \in X$, let $I(x)$ denote the set of active indexes at x, that is, $I(x) = \{i \in I | g_i(x) = f(x)\}$. We recall from [7] that

$$\partial f(x) = \left\{ \sum_{i \in I(x)} t_i x_i^* | t_i \geqslant 0 \text{ and } \sum_{i \in I(x)} t_i = 1 \right\}. \tag{4.1}$$

We make the following definitions.

Definition 4.1. *A nonempty subset D of I is said to be*

(i) *full if*

$$\text{span}\{x_i^* | i \in D\} = \text{span}\{x_i^* | i \in I\}, \tag{4.2}$$

(ii) *regular if $0 \notin \text{co}\{x_i^* | i \in D\}$, that is, $\tau_D > 0$ where*

$$\tau_D := \inf \left\{ \| \sum_{i \in D} t_i x_i^* \| | t_i \geqslant 0 \quad \textit{and} \quad \sum_{i \in D} t_i = 1 \right\}. \tag{4.3}$$

DEFINITION 4.2. *A nonempty subset D of I is called*
(i) *a peak-set in I if there exists $x_D \in X$ such that*

$$g_k(x_D) < g_i(x_D) = f(x_D) \tag{4.4}$$

for all $k \in I \setminus D$ and all $i \in D$,
(ii) *a positive peak-set in I if there exists x_D satisfying (4.4) such that $f(x_D) > 0$,*
(iii) *a normal set in I if it is full and a positive peak-set.*

Let $\mathcal{R}(I)$ denote the family of all full and regular subsets of I, and let $\mathcal{N}(I)$ denote the family of all normal subsets of I. Define

$$\mu := \inf\{\tau_D | D \in \mathcal{N}(I)\} \tag{4.5}$$

and

$$\nu := \inf\{\tau_D | D \in \mathcal{R}(I)\}, \tag{4.6}$$

where τ_D is defined by (4.3). $\mathcal{R}(I)$ can be easily identified. It also is not difficult to identify $\mathcal{N}(I)$, and hence μ is a computable constant. It is easy to see that a subset D of I is in $\mathcal{N}(I)$ if and only if (4.2) holds and there exists a solution x_D of the linear equation system

$$\langle x_i^*, x\rangle - c_i = \langle x_j^*, x\rangle - c_j, \quad i, j \in D, \tag{4.7}$$

such that for each $k \in I \setminus D$ and $i \in D$

$$\langle x_k^*, x_D\rangle - c_k < \langle x_i^*, x_D\rangle - c_i \quad \text{and} \quad 0 < \langle x_i^*, x_D\rangle - c_i. \tag{4.8}$$

For a subset D of I, if (4.2) holds then for each $j \in D$, $\dim(\text{span}\{x_i^* | i \in I\})$ differs to $\dim(\text{span}\{x_i^* - x_j^* | i \in D\})$ at most by 1, and so the solution set of (4.7) is either $z + \bigcap_{i\in I} \ker(x_i^*)$ for some $z \in X$ or $z + Re + \bigcap_{i\in I} \ker(x_i^*)$ for some $z \in X$ and $e \in X \setminus \bigcap_{i\in I} \ker(x_i^*)$, where $\ker(x_i^*) = \{x \in X | \langle x_i^*, x\rangle = 0\}$. Thus, it is not difficult to check whether or not a subset of I is normal. Note that $\mathcal{N}(I)$ depends on the system $\{g_i | i \in I\}$ while $\mathcal{R}(I)$ depends only on $\{x_i^* | i \in I\}$ (not depending on the constants c_i). Note that $\nu > 0$ because I is a finite set (so is $\mathcal{R}(I)$). The main result in this section is to show that $\frac{1}{\mu}$ and $\frac{1}{\nu}$ are Lipschitz error bound constants (for the system $S = \{x \in X | f(x) \leqslant 0\}$).

LEMMA 4.3. *$\mathcal{N}(I) \subset \mathcal{R}(I)$ and so $\mu \geqslant \nu > 0$.*

Proof. By Definitions 4.1 and 4.2, it suffices to show that for each $D \in \mathcal{N}(I)$, $0 \notin \text{co}\{x_i^* | i \in D\}$. Take x_D satisfying (4.4). Then $I(x_D) = D$ and x_D is not a minimizer of f as S is assumed nonempty. It follows from [7] and (4.1) that $0 \notin \partial f(x_D) = \text{co}\{x_i^* | i \in D\}$.

THEOREM 4.4. *Let X be a normed space and $x_1^*, \ldots, x_n^* \in X^*$. Let S, f, g_i, and $x_i^*, \ldots, x_n^* \in X^*$ as above in this section. Let δ, μ, and ν be as in (3.3), (4.5), and (4.6), respectively. Then $0 < \nu \leqslant \mu \leqslant \delta$ and, for each $x \in X$,*

$$dist(x, S) \leqslant \frac{1}{\delta}[f(x)]_+ \leqslant \frac{1}{\nu}[f(x)]_+ \leqslant \frac{1}{\mu}[f(x)]_+.$$

The Lipschitz error bound constants $\frac{1}{\mu}$ and $\frac{1}{\nu}$ given in Theorem 4.4 can be practically computed. The following example shows that the constant $\frac{1}{\mu}$ in Theorem 4.4 is better than previous Lipschitz bound constants in some cases.

Pick $X = R$, $x_1^* = 1$, $x_2^* = \frac{1}{2}$, $x_3^* = \frac{1}{3}$, $c_1 = 3, c_2 = 1, c_3 = 2$, and $f(x) = \max\{x-3, \frac{1}{2}x-1, \frac{1}{3}x-2\}$ for each $x \in R$, that is,

$$f(x) = \begin{cases} \frac{1}{3}x - 2, & x \leqslant -6, \\ \frac{1}{2}x - 1, & -6 \leqslant x \leqslant 4, \\ x - 3, & 4 \leqslant x. \end{cases}$$

Hence $S = \{x \in R : \ f(x) \leqslant 0\} = (-\infty, \ 2] \neq \emptyset$. In this case, $I = \{1, 2, 3\}$ and $\mathcal{N}(I)$ consists of three sets: $\{1\}$, $\{2\}$, and $\{1, 2\}$. It is clear that $\mu = \frac{1}{2}$ and $\nu = \frac{1}{3}$. It is easy to verify that, in this case, Lipschitz error bound constants in [3, 10] are all 3. This shows that, in this case, the Lipschitz constant $\frac{1}{\mu} = 2$ in Theorem 4.4 is better. In fact, one can easily see that 2 is the best Lipschitz error bound constant in this case.

The proof of this theorem is based on its finite dimension version.

LEMMA 4.5. *Assume that X is finite dimensional and that* $\dim(X) = \dim(\text{span}\{x_i^* \ | 1 \leqslant i \leqslant n\})$. *Then Theorem* 4.4 *holds.*

Proof. In view of Theorem 3.3 and Lemma 4.3, we need only prove that $\mu \leqslant \delta$. To do this, let $x \in X \setminus S$ and $x^* \in \partial f(x)$. We have to show that $\mu \leqslant \|x^*\|$. By (4.1), we can express x^* as $x^* = \sum_{i \in I(x)} t_i x_i^*$ for some $t_i \geqslant 0$ such that $\sum_{i \in I(x)} t_i = 1$. We will find a normal set $D \subset I$ containing $I(x)$. Granting this, (4.3) implies that $\tau_D \leqslant \| \sum_{i \in I(x)} t_i x_i^* \|$ (by taking $t_i = 0$ for each $i \in D \setminus I(x)$); hence $\mu \leqslant \tau_D \leqslant \|x^*\|$ as required. Therefore it remains to find D with the stated properties. Since x is fixed, we write $\alpha > 0$ for the constant $f(x)$. Define the polyhedral sets A and B by

$$A = \{y \in X | g_i(y) \leqslant \alpha \ \forall i \in I\} \quad \text{and} \quad B = \{y \in A | g_i(y) = \alpha \ \forall i \in I(x)\}.$$

Note that B is an extreme subset of A. Moreover, A contains no lines. Otherwise, there exists $z \neq 0$ such that $\langle x_i^*, z \rangle = 0$ for all $i \in I$, contradicting the assumption that $\dim(X) = \dim(\text{span}\{x_i^* | i \in I\})$. By Corollary 18.5.3 in [28], B must have at least one extreme point, say e. Then e is also an extreme point of A. Since $e \in B \subset A$, it is easy to verify that $I(x) \subset I(e)$ and $f(e) = f(x) = \alpha$. By Definition 4.2 and the definitions of A and B, it is clear that $I(e)$ is a positive peak-set in I. Finally, we show that $I(e)$ is full. Suppose to the contrary that $\dim(\text{span}\{x_i^* | i \in I(e)\}) < \dim(X)$. Then there exists $z \in X \setminus \{0\}$ such that $\langle x_i^*, z \rangle = 0$ for each $i \in I(e)$. Take $\varepsilon > 0$ small enough such that $g_i(e \pm \varepsilon z) < 0$ for each $i \in I \setminus I(e)$. Then $e \pm \varepsilon z \in B$. This contradicts the fact that e is an extreme point of B. Therefore, letting $D = I(e)$, we see that D is a normal subset of I containing $I(x)$, and the proof of Lemma 4.5 is completed.

Proof of Theorem 4.4. Let $X_0 = \bigcap_{i=1}^n \ker(x_i^*)$. Then X/X_0 is finite dimensional. For each $x \in X$, let $[x]$ denote the equivalent class containing x in X/X_0, that is, $[x] = x + X_0$. Define $\hat{x}_i^* \in (X/X_0)^*$ such that $\langle \hat{x}_i^*, [x] \rangle = \langle x_i^*, x \rangle$ for each $x \in X$ and $i \in I$. It is clear that

$$(X/X_0)^* = \text{span}\{\hat{x}_1^*, \ldots, \hat{x}_n^*\}.$$

Let $\hat{g}_i([x]) := \langle \hat{x}_i^*, [x] \rangle - c_i$ and $\hat{f}([x]) = \max\{\langle \hat{x}_i^*, [x] \rangle - c_i | i \in I\}$ for each $[x] \in X/X_0$. Let $\hat{S} = \{[x] \in X/X_0 : \ \hat{f}([x]) \leqslant 0\}$. It is clear that $\hat{S} = \{[x] : \ x \in S\}$. Hence $dist(x, S) = dist([x], \hat{S})$ for each $x \in X$. Note that for each subset D of I, the norm of $\sum_{i \in D} t_i \hat{x_i^*}$ is equal to the norm of $\sum_{i \in D} t_i x_i^*$ for any $t_i \in R$ and $i \in D$. Moreover, D is of the properties defined in Definitions 4.1 and 4.2 with respect to the system $\{g_i(x) \leqslant 0, \ i \in I\}$ if and only if D has the corresponding properties with respect to

$\{\hat{g}_i^*([x]) \leqslant 0,\ i \in I\}$. Therefore the constants μ, ν remain unchanged regardless of whether they hold with respect to $\{g_i\}$ or $\{\hat{g}_i\}$. Therefore Theorem 4.4 follows from Lemma 4.5.

5. Error bound for a quadratic function on R^n. Consider a general quadratic function on R^n

$$f(x) = x^\perp Q x + b^\perp x + c,$$

where Q is an $n \times n$ symmetric matrix, $b \in R^n$, and $c \in R$; $x^\perp$ denotes the transpose of x. Since we do not assume f being of convexity, Q is not necessarily positive semidefinite. Recall that for a symmetric matrix Q, there exists an inversible matrix A and integer numbers k and m with $0 \leqslant k \leqslant m \leqslant n$ such that

$$A^\perp Q A = \begin{pmatrix} I_k & 0 & 0 \\ 0 & -I_{m-k} & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

where I_k and I_{m-k} are the $k \times k$ unit matrix and the $(m-k) \times (m-k)$ unit matrix, respectively. Define $g(x) = f(Ax)$ for $x \in R^n$. Then

$$\begin{aligned} g(x) &= \sum_{i=1}^{k} x_i^2 - \sum_{i=k+1}^{m} x_i^2 + \sum_{i=1}^{n} c_i x_i + c \\ &= \sum_{i=1}^{k} \left(x_i + \frac{c_i}{2}\right)^2 - \sum_{i=k+1}^{m} \left(x_i - \frac{c_i}{2}\right)^2 + \sum_{i=m+1}^{n} c_i x_i + r, \end{aligned}$$

where $r = -\sum_{i=1}^{k} \frac{c_i^2}{4} + \sum_{i=k+1}^{m} \frac{c_i^2}{4} + c$. Define

$$\phi(x) = \sum_{i=1}^{k} x_i^2 - \sum_{i=k+1}^{m} x_i^2 + \sum_{i=m+1}^{n} c_i x_i + r, \quad x = (x_1, \ldots, x_n) \in R^n. \tag{5.1}$$

It is easy to verify that an error bound for g holds if and only if an error bound for ϕ holds. Notice also that $S_f = AS_g$, where $S_f = \{x \in X | f(x) \leqslant 0\}$ and $S_g = \{x \in X | g(x) \leqslant 0\}$. It is clear that

$$\|A^{-1}\|^{-1} dist(A^{-1}x, S_g) \leqslant dist(x, S_f) \leqslant \|A\| dist(A^{-1}x, S_g).$$

This implies that an error bound for f holds if and only if an error bound for g holds, and so if and only if an error bound for ϕ holds. In the remainder of the paper, we only consider an error for a quadratic function ϕ expressed in the "normal" form (5.1). Clearly, for such a quadratic function ϕ, exactly one of the following six cases will occur. ((C1) is the case that ϕ has linear terms and (C2)–(C6) deal with the remaining cases.)

(C1) There exists $i \in N$ such that $m < i \leqslant n$ and $c_i \neq 0$.
(C2) $k < m$, $r \neq 0$, and $c_i = 0$ for all $m < i \leqslant n$.
(C3) $0 = k < m$, $r = 0$, and $c_i = 0$ for all $m < i \leqslant n$.
(C4) $0 \neq k < m$, $r = 0$, and $c_i = 0$ for all $m < i \leqslant n$.
(C5) $k = m$, $r \geqslant 0$, and $c_i = 0$ for all $m < i \leqslant n$.
(C6) $k = m$, $r < 0$, and $c_i = 0$ for all $m < i \leqslant n$.

THEOREM 5.1. *Let ϕ be a quadratic function expressed in the form* (5.1)*. Then any one of* (C1), (C2), (C3), *and* (C6) *implies that an error bound for ϕ holds, and any one of* (C4) *and* (C5) *implies that there are no error bounds for ϕ.*

Proof. Case 1: (C1) is true. Without loss of generality, we can assume that $c_n \neq 0$. Let $h = (0, \dots, 0, -sign(c_n))$. Then $\|h\| = 1$. Notice that for each $x = (x_1, \dots, x_n) \in R^n$,

$$\bigtriangledown\phi(x) = (2x_1, \dots, 2x_k, -2x_{k+1}, \dots, -2x_m, c_{m+1}, \dots, c_n).$$

For each $x \in R^n \setminus S$,

$$\underline{d}^+\phi(x)(h) = \langle \bigtriangledown\phi(x), h\rangle = -|c_n|.$$

Hence an error bound for ϕ holds by Theorem 2.5.

Case 2: (C2) is true. Then

$$\phi(x) = \sum_{i=1}^{k} x_i^2 - \sum_{i=k+1}^{m} x_i^2 + r.$$

This case is subdivided into three subcases.

$(C2)_1$: $k = 0$ and $r < 0$. Then $S = R^n$, and the result is trivial.

$(C2)_2$: $k \neq 0$ and $r < 0$. For each $x \in R^n \setminus S$, one has

$$\sum_{i=1}^{k} x_i^2 > \sum_{i=k+1}^{m} x_i^2 + |r| \geqslant |r|.$$

Let

$$h_x = (-x_1, \dots, -x_k, 0, \dots, 0)/\left(\sum_{i=1}^{k} x_i^2\right)^{\frac{1}{2}}.$$

Then $\|h_x\| = 1$ and

$$\bar{d}^+\phi(x)(h_x) = \langle \bigtriangledown\phi(x), h_x\rangle = -2\left(\sum_{i=1}^{k} x_i^2\right)^{\frac{1}{2}} \leqslant -2|r|^{\frac{1}{2}}.$$

By Theorem 2.5, an error bound for ϕ holds.

$(C2)_3$: $r > 0$. For each $x \in R^n \setminus S$ with $\sum_{i=k+1}^{m} x_i^2 \leqslant \frac{r}{2}$, one has

$$\phi(x) \geqslant \sum_{i=1}^{k} x_i^2 + \frac{r}{2}. \tag{5.2}$$

Pick a point $y = (y_1, \dots, y_n) \in R^n$ such that $y_i = 0$ for $1 \leqslant i \leqslant k$, $\sum_{i=k+1}^{m} y_i^2 = r$ and $y_i = x_i$ for $m < i \leqslant n$. Then $\phi(y) = 0$, and so

$$dist(x, S) \leqslant \|x - y\| \leqslant \left(\sum_{i=1}^{k} x_i^2\right)^{\frac{1}{2}} + 2r^{\frac{1}{2}}. \tag{5.3}$$

Notice that there is a positive constant τ_r such that for $t \geqslant 0$,

$$\frac{t^{\frac{1}{2}} + 2r^{\frac{1}{2}}}{t + \frac{r}{2}} \leqslant \tau_r.$$

It follows from (5.2) and (5.3) that for each $x \in R^n \setminus S$ with $\sum_{i=k+1}^m x_i^2 < \frac{r}{2}$,

$$dist(x, S) \leqslant \tau_r \phi(x).$$

Next consider those x in $R^n \setminus S$ with $\sum_{i=k+1}^m x_i^2 > \frac{r}{2}$; let

$$h_x = (0, \ldots, 0, x_{k+1}, \ldots, x_m, 0, \ldots, 0) / \left(\sum_{i=k+1}^m x_i^2 \right)^{\frac{1}{2}}.$$

Then $\|h_x\| = 1$ and, for $t \geqslant 0$,

$$\underline{d}^+ \phi(x + th_x)(h_x) = \langle \bigtriangledown \phi(x + th_x), h_x \rangle = -2 \left[\left(\sum_{i=k+1}^m x_i^2 \right)^{\frac{1}{2}} + t \right] \leqslant -(2r)^{\frac{1}{2}}.$$

Let

$$t_x = \left(\sum_{i=1}^k x_i^2 + r \right)^{\frac{1}{2}} - \left(\sum_{i=k+1}^m x_i^2 \right)^{\frac{1}{2}}.$$

Then $\phi(x + t_x h_x) = 0$ and $t_x > 0$ because $x \notin S$. By Lemma 2.1, one has

$$-\phi(x) = \phi(x + t_x h_x) - \phi(x) \leqslant -t_x (2r)^{\frac{1}{2}}.$$

It follows that

$$dist(x, S) \leqslant \|x - (x + t_x h_x)\| = t_x \leqslant \frac{\phi(x)}{(2r)^{\frac{1}{2}}}.$$

Therefore, letting $\tau = \max\{\tau_r, \frac{1}{(2r)^{\frac{1}{2}}}\}$, we have shown that if $(\mathrm{C2})_3$ is true, then

$$dist(x, S) \leqslant \tau \phi(x) \text{ for all } x \in R^n \setminus S.$$

Case 3: (C3) is true. Then $S = R^n$, and the result is trivial.

Case 4: (C4) is true. Then

$$\phi(x) = \sum_{i=1}^k x_i^2 - \sum_{i=k+1}^m x_i^2, \qquad x \in R^n.$$

For $t > 0$, let $x_t = (t, 0, \ldots, 0)$. Then $\phi(x_t) = t^2$. It is clear from $dist(x_t, S) = dist(x_t, \partial S)$ that

$$\begin{aligned} dist(x_t, S) &= \inf\{\|x_t - x\| | x \in R^n, \phi(x) = 0\} \\ &= \inf\left\{ \left((x_1 - t)^2 + \sum_{i=2}^{n} x_i^2 \right)^{\frac{1}{2}} | x \in R^n, \sum_{i=1}^{k} x_i^2 = \sum_{i=k+1}^{m} x_i^2 \right\} \\ &= \inf\left\{ \left[(x_1 - t)^2 + x_1^2 + 2\sum_{i=2}^{k} x_i^2 + \sum_{i=m+1}^{n} x_i^2 \right]^{\frac{1}{2}} | x_i \in R \text{ for each } i \right\} \\ &= \inf\{[(x_1 - t)^2 + x_1^2]^{\frac{1}{2}} | x_1 \in R\} \\ &= \frac{t}{\sqrt{2}}. \end{aligned}$$

Therefore,

$$\lim_{t \to 0^+} \frac{dist(x_t, S)}{\phi(x_t)} = \lim_{t \to 0^+} \frac{1}{\sqrt{2}t} = +\infty.$$

This shows that there are no error bounds for ϕ.

Case 5: (C5) is true. If $r > 0$, $S = \emptyset$. We do not consider this trivial case as stated at the beginning. If $r = 0$, then $\phi(x) = \sum_{i=1}^{k} x_i^2$ and $S = \{0\}$. For $t > 0$, let $x_t = (t, 0, \ldots, 0)$. Then

$$\lim_{t \to 0^+} \frac{dist(x_t, S)}{\phi(x_t)} = \lim_{t \to 0^+} \frac{t}{t^2} = +\infty.$$

This implies that there are no error bounds for ϕ.

Case 6: (C6) is true. Then $\phi(x) = \sum_{i=1}^{k} x_i^2 - |r|$. It follows as part $(C2)_2$ of Case 2 that an error bound for ϕ holds.

Notes added in revision. After the submission of this paper, the authors further carried out, in a follow-up paper [25], a detailed analysis on the issue of error bounds with fractional exponents. The Luo–Sturm inequality for quadratic functions (recently established in [21])

$$dist(x, S) \leqslant \tau(|\phi(x)| + |\phi(x)|^{\frac{1}{2}}).$$

is sharpened: we show that either the first term or the second term at the right-hand side of the above Luo–Sturm inequality can be dropped. The approach of [25] is based on analysis of eigenvalues.

Acknowledgment. We are grateful to the referee who drew our attention to reference [11] and pointed out its connection with our work here.

REFERENCES

[1] A. Auslender and J. P. Crouzeix, *Global regularity theorems*, Math. Oper. Res., 13 (1988), pp. 243–253.

[2] C. Bergthaller and I. Singer, *The distance to a polyhedron*, Linear Algebra Appl., 169 (1992), pp. 111–129.

[3] J. V. Burke and P. Tseng, *A unified analysis of Hoffman's bound via Fenchel duality*, SIAM J. Optim., 6 (1996), pp. 265–282.

[4] W. L. Chan, L. R. Huang, and K. F. Ng, *On generalized second-order derivatives and Taylor expansions in nonsmooth optimization*, SIAM J. Control Optim., 32 (1994), pp. 591–611.
[5] C. C. Chou, K. F. Ng, and J. S. Pang, *Minimizing and stationary sequences of constrained optimization problems*, SIAM J. Control Optim., 36 (1998), pp. 1908–1936.
[6] R. Cominetti and R. Correa, *A generalized second-order derivative in nonsmooth optimization*, SIAM J. Control Optim., 28 (1990), pp. 789–809.
[7] F. H. Clarke, *Optimization and Nonsmooth Analysis*, John Wiley, New York, 1983.
[8] S. Deng, *Computable error bounds for convex inequality systems in reflexive Banach spaces*, SIAM J. Optim., 7 (1997), pp. 274–279.
[9] S. Deng, *Global error bounds for convex inequality systems in Banach spaces*, SIAM J. Control Optim., 36 (1998), pp. 1240–1249.
[10] O. Güler, A. J. Hoffman, and U. G. Rothblum, *Approximations to solutions to systems of linear inequalities*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 688–696.
[11] A. Hamel, *Remarks to equivalent formulation of Ekeland's variational principle*, Optimization, 31 (1994), pp. 233–238.
[12] Y. R. He, *private communication,* The Chinese University of Hong Kong, 2000.
[13] L. R. Huang, K. F. Ng and J. P. Penot, *On minimizing and critical sequences in nonsmooth optimization*, SIAM J. Optim., 10 (2000), pp. 999–1019.
[14] A. J. Hoffman, *On approximate solutions of systems of linear inequalities*, J. Res. Nat. Bur. Standards, 49 (1952), pp. 263–265.
[15] D. Klatte and W. Li, *Asymptotic constraint qualifications and error bounds for convex inequalities*, Math. Program., 84 (1999), pp. 137–160.
[16] A. Lewis and J. S. Pang, *Error bounds for convex inequality systems,* in Generalized Convexity, Generalized Monotonicity: Recent Results, Proceedings of the Fifth Symposium on Generalized Convexity, Luminy, June 1996, J.-P. Crouzeix, J.-E. Martinez-Legaz, and M. Volle, eds., Kluwer Academic Publishers, Dordrecht, 1997, pp. 75–100.
[17] W. Li, *Error bounds for piecewise convex quadratic programs and applications*, SIAM J. Control Optim., 33 (1995), pp. 1510–1529.
[18] W. Li, *Abadie's constraint qualification, metric regularity, and error bounds for differentiable convex inequalities*, SIAM J. Optim., 7 (1997), pp. 966–978.
[19] X. D. Luo and Z. Q. Luo, *Extensions of Hoffman's error bound to polynomial systems*, SIAM J. Optim., 4 (1994), pp. 383–392.
[20] Z. Q. Luo and J. S. Pang, *Error bounds for analytic systems and their applications*, Math. Program., 67 (1994), pp. 1–28.
[21] Z. Q. Luo and J. F. Sturm, *Error bounds for quadratic systems*, in High Performance Optimization, H. Frenk et al., eds., Kluwer, Dordrecht, the Netherlands, 2000, pp. 383–404.
[22] O. L. Mangasarian, *A condition number for differentiable convex inequalities*, Math. Oper. Res., 10 (1985), pp. 175–179.
[23] O. L. Mangasarian and T. H. Shiau, *Error bounds for monotone linear complementarity problems*, Math. Program., 36 (1986), pp. 81–89.
[24] O. L. Mangasarian and T. H. Shau, *A variable-complexity norm maximization problem*, SIAM J. Alg. Discrete Methods, 7 (1986), pp. 455–461.
[25] K. F. Ng and X. Y. Zheng, *Global error bounds with fractional exponents*, Math. Program. Ser. B, 88 (2000), pp. 357-370.
[26] J. S. Pang, *Error bounds in mathematical programming*, Math. Program. Ser. B, 79 (1997), pp. 299–332.
[27] S. M. Robinson, *An application of error bounds for convex programming in a linear space*, SIAM J. Control, 13 (1975), pp. 271–273.
[28] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

SIAM J. OPTIM.
Vol. 12, No. 1, pp. 18–35

FASTER ALGORITHMS FOR THE QUICKEST TRANSSHIPMENT PROBLEM*

LISA K. FLEISCHER†

Abstract. A transshipment problem with demands that exceed network capacity can be solved by sending flow in several waves. How can this be done in the minimum number of waves? This is the question tackled in the quickest transshipment problem. Hoppe and Tardos [*Math. Oper. Res.*, 25 (2000), pp. 36–62] describe the only known polynomial time algorithm to solve this problem. They actually solve the significantly harder problem in which it takes a prespecified amount of time for flow to travel from one end of an arc to the other. Their algorithm repeatedly calls an oracle for submodular function minimization. We present an algorithm that finds a quickest transshipment with a polynomial number of maximum flow computations, and a faster algorithm that also uses minimum cost flow computations. When there is only one sink, we show how the algorithm can be sped up to return a solution using $O(k)$ maximum flow computations, where k is the number of sources.

Hajek and Ogier [*Networks*, 14 (1984), pp. 457–487] describe an algorithm that finds a fractional solution to the single sink quickest transshipment problem on a network with n nodes and m arcs using $O(n)$ maximum flow computations. They actually solve the universally quickest transshipment—a flow over time that minimizes the amount of supply left in the network at every moment of time. In this paper, we show how to solve the universally quickest transshipment in $O(mn \log(n^2/m))$ time, the same asymptotic time as a push-relabel maximum flow computation.

Key words. dynamic network flows, transshipment problem, parametric flow, polynomial time algorithms

AMS subject classifications. 68Q25, 90C08, 90C27, 90C35

PII. S1052623497327295

1. Introduction.

The field of network flows blossomed in the 1940s and 1950s with interest in transportation planning and has developed rapidly since then. There is a significant body of literature devoted to this subject. However, it has largely ignored a crucial aspect of transportation: transportation occurs over time. In the 1960s, Ford and Fulkerson introduced flows over time to include time in the network model. Since then, flows over time have been used widely to model network-structured, decision-making problems over time: problems in electronic communication, production and distribution, economic planning, cash flow, job scheduling, and transportation. For examples, see the surveys of Aronson [4] and Powell, Jaillet, and Odoni [24].

A *flow-over-time* network consists of a network $\mathcal{N}$ on vertex set V with a capacity vector u and a transit-time vector, both associated with the edge set E. Flow moves through this network over time. Edge capacities restrict the rate of flow and edge transit times determine how long each unit of flow spends traversing the network.

Flows over time have been previously referred to as dynamic network flows [4, 8, 14, 15, 16, 18, 19, 20, 21, 22, 24, 26, 29]. However, this term causes confusion about

*Received by the editors September 12, 1997; accepted for publication (in revised form) May 24, 2000; published electronically May 22, 2001. A preliminary version of this paper has appeared in *Proceedings of the Ninth Annual ACM–SIAM Symposium on Discrete Algorithms*, SIAM, Philadelphia, 1998, pp. 147–156.

http://www.siam.org/journals/siopt/12-1/32729.html

†Graduate School of Industrial Administration, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213 (lkf@andrew.cmu.edu). This research was supported in part by ONR through an NDSEG fellowship, by AASERT through grant N00014-95-1-0985, by an AAUW Educational Foundation Selected Professions Fellowship, by the NSF PYI award of Éva Tardos, and by the NSF through grant DMS 9505155.

the problem being considered: "dynamic" is more consistently used for a problem with input that changes over time. In this paper and many of the earlier papers, the network is fixed. It is the solution that changes over time. For this reason, we use the more descriptive, albeit slightly awkward, terminology of flows over time.

We discuss a special case of flows over time: we assume all transit times are zero. This special case has been considered in [2, 14, 20, 28, 19], where it is of interest as a tractable problem in data routing and congestion control. When all transit times are zero, the flow on an arc at any moment of time is independent of the flow on this arc in any previous moment. This independence allows for the problem to be easily decomposed into time intervals. One example of this is the greatly simplified time-expanded graph for networks with all zero transit times. One way to picture networks for flows over time is to equate the transit time of an arc with the arc length and the capacity of an arc with the diameter. Thus a zero transit time arc still can handle only a finite amount of flow in a finite time interval, since the capacity of the arc depends on the diameter, but flow passing through a series of zero transit time arcs arrives at the end of the series the instant it enters the beginning arc.

Models of flows over time with zero transit times capture some time-related issues: they can be used to model instances when network capacities restrict the quantity of flow that can be sent at any one time, and thus necessitate sending flow in phases. Solving these problems efficiently may help in finding a more efficient exact or approximate algorithm to solve harder problems with transit times or multicommodity demands.

1.1. The general model. In this paper we consider problems defined on a network $\mathcal{N} = (V, E, u, S)$ with vertex set V of cardinality n, arc set E of cardinality m, arc capacity vector u, and terminal set S. Vertex i has supply γ_i, which is nonzero if and only if $i \in S$. Let S^+ be the set of terminals, also called *sources*, with $\gamma_i > 0$, and S^- be the set of terminals, also called *sinks*, with $\gamma_i < 0$. The sum of all supply in a set of nodes $A \subset V$ is denoted $\gamma(A) := \sum_{i \in A} \gamma_i$. We assume $\gamma(V) = 0$. Define $U = \max_e u_e$, $\Gamma = \max_i |\gamma_i|$, and $k = |S|$.

1.2. The discrete model. A *(static) transshipment problem* is defined on an arbitrary network with edge capacity vector u and with node supply vector γ. The objective is to find a flow f obeying capacities such that the net flow leaving each node equals the supply at the node: $\sum_j [f_{ij} - f_{ji}] = \gamma_i$ for all vertices i, and $f_e \leq u_e$ for all edges e.

A *transshipment over time* with node supply vector γ that completes by time T is a time-dependent flow $f : \{1, 2, \ldots, T\} \to \mathbf{R}$ through a flow-over-time network that obeys edge capacity constraints $f(t) \leq u$ for all $t \in \{1, 2, \ldots, T\}$, flow conservation constraints $\sum_{t=1}^{r} \sum_j [f_{ij}(t) - f_{ji}(t)] \leq \max\{0, \gamma_i\}$ for all $r \in \{1, 2, \ldots, T\}$ and all $i \in V$, and zeroes all supplies and demands by time T: $\sum_{t=1}^{T} \sum_{j \in V} f_{ij}(t) = \gamma_i$ for all $i \in V$. The *quickest transshipment problem* is a transshipment over time that zeroes all supplies and demands in the minimum possible time. Solving a quickest transshipment problem with fixed supplies is useful for clearing a network after a communication breakdown.

Many of the applications of flows over time need integral solutions: when flows are of big objects, like airplanes or train engines, the amounts are often small, so fractional approximations are not very useful. Ford and Fulkerson [9] describe the first polynomial time algorithm to solve the *maximum flow over time problem*: This is a transshipment problem over time with just one source and sink and unspecified

demand. Their algorithm returns an integral solution. In conjunction with binary search, this can be used to solve the *quickest flow problem*: the quickest transshipment problem with one source and sink. Burkard, Dlaska, and Klinz [5] show that using the discrete Newton's method, instead of binary search, leads to improved run times for this problem. Recently, Hoppe and Tardos [15] described the only known polynomial time algorithm to solve the discrete quickest transshipment problem. All of the above papers actually solve the harder problem in which arcs have nonzero transit times. The algorithm in [15], however, repeatedly calls a general algorithm for submodular function minimization.

Traditional approaches to solving the transshipment over time problem consider the discrete time model and make use of a time-expanded version of the original network [4, 24]. A *time-expanded network* is a directed graph that contains a copy of the network for every time step, and *holdover arcs* from a copy of a node at time θ to the copy of the same node at time $\theta + 1$. It is well known and easy to see that the discrete-time transshipment over time problem is equivalent to a traditional static transshipment problem in the time expanded network with the set of sources composed of the copies of sources in the first copy of the network and the set of sinks consisting of the copies of sinks in the final copy of the network. Unfortunately, the size of this graph depends on T, not $\log T$, and thus its size may be exponential in the size of the input. Hoppe and Tardos [15] describe the only polynomial time algorithm to solve the discrete problem with nonnegative, integer transit times.

1.2.1. Results on quickest transshipment. We present a new framework for solving quickest transshipment problems. A fractional solution for a transshipment over time problem given time bound T is easy to find, and we will show this can be done with a single maximum flow. The new framework also allows us to find integer solutions quickly. In the special case when there is only one sink, we solve the integral quickest transshipment problem with $O(k)$ maximum flows. In the general, multisource, multisink case, we show that the integral quickest transshipment problem can be solved with $O(\min\{km, \frac{k \log \Gamma + k^2 \log(mU)}{1+\log\log(m\Gamma U)-\log\log(mU)}\})$ maximum flow computations, or $O(k \min\{\log \Gamma, \frac{\log(m\Gamma U)}{1+\log\log(m\Gamma U)-\log\log(mU)}\})$ maximum flow computations and $2k$ $\{0, \pm 1\}$-minimum cost flow computations. These algorithms are discussed in section 3.

We rely on minimum cost flow and maximum flow algorithms developed by others. Our minimum cost flow computations have very special structure in that all arc costs are zero, except for some arcs leaving the source or entering the sink, which have costs in $\{1, -1\}$. Thus they can be solved more efficiently than general minimum cost flow problems—for example, using the cancel and tighten algorithm of Goldberg and Tarjan [13]. There are many efficient maximum flow algorithms. The most recent result is the $O(\min(n^{2/3}, m^{1/2}) m \log(\frac{n^2}{m}) \log U)$ time algorithm of Goldberg and Rao [11]. We will also use the $O(mn \log(n^2/m)$ push-relabel algorithm of Goldberg and Tarjan [12]. Other efficient algorithms are described in the book of Ahuja, Magnanti, and Orlin [1].

In a follow-up paper, we describe a faster algorithm for the minimum cost quickest transshipment problem [7], again assuming all transit times are zero.

1.3. The continuous-time model. Network flows over time have also been considered in the continuous-time setting [2, 3, 23, 25]. Most of the work in this area has examined networks with time-varying edge capacities, storage capacities, or costs. The focus of this research is on proving the existence of optimal solutions for classes

of time-varying functions and proving the convergence of algorithms that eventually find solutions. These algorithms fall short of being efficient, either theoretically or practically, and implementations do not seem able to handle problems with more than a few nodes. For the case in which capacity functions are constant, it is possible to extend the polynomial, discrete-time transshipment over time algorithm in [15] to work in the continuous-time setting [8].

A continuous transshipment over time is a flow $f : E \times T \to \mathbf{R}_{\geq 0}$ that varies over time. (We use $\mathbf{R}_{\geq 0}$ to denote the set of all nonnegative real numbers.) Let $x : E \times T \to \mathbf{R}_{\geq 0}$ be the rate of flow of f: $x(t) = df(t)/dt$, and let x_{ij} be x restricted to edge (i, j). We assume that each x_{ij} is a Lebesgue-measurable function on $(0, T]$. Here, the capacities u are upper bounds on the rate of flow through the arcs. The formulation is similar to the formulation of the discrete-time problem. We do not require that mass balance constraints be satisfied at equality before time T. Instead this formulation allows for storage at any node of the network, but does not allow deficit to exceed the initial deficit at the node:

$$\begin{array}{rcll} x(t) & \leq & u_e & \text{for all } 0 \leq t \leq T, \\ \int_0^r \sum_{j \in V} [x_{ij}(t) - x_{ji}(t)]dt & \leq & \max\{0, \gamma_i\} & \text{for all } r \in (0, T], \quad \text{for all } i \in V, \\ \int_0^T \sum_{j \in V} [x_{ij}(t) - x_{ji}(t)]dt & = & \gamma_i & \text{for all } i \in V. \end{array}$$

If this problem is feasible and T is integral, there is a solution f that changes only at times in $\{1, 2, \ldots, T\}$: A discrete-time solution can be transformed into a continuous-time solution by sending flow at rate $f(t)$ in the interval $(t-1, t]$. This transformation of the optimal discrete-time solution is optimal for the continuous-time problem [8]. Thus, the continuous-time problem is no harder than the discrete-time problem; the integral quickest transshipment algorithms mentioned in the preceding section and presented in this paper also solve the continuous-time problem.

1.3.1. Results on universally quickest transshipment. A *universally quickest transshipment* is a quickest transshipment that simultaneously minimizes the amount of excess left in the network at every moment of time. An optimal solution may require fractional flow sent over fractional intervals of time. There is a two source, two sink example for which a universally quickest transshipment does not exist (see Figure 4.1 in section 4). Hajek and Ogier [14] describe an algorithm that solves the universally quickest transshipment problem in networks with multiple sources and a single sink. Their algorithm uses $O(n)$ maximum flow computations. We describe how this problem can be solved in $O(mn \log(n^2/m))$ time—the same asymptotic time as the fastest implementation of the push-relabel maximum flow algorithm. This algorithm is described in section 4. Table 1.1 summarizes the work on polynomial time algorithms to solve various quickest transshipment problems.

1.4. Extensions. Everyday usage often involves continuous streams of traffic. All of the algorithms presented here, like the algorithm of Hajek and Ogier [14], allow for constant streams of flow into or out of any node in the network. The details are discussed in section 5.

The model presented above assumes that there is infinite buffer capacity at all nodes of the network. The model can be adapted to handle finite buffer capacity. We assume the minimum buffer capacity at node i is $\max\{0, \gamma_i\}$. If we allow additional finite storage a_i at i, then we have the additional constraint (in the discrete-time model) that $\sum_{t=1}^{r} \sum_j [f_{ij}(t) - f_{ji}(t)] \geq \min\{0, \gamma_i\} - a_i$ for all $r \in \{1, \ldots, T\}$ and all $i \in V$. However, the algorithms we use for both the discrete-time problems and the

TABLE 1.1

New and existing polynomial time algorithms for the multiple source, quickest transshipment problem. (Here, k is the number of terminals, Γ upper bounds the absolute value of any supply or demand, U equals the maximum edge capacity, and τ is the maximum transit time).

In	Multi-sink	Integral flow	Discrete time	Transit times	Universal	Run time
[14]					√	$O(n)$ maximum flows
[15]	√	√	√	√		$O(k^3 \log(nU\Gamma\tau)\log(nU\tau))$ minimum cost flows
§ 4					√	$O(1)$ maximum flows
§ 3.4		√	√			$O(k)$ maximum flows
§ 3.2	√	√	√			$O(k^2 \log(Un) + k \log \Gamma)$ maximum flows, or $O(km)$ maximum flows
§ 3.3	√	√	√			$O(k \log \Gamma)$ maximum flows + $2k$ min $(0, \pm 1)$-cost flows

continuous-time problems will find a flow over time that depletes all supplies in the minimal time and does not ever require more storage at node i than $\max\{0, \gamma_i\}$. That is, the optimal solution for the case when $a_i = 0$ for all $i \in V$ is also optimal when all $a_i = \infty$. This is also true with general transit times. However, once time-dependent capacities, or other generalizations, are introduced, buffer capacity may need to be utilized in an optimal solution [20, 6].

2. Computing the minimum time required for feasibility. All algorithms described in this paper rely on testing feasibility of transshipment over time. In the problem with transit times, Hoppe and Tardos [15] use submodular function minimization to resolve transshipment over time feasibility. Zero transit times make the problem much easier.

THEOREM 2.1. *Given time bound T, a feasible, fractional transshipment over time can be found with one maximum flow computation.*

Proof. A problem of transshipment over time is feasible in time T if and only if the static transshipment problem is feasible in the same network with edge capacities multiplied by T: Flow on any edge summed over the course of a feasible transshipment over time cannot exceed T times the capacity of the edge. A feasible static transshipment f can be transformed into a feasible transshipment over time by sending flow at rate f_{ij}/T through each arc (i, j) from time 0 until time T. □

The feasibility test described in the proof of Theorem 2.1 computes a flow in the network with capacities multiplied by T. The minimum time bound T^* is the smallest value of T for which the flow problem is feasible. This is a *parametric flow problem*: find the minimum value of parameter T such that the static transshipment problem with supplies γ and capacities uT is feasible.

Given $A \subset V$, let $o(A)$ denote the amount of flow that can be sent from sources inside A to sinks outside A in one unit of time. Since all transit times are zero, the amount of flow that can be sent from A to outside A in time T is $o(A)T$. The minimum time T^* required for feasibility is therefore defined by some cut A in the network with the property that $o(A)T^* = \gamma(A)$. Any set A with this property is called *tight*.

While T^* can be found by binary search, Radzik [27] describes a more efficient procedure for a slightly more general class of parametric flow problems: where ca-

pacities are of the form $a + bT$ for fixed vectors a and b in R^E and parameter T. One common name for the procedure he describes is the discrete Newton's method. Radzik shows that for parametric flow problems, it is possible to give an improved bound on the number of search iterations. Let A bound the maximum absolute value of components in γ and a, and B bound the maximum absolute value of components in B.

THEOREM 2.2 (see [27, Theorems 3.4 and 4.3]). *The parametric flow problem with supplies γ and capacities $a+bT$ can be solved using $O(\frac{\log(mAB)}{1+\log\log(mAB)-\log\log(mB)})$ maximum flow computations or $O(m)$ maximum flow computations.* □

COROLLARY 2.3. *The minimum time T^* required for feasibility of a transshipment problem over time can be computed using $O(\frac{\log(m\Gamma U)}{1+\log\log(m\Gamma U)-\log\log(mU)})$ maximum flow computations or $O(m)$ maximum flow computations.*

Theorem 2.1 implies that the maximum flow f for optimal time T^* yields an optimal fractional transshipment. A further challenge is to find a solution in which the flow rates (or flow amounts, in the discrete-time case) are integers. If we are looking for a completely integral solution, where the flow rates and the time intervals are integers, we can assume $T^* = \lceil T^* \rceil$. If T^* is integral, the transshipment over time problem is equivalent to a static transshipment in the time expanded graph, so standard network flow theory proves the existence of an integral solution. We describe how to construct an integral solution in the following section.

3. Quickest integral transshipment. We describe two variants of an algorithm to solve the integral quickest transshipment problem. The first variant requires $O(\min\{km, \frac{k\log\Gamma + k^2\log(mU)}{1+\log\log(m\Gamma U)-\log\log(mU)}\})$ maximum flow computations, while the second requires $k\min\{\log\Gamma, \frac{\log(m\Gamma U)}{1+\log\log(m\Gamma U)-\log\log(mU)}\}$ maximum flow computations and $2k$ $\{0,\pm1\}$-minimum cost flow computations. When there is only one sink, a modified version of the first algorithm solves the integral quickest transshipment problem with only $O(k)$ maximum flow computations.

3.1. The basic algorithm. All of the algorithms start off using a two-level network inspired by the time-expanded network. Unlike the time-expanded network, the *two-level network* consists of just two copies of the original network (Figure 3.1). The upper copy, $\mathcal{N}^U$, represents the first unit of time and has original arc capacities. The lower copy, $\mathcal{N}^L$, represents the remaining $T-1$ units of time and thus has its capacities multiplied by $T-1$. These two copies of the network are connected via an additional copy of each terminal, called a super-terminal, and denoted s_i^S, $i = 1$ to k. Let s_i^U be the copy of terminal s_i in $\mathcal{N}^U$, and let s_i^L be the corresponding terminal in $\mathcal{N}^L$. If s_i is a source node, then the two-level network contains arcs $e_i^U = (s_i^S, s_i^U)$ and $e_i^L = (s_i^S, s_i^L)$, each with infinite capacity. If s_j is a sink node, then the two-level network contains infinite capacity arcs $e_j^U = (s_j^U, s_j^S)$ and $e_j^L = (s_j^L, s_j^S)$. Node s_i^S is assigned supply γ_i and node s_j^S is assigned demand γ_j.

A static transshipment computation in the two-level network gives an integral flow, since all inputs are integral. The flow in $\mathcal{N}^U$ corresponds to the flow sent in the first unit of time. The flow in $\mathcal{N}^L$ indicates that the remaining supplies can be satisfied in time $T-1$. That is, this reduced transshipment-over-time problem with reduced supplies $f(e_i^L) = |\gamma_i| - f(e_i^U)$ at s_i is feasible in time $T-1$.

If we construct a new two-level network for this reduced problem and repeat the process, we then find an integral flow for the second unit of time and a further reduced problem feasible in time $T-2$. This leads to the following observation.

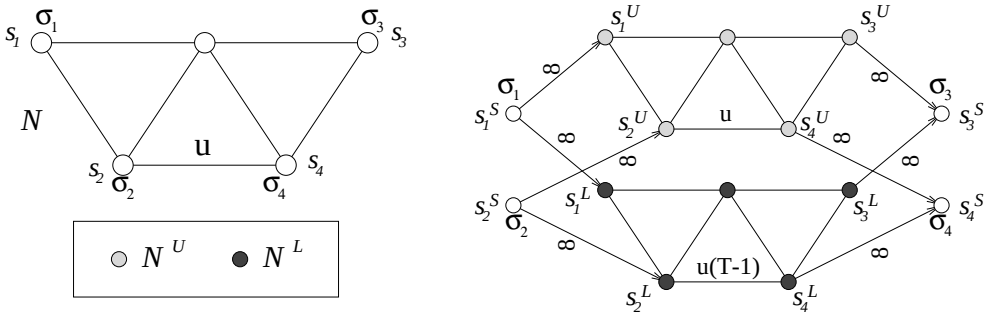


FIG. 3.1. *A network and the corresponding two-level network.*

OBSERVATION 3.1. *Given feasible time bound T, an integral transshipment over time can be found with $T-1$ maximum flow computations.*

To develop a more efficient algorithm, it is necessary to reduce the number of static transshipment problems considered. Let f^U be the flow f restricted to $\mathcal{N}^U$. Denote by $f^U(A)$ the net flow f^U leaving node subset A. That is $f^U(A) = \sum_{i\in A, j\notin A}[f_{ij} - f_{ji}]$. We would like to continue to use f^U for as long as the remaining supplies $\{\gamma_i - \lambda f^U(\{i\})\}_{i\in S}$ can be satisfied in the remaining time, $T^* - \lambda$. The problem with supplies $\{\gamma_i - \lambda f^U(\{i\})\}_{i\in S}$ and time bound $T^* - \lambda$ is called the λ-*reduced problem*. Here, λ is the amount of time we will use the first integral flow. Starting with $\lambda = 1$, we can increase λ either until the supply is depleted at some terminal (e.g., $\gamma_i - \lambda f^U(\{i\}) = 0$ for some i) or until the λ-reduced problem becomes infeasible. We seek the maximum λ such that the λ-reduced problem is feasible. This is a parametric flow problem of a form apparently more general than the problem considered in Theorem 2.2, since both supplies and capacities are linear functions of the parameter. We now show how to transform this parametric flow problem into an equivalent one of the form given in Theorem 2.2.

Let $T' := \min_{i:\gamma_i\neq 0} \frac{\gamma_i}{f^U(\{i\})}$. Note that $T' > 0$ since f^U has the same sign as γ. This T' is the minimum value of λ for which supply at some source or sink depletes. Thus we may assume that we search for a $\lambda \leq T'$. (If $T' > T$, set $T' = T$.) Consider the parametric flow problem parameterized by k with supplies $g_i := \gamma_i - T' f^U(\{i\})$ and capacities $u(T - T') + (u - f^U)k$. This problem is of the form considered in Theorem 2.2.

LEMMA 3.1. *Let k be the minimum value in the range $[0, T'-1]$ for which the parametric flow problem with supplies g_i and capacities $u(T-T') + (u - f^U)k$ is feasible. Then, the maximum value of λ in the range $[1, T']$ for which the parametric flow problem with supplies $\gamma_i - f^U(\{i\})\lambda$ and capacities $u(T-\lambda)$ is feasible equals $T' - k$.*

Proof. For any value of k, consider the set A that minimizes $\kappa(A) := [u(T-T') + (u - f^U)k](A) - g(A)$. This value equals $\kappa'(A) := [u(T-\lambda)](A) - \gamma(A) + \lambda f^U(A)$ for $\lambda = T' - k$, since $f^U(A) := \sum_{i\in A, j\notin A} f^U_{ij} = \sum_{i\in A} f^U(\{i\})$ by the fact that f^U is a flow. The first problem in Lemma 3.1 is feasible if $\kappa(A) \geq 0$, and the second problem is feasible if $\kappa'(A) \geq 0$. Since $\lambda = 1$ yields a feasible form of the second problem, this implies that $k = T' - 1$ is feasible for the first problem. Thus the minimum value of $k \in [0, T'-1]$ that makes the first problem feasible determines the maximum value $\lambda = T' - k$ that makes the second problem feasible. □

COROLLARY 3.2. *The maximum value of the parameter λ such that the parametric*

Basic $(\mathcal{N}, u, \gamma, T^*)$

$T = T^*$, $p = 0$.
while $\gamma \neq 0$,
  $p = p + 1$.
  Construct the two-level network with u, γ, T.
  Solve the transshipment problem on this two level network.
  f^p = flow through $\mathcal{N}^U$, the small capacity level.
  λ_p = maximum λ such that transshipment problem $(\mathcal{N}, u, \gamma - \lambda f^p, T - \lambda)$
    is feasible.
  $T = (T - \lambda_p)$.
  $\gamma = (\gamma - \lambda_p f^p)$.
end while
return $\{(f^1, \lambda_1), (f^2, \lambda_2), \ldots, (f^p, \lambda_p)\}$.

FIG. 3.2. *The basic quickest transshipment algorithm.*

flow problem with supplies $\gamma_i - \lambda\tilde{f}(\{i\})$ *and capacities* $u(T - \lambda)$ *is feasible can be found* $O(\frac{\log(m\Gamma U)}{1+\log\log(m\Gamma U)-\log\log(mU)})$ *maximum flow computations, or* $O(m)$ *maximum flow computations.*

Proof. By definition of T', $0 \leq |g_i| \leq |\gamma_i|$. Since the most flow through an arc is the total supply, which is at most $k\Gamma \leq m\Gamma$, we can assume $u(T - T')$ is bounded by $m\Gamma$. Also, U remains a bound on the multiplier of k. Plugging these bounds into the statement of Theorem 2.2, we see that we obtain a factor of $\log(m^2\Gamma U) = O(m\Gamma U)$. Thus, together Lemma 3.1 and Theorem 2.2 imply the corollary. ∎

Figure 3.2 summarizes the algorithm. The quickest integral transshipment takes as input a network with arc capacity vector u, vertex supply vector γ, and time bound T, and returns a set of flow-duration pairs $\{(f^p, \lambda_p)\}_{p=1}^r$ that specify the flows f^p that compose the optimal transshipment and the duration of time λ_p for which they continue.

The algorithm repeatedly finds integral flows and their corresponding time periods until all supplies are exhausted. Each time we fix a λ, one of two things happens. Either supply at some terminal is exhausted, or there is some new set of nodes A whose total supply is equal to the amount of flow that can leave this set in the current time bound. That is, A is tight, as defined in section 2, and it wasn't tight previously. We will refer to such a tight set as a *new tight set*. The flow leaving A must be at a maximum for the remainder of the algorithm: once a set is tight, it remains tight. The following sequence of lemmas shows that the algorithm searches for at most k values of λ before all supplies are exhausted.

LEMMA 3.3. *The intersection and union of tight sets are tight.*

Proof. Recall $o(A)$ is the amount of flow that can leave set A in one unit of time. Megiddo [17] shows that $o(A) + o(B) \geq o(A \cup B) + o(A \cap B)$. Any set function satisfying this inequality for all subsets A and B is a *submodular function*. Since $\gamma(A) = \sum_{i \in A} \gamma_i$ satisfies this inequality at equality, it is easy to see that $o(A)T - \gamma(A)$ is also submodular. For a tight set A, $o(A)T - \gamma(A) = 0$ by definition. Since T is feasible if and only if $o(A)T - \gamma(A) \geq 0$, the submodular inequality implies that the intersection of tight sets is tight and the union of tight sets is tight for any feasible T. ∎

Observe that it is possible for a new tight set to be a subset of an existing tight set, and for sources in a tight set to run out of supply before time T: It could be that the total flow out of a tight set B must be at its maximum possible for the remainder of the algorithm, but that under the current flow, some sources in B are sending out flow at a rate that will deplete their supply before time T, while others are sending out flow at a rate slower than needed to deplete their supply by time T. In this scenario, the set of sources in B that are sending out flow too quickly either will be contained in a tight set $A \subset B$ found later in the algorithm or will determine λ at some point when they run out of supply. At this point, the rates of flow out of these sources will decrease, and the rates of flow out of the remaining sources in B will increase.

Define a *chain of sets* to be a sequence of nested sets: each set is strictly contained in its successor.

LEMMA 3.4. *Each time the algorithm forces a new set to become tight, the number of nested tight sets in the largest chain increases by at least one.*

Proof. When we find a new tight set of terminals, we know the old tight sets are still tight. Let N be the new tight set, and let $B_0 \subset B_1 \subset \cdots \subset B_r$ be the existing chain of nested tight sets, with $B_0 = \emptyset$. If j is an index such that $N \cap (B_{j+1} \backslash B_j)$ is a proper subset of $B_{j+1} \backslash B_j$, then Lemma 3.3 implies $N' = N \cap B_{j+1}$ is a tight set and thus $N' \cup B_j$ also is a tight set. This implies that $B_0 \subset \cdots \subset B_j \subset B_j \cup N' \subset B_{j+1} \subset \cdots \subset B_r$ is a larger chain of nested tight sets.

We now show that there is at least one index j such that $N \cap (B_{j+1} \backslash B_j)$ is a proper subset of $B_{j+1} \backslash B_j$. First, note that for any j, $H_j := B_{j+1} \backslash B_j$ either is tight before this iteration or is never tight throughout the remainder of the algorithm. This is because the amount of flow leaving $B_{j+1} \backslash B_j$ in each time unit is fixed at $o(B_{j+1}) - o(B_j)$ for the remainder of the algorithm. Similarly, the amount of flow leaving $\cup_{j \in J} H_j$ for any subset of indices J is fixed at $\sum_{j \in J} o(B_{j+1}) - o(B_j)$ for the remainder of the algorithm. Hence $\cup_{j \in J} H_j$ cannot be a new tight set. This implies that $N \neq \cup_{j \in J} H_j$ for any subset J. Thus, there is some index j for which $N \cap (B_{j+1} \backslash B_j)$ is a proper subset of $B_{j+1} \backslash B_j$. □

A terminal is *active* if it has nonzero supply remaining. Let A represent the set of active terminals. Let C represent the chain of tight sets encountered by the algorithm. We prove below that each iteration of the algorithm decreases $|A| - |C|$ by at least one. Thus after at most k iterations, we have a complete chain of nested tight sets, and the flow is constant for the remaining time.

LEMMA 3.5. *The basic algorithm searches for a new feasible integral flow f^p at most k times.*

Proof. Each time we find a new λ by finding a new tight set, $|A| - |C|$ decreases by one: $|A|$ does not change and by Lemma 3.4, $|C|$ increases.

Each time we deplete a supply, $|A|$ decreases by one. If $|C|$ decreases by 1, then $|A|$ decreases by at least 2: if the difference between two tight sets of terminals is just one terminal, then the flow rate out of that terminal must be constant for the remaining time. If such a terminal is active, it must be active until the end. If $B_j \subset B_{j+1}$ are consecutive tight sets that collapse into one tight set, then $B_{j+1} \backslash B_j$ must contain at least two terminals (one source and one sink) that deplete at the same time.

Since there are at most k terminals, the total number of times the algorithm stops with a nonempty reduced problem is at most k. (It could be that two independent events occur simultaneously, causing the algorithm to complete after less than k iterations.) □

There is the problem that some of the λ_p we find may be fractional. The next

two sections present two different approaches to handling this problem.

3.2. A continuous-time approach to the quickest transshipment problem. In this section, we solve the quickest transshipment problem by first producing a solution with constant, integral rates of flow over time periods of arbitrary lengths. We then transform this continuous-time solution into a discrete-time integral transshipment with at most k more maximum flow computations. The solution is then fully integral, which is of interest for applications that can handle only integer quantities.

We apply the algorithm of the previous section. This could give us not only fractional λ_p but also fractional f^p, since after one iteration, the remaining supplies may be fractional. To avoid this, after one computation of λ, we rescale the remaining problem so that all data is integer. For iteration p, this involves multiplying remaining supplies and time bound by the denominator d_{p-1} of λ_{p-1} to obtain an equivalent, but integral problem: Any flow g for the original problem can be transformed into a flow satisfying the multiplied problem by sending each amount of flow for a period that is d_p times longer than the time it is sent in the original problem. Any integer solution to the multiplied problem can be transformed into a feasible, integral flow for the original problem by reducing the *time* any particular flow is sent by a factor of d_p. This means we can find a solution where the flow rates are integral, but the intervals during which they are sent are fractional.

If all λ's are integers, then the continuous-time solution is easily transformed into a discrete-time solution by sending f_{ij} units of flow on arc (i, j) at each of λ time units. If there are fractional λ's, then we subtract the fractional part from each λ to get a partial solution that is integral and hence satisfies an integral amount of the supplies. Since our initial supplies are integers, the sums of the supplies sent in the fractional times are also integers, and they can be satisfied in time $\Delta = \sum_{p=1}^{k} \lambda_p - \lfloor \lambda_p \rfloor < k$. We can now use Observation 3.1 to solve this smaller transshipment over time problem with $< k$ maximum flow computations. These two partial solutions scheduled one after the other form a feasible, integral, discrete transshipment over time completing by the optimal time.

THEOREM 3.6. *A fully integral solution for the quickest transshipment problem with zero transit times can be found with* $O(\min\{km, \frac{k \log \Gamma + k^2 \log(mU)}{1+\log\log(m\Gamma U)-\log\log(mU)}\})$ *maximum flow computations.*

Proof. By Corollary 3.2, at most $O(\min\{m, \frac{\log(m\Gamma U)}{1+\log\log(m\Gamma U)-\log\log(mU)}\})$ maximum flow computations are required to find λ_1. Note that λ_1, and in general λ_p, may be fractional with denominator $d_p \leq Um$. This latter quantity is a bound on the size of a cut in the original graph. After the first iteration, we multiply remaining supplies by at most d_1. Let d_r be the denominator of λ_r, $r < j$. The supply vector in the parametric flow problem for λ_j is bounded by $d_1 d_2 \cdots d_{j-1}\Gamma \leq (Um)^{j-1}\Gamma$. Thus the number of maximum flow computations required to find λ_j is at most $O(\min\{m, \frac{\log \Gamma + j \log(mU)}{1+\log(\log \Gamma + j \log(mU))-\log\log(mU)}\})$. Summing over all $p = 1, \ldots, k$, the total number of maximum flow computations to find all λ_p, $p = 1, \ldots, k$ is $O(\min\{km, \frac{k \log \Gamma + k^2 \log(mU)}{1+\log\log(m\Gamma U)-\log\log(mU)}\})$. ∎

3.3. A minimum cost flow approach to the quickest transshipment problem. In this section, we demonstrate how the quickest integral transshipment can be solved without creating parametric flow problems with very large supply vectors, and hence large input sizes. We start with the basic algorithm as described in section 3.1, constructing the two-level network with capacities determined by the remaining time T and computing a static transshipment f in this network. Again, we

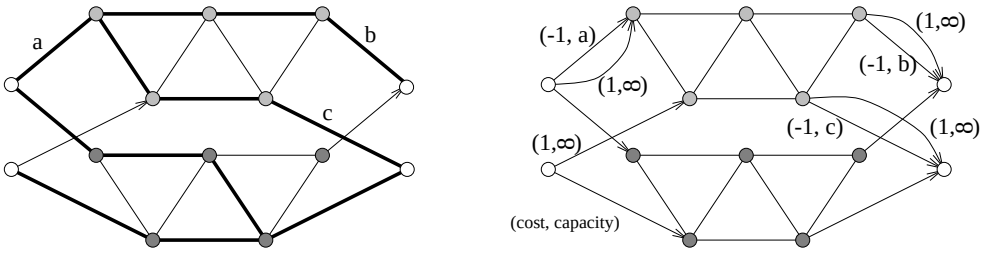


FIG. 3.3. *Flow f represented by thicker lines on the left and, on the right, the two-level network for the corresponding minimum cost flow problem.*

seek to repeat the flow described by f restricted to the upper network $\mathcal{N}^U$, denoted f^U, as long as the remaining problem is feasible. However, this time we insist f^U is repeated for an integer number of periods. In the basic algorithm in the previous section, we calculated the maximum time λ that we could send flow f^U, and we sent the flow for this period of time. Now, we will send f^U for only $\lfloor\lambda\rfloor$ units of time. This avoids creating new parametric flow problems with supply vectors increasing arithmetically in size. But with this restriction, there may not be a new tight set at time $T - \lfloor\lambda\rfloor$. Recall that in section 3.1, our progress is measured by the number of nested tight sets created. We modify the algorithm to address this problem as follows.

We describe a procedure using a minimum cost flow computation that will force a new set to become tight in one additional unit of time, after sending f^U for $\lfloor\lambda\rfloor$ units of time. We observe that since repeating f^U for one additional time unit is infeasible, there must be some set of terminals that is close to being tight. We try to force such a set to become tight by sending either as little out of the set, or as much into the set, as possible in the next time unit, while keeping the remaining problem feasible.

To do this, we use one minimum cost flow computation in a modified two-level network. We construct a minimum cost flow problem that encourages sending as much of f^U as possible and as little additional flow as possible: We place the remaining supplies and demands at the terminals of a two-level network with capacities determined by remaining time $T - \lambda$ as described in section 3.1. All arcs in the network are assigned cost 0. For each source in $\mathcal{N}^U$, we reduce the capacity of the arc from the super-source to be the amount of flow f^U leaving the source in $\mathcal{N}^U$, and assign this arc a cost of -1. We also add an infinite capacity, cost 1 arc from the super-source to each source in $\mathcal{N}^U$. We make similar additions and adjustments of arcs from the sinks in $\mathcal{N}^U$ to the super-sinks. Figure 3.3 gives an example of this modification. We prove below that after sending the part of the minimum cost flow restricted to $\mathcal{N}^U$, $|A| - |C|$ is reduced by at least one.

THEOREM 3.7. *A solution for the integral quickest transshipment with zero transit times can be computed with $2k$ $\{0, \pm 1\}$-minimum cost flow computations and $k \min\{\log \Gamma, \frac{\log(m\Gamma U)}{1+\log\log(m\Gamma U)-\log\log(mU)}\}$ maximum flow computations.*

To prove Theorem 3.7, we show that after sending the minimum cost flow restricted to $\mathcal{N}^U$, $|A| - |C|$ is reduced by at least one. Since each decrease in $|C|$ implies at least a decrease of two in $|A|$ (see proof of Lemma 3.5) and $|A|$ never increases, this amounts to showing that either $|A|$ decreases or $|C|$ increases. Lemma 3.5 then implies the theorem.

Let x be the minimum cost flow in the modified network. Standard optimality conditions for minimum cost flows [1] imply that there exists a set of labels π on

the vertices in the two-level network that obey the following *complementary slackness* conditions. Define $c^{\pi}_{ij} := c_{ij} - \pi(i) + \pi(j)$ for all arcs (i, j) in the two-level network. Then

$$x_{ij} < u_{ij} \Rightarrow c^{\pi}_{ij} \geq 0,$$
$$x_{ij} > l_{ij} \Rightarrow c^{\pi}_{ij} \leq 0.$$

If some source or sink that is active at time $T - \lambda$ does not send or receive any flow in $\mathcal{N}^L$, then it has been emptied in $\mathcal{N}^U$ and thus $|A|$ decreases by one.

We must now show that $|A| - |C|$ decreases even if all active sources and sinks send some flow in $\mathcal{N}^L$. We show this by finding a new tight set of terminals. Thus by Lemma 3.4, this implies that $|C|$ increases by one.

We first suppose that there is nonzero flow on a cost 1 edge entering a source. For terminal s_j, let e^U_j denote the original arc connecting the terminal in $\mathcal{N}^U$ and super-terminal. Let e^U_{j-} and e^U_{j+} denote the two arcs that replace e^U_j in the minimum cost flow problem, of cost -1 and $+1$, respectively.

LEMMA 3.8. *If all active terminals send or receive flow in $\mathcal{N}^L$ and $s_* \in S^+$ is such that $x(e^U_{*+}) > 0$, then $L_0 := \{v_i \in V | \pi(v^L_i) \geq \pi(s^L_*)\}$ is a new tight set.*

Proof. Since c^{π} is unchanged by adding the same constant to every label, we can assume without loss of generality that $\pi(s^L_*) = 0$. Thus $L_0 = \{v_i \in V | \pi(v^L_i) \geq 0\}$. Similarly, define $U_0 := \{v_i \in V | \pi(v^U_i) \geq 0\}$. The proof consists of establishing a sequence of statements that build on each other, ending in the claim of the lemma.

(i) If $s_j \in L_0 \cap S^+$ sends out less x-flow in $\mathcal{N}^U$ than it sent out f-flow, then $s_j \in U_0$.
(ii) If $s_j \in L_0 \cap S^-$ receives more x-flow in $\mathcal{N}^U$ than it received f-flow, then $s_j \in U_0$.
(iii) If L_0 is tight in $\mathcal{N}^U$, then $L_0 \cap U_0$ is tight in $\mathcal{N}^U$.
(iv) $s_* \notin U_0$.
(v) Either L_0 is not tight in $\mathcal{N}^U$ or L_0 sends out strictly more x-flow in $\mathcal{N}^U$ than it sends out f-flow.
(vi) L_0 is a new tight set in $\mathcal{N}^L$.

In the arguments below, we repeatedly use complementary slackness and the assumption that all active terminals send or receive flow in $\mathcal{N}^L$. For instance, the capacity of all arcs connecting super-terminals to terminals in $\mathcal{N}^L$ is infinite, so $c^{\pi} = 0$ for each of these arcs. Since $c = 0$ by design for each of these arcs, $\pi(s^L_i) = \pi(s^S_i)$ for all active terminals s_i.

(i) If $x(e^U_{j-}) < f(e^U_j)$, complementary slackness and $s_j \in L_0 \cap S^+$ imply $\pi(s^U_j) \geq \pi(s^S_j) + 1 = \pi(s^L_j) + 1 \geq 1$.
(ii) If $x(e^U_{j+}) > 0$, complementary slackness and $s_j \in L_0 \cap S^-$ imply $\pi(s^U_j) = \pi(s^S_j) + 1 = \pi(s^L_j) + 1 \geq 1$.
(iii) Complementary slackness implies all arcs leaving U_0 in $\mathcal{N}^U$ are at full capacity and all arcs entering U_0 in $\mathcal{N}^U$ are empty. Thus U_0 is tight in $\mathcal{N}^U$. Since the intersection of tight sets is tight (Lemma 3.3), L_0 tight in $\mathcal{N}^U$ implies that $L_0 \cap U_0$ is also tight in $\mathcal{N}^U$.
(iv) Using complementary slackness, $\pi(s^U_*) = \pi(s^S_*) - 1 = \pi(s^L_*) - 1 = -1$.
(v) If L_0 is tight in $\mathcal{N}^U$, then (iii) implies that $L_0 \cap U_0$ is tight in $\mathcal{N}^U$. Thus in $\mathcal{N}^U$, (i) and (iii) together imply that for all $i \in L_0 \cap S^+$, the x-flow out of i is $\geq$ the f-flow out of i. Similarly, in $\mathcal{N}^U$, (ii) and (iii) imply that for all

$j \in L_0 \cap S^-$, the x-flow into j is $\leq$ the f-flow into j. Thus L_0 sends out at least as much flow in $\mathcal{N}^U$ as it does with f. By (iv), $s_* \in L_0 \backslash U_0$. Since s_*^U sends out more flow with x than s_* does with f, L_0 actually sends out more flow in $\mathcal{N}^U$ than it does with f.

(vi) By complementary slackness, all arcs leaving L_0 in $\mathcal{N}^L$ are at full capacity and all arcs entering L_0 in $\mathcal{N}^L$ carry no flow. Thus, L_0 is tight in $\mathcal{N}^L$. If L_0 is not tight in $\mathcal{N}^U$, it is clearly a new tight set. Otherwise L_0 is sending out more flow in $\mathcal{N}^U$ than it did with f. Thus, it was not tight for flow f, and hence is newly tight. □

If the only cost 1 arcs used are those leaving sinks, consider the *reverse* network obtained by reversing the direction of arcs and flow x, and multiplying all labels by -1. The new labels and new flow are feasible and satisfy the complementary slackness conditions and hence are optimal. Since there is flow entering a source on a cost 1 arc in this reverse network, Lemma 3.8 implies that there is a new tight set L_0 in this reverse network. Thus $\overline{L_0}$ is a new tight set in the original graph.

If no cost 1 edge is used, then since f^U is not feasible, there is a cost -1 edge adjacent to a source that is not at full capacity. The following lemma shows that there is also a new tight set of terminals in this case.

LEMMA 3.9. *If all active terminals send or receive flow in $\mathcal{N}^L$ and $s_* \in S^+$ is such that $x(e_{*-}^U) < f(e_*^U)$, then $L_0 := \{v_i \in V | \pi(v_i^L) \geq \pi(s_*^L)\}$ is a new tight set.*

Proof. As in Lemma 3.8, we assume $\pi(s_*^L) = 0$, and we show that $L_0 = \{v_i \in V | \pi(v_i^L) \geq 0\}$ is a new tight set. The proof of this proceeds along the lines of the proof of Lemma 3.8, establishing the following sequence of statements using similar ideas. Because of its similarity to the proof of Lemma 3.8, we leave the details of the proof to the reader. As before, define $U_0 := \{v_i \in V | \pi(v_i^U) \geq 0\}$.

(i) If $s_j \in L_0 \cap S^-$ receives less x-flow in $\mathcal{N}^U$ than it receives f-flow, then $s_j \in U_0$.

(ii) If $s_j \in L_0 \cap S^+$ sends out more x-flow in $\mathcal{N}^U$ than it sends out f-flow, then $s_j \in U_0$.

(iii) If L_0 is tight in $\mathcal{N}^U$, then $L_0 \cap U_0$ is tight in $\mathcal{N}^U$.

(iv) $s_* \notin U_0$.

(v) Either L_0 is not tight in $\mathcal{N}^U$ or L_0 receives strictly less x-flow in $\mathcal{N}^U$ than it receives f-flow.

(vi) L_0 is a new tight set in $\mathcal{N}^L$. □

Proof of Theorem 3.7. Each iteration of the altered basic algorithm uses one minimum cost flow computation, reducing $|A| - |C|$ by at least one. The search for λ_p can be performed using either Radzik's parametric flow algorithm or binary search. In the former case, this requires $O(\min\{m, \frac{\log(m\Gamma U)}{1+\log\log(m\Gamma U)-\log\log(mU)}\})$ maximum flow computations, by Corollary 3.2. In the latter case, this requires at most $\log \lambda_p$ maximum flow computations. Since Γ is an upper bound on λ_p for all p, the total number of maximum flow computations required is $k \min\{\log \Gamma, \frac{\log(m\Gamma U)}{1+\log\log(m\Gamma U)-\log\log(mU)}\}$. □

3.4. The single sink case. When there is only one sink, the transshipment over time problem is also known as the evacuation problem. In this section, we explain how to compute a quickest integral evacuation with $O(k)$ maximum flow computations. The best previous solution for computing a quickest integral evacuation is the algorithm of Hoppe and Tardos [15] which uses the ellipsoid method as a subroutine. The algorithm of Hajek and Ogier [14] computes a fractional solution using $O(n)$ maximum flow computations.

We start with the basic algorithm described in section 3.1 and observe that, since there is only one sink, the general parametric flow problem that we have to solve

to find the maximum feasible λ is simpler. Namely, we can replace Radzik's [27] parametric flow algorithm or binary search for the maximum λ with a parametric flow algorithm of Gallo, Grigoriadis, and Tarjan [10]. This parametric flow algorithm is based on the push-relabel maximum flow algorithm of Goldberg and Tarjan [12] and runs in the same asymptotic time: $O(mn \log(n^2/m))$.

The simpler parametric flow problem is defined on the original network with only one new node: a super-source—the single sink obviates the need for a super-sink. Now the arcs from super-source to source i have capacity $\gamma_i - \lambda \tilde{f}(\{i\})$, and all original network arcs have capacity $u_{ij}(T - \lambda)$. Gallo, Grigoriadis, and Tarjan [10] show that if the capacities on arcs leaving the source are nonincreasing functions of a parameter and all other capacities are constant, then the minimum value of the parameter for the source to be a minimum cut can be found in the same asymptotic time as computing one maximum flow with the push-relabel algorithm. In our example, let $\alpha = T - \lambda$. Dividing all capacities by α gives us an equivalent problem with the original constant capacities on the original arcs, and capacities which are nonincreasing functions of $1/\alpha$ on the arcs leaving the source. This is the form required by the parametric flow algorithm of Gallo, Grigoriadis, and Tarjan. Thus, for the special case of a single sink, we find one λ_p in the same asymptotic time as one push-relabel maximum flow computation. Since there may be at most k iterations to find all flows and intervals, the quickest transshipment problem—also known as the quickest evacuation problem—can be solved in the same asymptotic time as $O(k)$ push-relabel maximum flow computations.

THEOREM 3.10. *The quickest evacuation problem with zero transit times can be solved in the same asymptotic time as k push-relabel maximum flow computations.* □

4. Universally quickest transshipment. The universally quickest transshipment is a flow that minimizes the amount of excess left in the network at every moment of time. Hajek and Ogier [14] solve this problem with $O(n)$ maximum flows when there is only one sink. There is a two-source, two-sink example for which a universally quickest transshipment does not exist (Figure 4.1). In this section, we describe an algorithm that solves the single sink, universally quickest transshipment in the same asymptotic time as one maximum flow computation.

If there are multiple sinks, a universally quickest solution may not exist. For example, consider the network in Figure 4.1. In one unit of time it is possible to satisfy four units of demand by sending one unit of supply from each source to each sink (Figure 4.1(a)). In this case, in the second time unit, it is possible only to send supply from the first source to the second sink, and the amount is restricted to one unit by the capacity of the arc. Thus a total of 3 time units are necessary to satisfy all demands. Consider instead Figure 4.1(b), where the maximum amount of supply is not sent in the first time unit. In particular the second source does not send any flow to the first sink. In this case, all demands can be satisfied in two time units. This example shows that it may not be possible to maximize the total satisfied demand at every time unit if there are multiple sources and sinks. The problem with multiple sinks is that, in the rush to send flow, some source may send flow to the wrong sink. This is not a problem when there is only one sink.

The universally quickest transshipment algorithm described here will find a series of subsets of V, $A_1 \subset A_2 \subset \cdots \subset A_r$, such that each set contains at least one more source than the previous set, and A_r contains all sources but not the sink. Each set will have a corresponding time bound T_i with $T_1 > T_2 > \ldots > T_r$, such that, in the

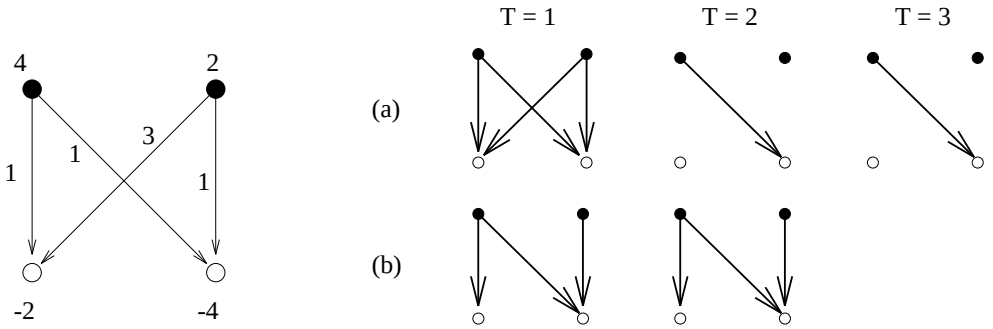


FIG. 4.1. *An example with no universally quickest transshipment.*

constructed solution, the rate of flow leaving set A_i equals $o(A_i)$ from time 0 until time T_i after which all sources in $A_i \backslash A_{i-1}$ are empty.

THEOREM 4.1. *A single sink transshipment over time that sends flow out of node set A_i at rate $o(A_i)$ from time 0 until time T_i, after which all sources in $A_i \backslash A_{i-1}$ are empty of supply, for all A_i and T_i as defined above, is a universally quickest transshipment.*

In the following proof and algorithm, we use the following notation: Given a flow-over-time network $\mathcal{N}$, a set of sources with supplies, and a time bound T let G_T be the graph obtained from $\mathcal{N}$ by multiplying the capacity of all arcs by T and adding a super-source with arcs from the super-source to all sources of capacities equal to the supply at the sources.

Proof. By the same argument in the proof of Theorem 2.1, the maximum amount of supply that can be sent to the sink by time T equals the maximum static flow sent to the sink in G_T. Represent this maximum quantity by $s(T)$. This is also the value of a minimum cut in the time-expanded network. A universally quickest transshipment must therefore send the supply of $s(T)$ to the sink by time T for every $0 < T \leq T^*$. We show that a transshipment over time obeying the conditions of the theorem achieves this.

Proof is by induction on c, the number of sets A_i. If $c = 1$, then A_1 contains all sources, and flow leaving A_1 equals $o(A_1)$ from time 0 until time T_1. This is clearly a universally quickest evacuation.

Suppose the theorem is true for $c < r$, and consider the case when $c = r$. Now consider the time bound T_r corresponding to the set A_r that contains all sources. By definition of T_r and A_r, a minimum cut in G_{T_r} contains A_r. Because A_r contains all sources, only arcs in the original network $\mathcal{N}$ cross this cut; all arcs leaving the source have their endpoints in the source side of this cut. Hence, this cut is also a minimum cut in G_T for every $T \leq T_r$. So the flow over time that sends flow out of A_r at rate $o(A_r)$ at every moment $0 \leq \theta \leq T_r$ is a universally quickest flow in this time interval.

By definition, after time T_r, all sources in $A_r \backslash A_{r-1}$ are empty of supply. In addition, since $T_i > T_r$ for all $i < r$, up until time T_r all other sets A_i $(i < r)$ have been sending flow out at a rate equal to $o(A_i)$. Now consider the altered problem with no supplies at the sources in $A_r \backslash A_{r-1}$. By induction, the flow that sends flow out of all A_i at rate $o(A_i)$ up until time T_i for $i = 1$ to $r - 1$ is a universally quickest transshipment. In particular, it is universally quickest at all times after T_r. Before time T_i, the flow on all arcs entering or leaving $A_i \backslash A_{i-1}$ is fixed. Thus, the flow within $A_i \backslash A_{i-1}$ can be determined independently in this time period. This implies

that any flow that maintains the appropriate flow conservation constraints at all nodes in $A_i \backslash A_{i-1}$ for $1 \leq i < r$ for all times $T \leq T_r$ is equivalent to any other. In particular, the flow computed with supplies in $A_r \backslash A_{r-1}$ and the flow computed without supplies in $A_r \backslash A_{r-1}$ are interchangeable on A_r for the interval $[0, T_r)$. Hence, the output of the algorithm on the original problem with r sets is also a universally quickest evacuation. □

We demonstrate the existence of a flow obeying the conditions of Theorem 4.1 constructively. Our algorithm has three stages: (a) It first finds sets A_i using the parametric flow algorithm of Gallo, Grigoriadis, and Tarjan [10], which also returns the corresponding time bounds T_i. Here, A_i is a minimum cut in G_{T_i}. (b) This information is then used to construct a static flow which represents the initial flow rate. (c) Finally, the flow over time is constructed from this static flow by reducing the flow rate along source-sink paths when the supplies at the corresponding sources are depleted.

(a) Finding nested sets A_i and time bounds T_i. The proof of Theorem 4.1 shows the need to compute maximum flows on $G_{\hat{T}}$ for values of $\hat{T} \in [0, T]$. An equivalent problem is to find maximum flows on the network with original capacities on original arcs and capacities equal to supply divided by T on the arcs from the super-source to the original sources. Gallo, Grigoriadis, and Tarjan [10] notice that the value of the maximum flow when the only arcs that are parameterized by T leave the source, as is the case for this latter problem, is a piecewise linear function of T with at most k breakpoints. They also show that the associated minimum cuts form a nested set of cuts. They describe an algorithm to find these breakpoints and cuts in $O(nm \log(n^2/m))$ time. Let the T_i be the values of T at these breakpoints and define A_i to be the minimum cut for T in interval $[T_{i+1}, T_i)$. Using the algorithm of Gallo, Grigoriadis, and Tarjan, all A_i and T_i are found in time $O(nm \log(n^2/m))$, i.e., the same asymptotic time as one maximum flow computation.

(b) Constructing the static flow. Gallo, Grigoriadis, and Tarjan [10, Theorem 4.1] show that there exists a flow in the network G_T with $f(s) = \gamma(s)/T_i$ for all sources s in $A_i \backslash A_{i-1}$ for all i simultaneously. They give an $O(mn \log(n^2/m))$ algorithm to compute such a flow. This is the initial flow used by our algorithm.

(c) Constructing the flow over time. Given the static flow f obtained in (b), we compute a universally quickest transshipment by computing, for time interval $[0, T_r)$ and for each time interval of form $[T_i, T_{i-1})$, a static flow that defines the flow rate in each interval. To start, $f_r = f$ is the flow rate from time 0 until time T_r. At time T_i, the supplies of the sources in $A_i \backslash A_{i-1}$ are depleted, so f_i is reduced by the flow leaving these sources to form f_{i-1}. To compute these successive static flows, decompose f into paths and cycles. Let f_i denote the desired flow rate in interval $[T_{i+1}, T_i)$, with $f_r = f$. Once f_{i+1} is obtained, f_i can be computed by reducing flow along paths with positive flow leaving sources in $A_i \backslash A_{i-1}$.

Ford and Fulkerson explain how a flow decomposition can be computed efficiently in $O(mn)$ time [9]: Until no arc carries flow, find a simple cycle or source-sink path with flow, and subtract the maximum amount of flow possible from this path or cycle. Each subtraction reduces the flow on some arc to 0, so there are at most m subtractions. Finding a simple source-sink path or cycle takes at most $O(n)$ time, as does subtracting the flow.

Note that all three steps of the algorithm run in time asymptotic to one maximum flow computation. Thus the run time of the algorithm is established.

THEOREM 4.2. *The single sink, universally quickest transshipment problem*

can be solved in the same asymptotic time as one push-relabel maximum flow computation. □

5. Incoming and outgoing traffic. While solving a transshipment problem with fixed supplies is useful for clearing a network after a communication breakdown, everyday usage more often involves continuous streams of traffic. The algorithms presented here, like the algorithm of Hajek and Ogier [14], allow for constant streams of flow into or out of any node in the network.

If there are constant streams of flow into and out of the nodes, the sum of the rates of these flows must equal zero in order for the problem to remain stable. Before solving the transshipment over time part of the problem, we can determine the course of this flow with one maximum flow computation.

Let ε_i be the rate of external flow into node i. Introduce a super-source connected to all nodes i with incoming flow by arcs with capacity ε_i. Similarly, introduce a super-sink, and connect all nodes j with outgoing flow to the super-sink with arcs of capacity $-\varepsilon_j$. If the maximum flow has value strictly less than the sum of the rates of incoming flow, then excess will build up in the network, and the problem is infeasible. Otherwise, the maximum flow determines the course these external flows will take through the network. The residual network of this flow, i.e., the network of arcs e with capacities $u'_e = u_e - f_e$, is passed on to any of the previously described algorithms.

THEOREM 5.1. *Any algorithm that solves the quickest transshipment problem or the universally quickest transshipment problem, can also solve the corresponding problem with constant streams of flow into and out of any node in the network.*

Proof. Define $\varepsilon(A) = \sum_{i \in A} \varepsilon_i$ to be the rate of external flow into set A, and consider a fixed time T. In order for the original problem to be feasible in time T, the total flow that can leave any set A in time T must be at least as great as the total external flow that enters A in time T, plus the total supply in A. That is, the problem is feasible in time T if and only if $o(A)T \geq \gamma(A) + \varepsilon(A)T$ for all $A \subset V$. Define A to be *snug* if A satisfies this inequality at equality. Now consider a minimum time bound T^* computed by one of the algorithms in the paper, on a residual network. For example, suppose T^* is the minimum time in which the residual network can be emptied of excess supply, computed as described in section 2. T^* is constrained by some tight set A with the property that $o'(A)T^* = \gamma(A)$, where $o'(A)$ is the sum of all residual capacities u' of edges leaving A. Call this set of edges I. In the original network, A is snug: $o(A)T^* = [o'(A) + \sum_{e \in I} u_e - u'_e]T^* = \gamma(A) + \sum_{e \in I} f_e T^* = \gamma(A) + \varepsilon(A)T^*$. Hence the T^* is also constrained by A, a snug set, in the original problem, and thus remains minimum. □

Acknowledgments. I am grateful to Éva Tardos for helpful comments on various drafts of this paper, for pointing out the much simpler upper bound in section 3.2, and for an observation simplifying the algorithm in section 3.3. I am also grateful to Tom McCormick for very helpful comments and for pointing out the reference [27], which led to improved run time bounds for several algorithms discussed in the paper.

REFERENCES

[1] R. K. AHUJA, T. L. MAGNANTI, AND J. B. ORLIN, *Network Flows: Theory, Algorithms, and Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1993.

[2] E. J. ANDERSON AND P. NASH, *Linear Programming in Infinite-Dimensional Spaces*, John Wiley, Chichester, 1987.

[3] E. J. Anderson and A. B. Philpott, *A continuous-time network simplex algorithm,* Networks, 19 (1989), pp. 395–425.
[4] J. E. Aronson, *A survey of dynamic network flows*, Ann. Oper. Res., 20 (1989), pp. 1–66.
[5] R. E. Burkard, K. Dlaska, and B. Klinz, *The quickest flow problem*, ZOR Methods and Models Oper. Res., 37 (1993), pp. 31–58.
[6] L. Fleischer, Universally maximum flow with piecewise constant capacities, in Integer Programming and Combinatorial Optimization: 7th International IPCO Conference, G. Cornuejols, R. E. Burkard, and G. J. Woeginger, eds., Lecture Notes in Comput. Sci. 16105, Springer-Verlag, Berlin, 1999, pp. 151–165.
[7] L. Fleischer and J. B. Orlin, *Optimal rounding of instantaneous fractional flows over time*, SIAM J. Discrete Math, 13 (2000), pp. 145–153.
[8] L. Fleischer and É. Tardos, *Efficient continuous-time dynamic network flow algorithms*, Oper. Res. Lett., 23 (1998), pp. 71–80.
[9] L. R. Ford and D. R. Fulkerson, *Flows in Networks*, Princeton University Press, Princeton, NJ, 1962.
[10] G. Gallo, M. D. Grigoriadis, and R. E. Tarjan, *A fast parametric maximum flow algorithm and applications*, SIAM J. Comput., 18 (1989), pp. 30–55.
[11] A. V. Goldberg and S. Rao, *Beyond the flow decomposition barrier*, J. ACM, 45 (1998), pp. 783–797.
[12] A. V. Goldberg and R. E. Tarjan, *A new approach to the maximum flow problem*, J. ACM, 35 (1988), pp. 921–940.
[13] A. V. Goldberg and R. E. Tarjan, *Finding minimum-cost circulations by canceling negative cycles*, J. ACM, 36 (1989), pp. 388–397.
[14] B. Hajek and R. G. Ogier, *Optimal dynamic routing in communication networks with continuous traffic,* Networks, 14 (1984), pp. 457–487.
[15] B. Hoppe and É. Tardos, *The quickest transshipment problem*, Math. Oper. Res., 25 (2000), pp. 36–62. Extended abstract appeared in Proceedings of SODA 1995.
[16] B. Klinz and G. J. Woeginger, *Minimum cost dynamic flows: The series-parallel case*, in Integer Programming and Combinatorial Optimization, Proceedings of the 4th International IPCO Conference, E. Balas and J. Clausen, eds., Lecture Notes in Comput. Sci. 920, Springer-Verlag, Berlin, 1995, pp. 329–343.
[17] N. Megiddo, *Optimal flows in networks with multiple sources and sinks*, Math. Programming, 7 (1974), pp. 97–107.
[18] E. Minieka, *Maximal, lexicographic, and dynamic network flows*, Oper. Res., 21 (1973), pp. 517–527.
[19] F. H. Moss and A. Segall, *An optimal control approach to dynamic routing in networks*, IEEE Trans. Automat. Control, 27 (1982), pp. 329–339.
[20] R. G. Ogier, *Minimum-delay routing in continuous-time dynamic networks with piecewise-constant capacities*, Networks, 18 (1988), pp. 303–318.
[21] J. B. Orlin, *Maximum-throughput dynamic network flows*, Math. Programming, 27 (1983), pp. 214–231.
[22] J. B. Orlin, *Minimum convex cost dynamic network flows*, Math. Oper. Res., 9 (1984), pp. 190–207.
[23] A. B. Philpott, *Continuous-time flows in networks*, Math. Oper. Res., 15 (1990), pp. 640–661.
[24] W. B. Powell, P. Jaillet, and A. Odoni, *Stochastic and dynamic networks and routing*, in Network Routing, Handbooks Oper. Res. Management Sci. 8, M. O. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemhauser, eds., North-Holland, Amsterdam, 1995, pp. 141–295.
[25] M. C. Pullan, *An algorithm for a class of continuous linear programs*, SIAM J. Control Optim., 31 (1993), pp. 1558–1577.
[26] M. C. Pullan, *A study of general dynamic network programs with arc time-delays*, SIAM J. Optim., 7 (1997), pp. 889–912.
[27] T. Radzik, *Parametric flows, weighted means of cuts, and fractional combinatorial optimization*, in Complexity in Numerical Optimization, P. M. Pardalos, ed., World Scientific, River Edge, NJ, 1993, pp. 351–386.
[28] G. I. Stassinopoulos and P. Konstantopoulos, *Optimal congestion control in single destination networks*, IEEE Trans. Comm., 33 (1985), pp. 792–800.
[29] W. L. Wilkinson, *An algorithm for universal maximal dynamic flows in a network*, Oper. Res., 19 (1971), pp. 1602–1612.

SIAM J. OPTIM.
Vol. 12, No. 1, pp. 36–78

EFFECTS OF FINITE-PRECISION ARITHMETIC ON INTERIOR-POINT METHODS FOR NONLINEAR PROGRAMMING*

STEPHEN J. WRIGHT†

Abstract. We show that the effects of finite-precision arithmetic in forming and solving the linear system that arises at each iteration of primal-dual interior-point algorithms for nonlinear programming are benign, provided that the iterates satisfy centrality and feasibility conditions of the type usually associated with path-following methods. When we replace the standard assumption that the active constraint gradients are independent by the weaker Mangasarian–Fromovitz constraint qualification, rapid convergence usually is attainable, even when cancellation and roundoff errors occur during the calculations. In deriving our main results, we prove a key technical result about the size of the exact primal-dual step. This result can be used to modify existing analysis of primal-dual interior-point methods for convex programming, making it possible to extend the superlinear local convergence results to the nonconvex case.

Key words. primal-dual interior-point algorithms, finite-precision arithmetic, nonlinear programming, constraint qualification

AMS subject classifications. 90C33, 90C30, 49M45

PII. S1052623498347438

1. Introduction.

We investigate the effects of finite-precision arithmetic on the calculated steps of primal-dual interior-point (PDIP) algorithms for the nonlinear programming problem

$$\text{NLP:} \qquad \min_z \ \phi(z) \qquad \text{subject to } g(z) \le 0, \tag{1.1}$$

where $\phi : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}^m$ are twice Lipschitz continuously differentiable functions. Optimality conditions for this problem can be derived from the Lagrangian function $\mathcal{L}(z, \lambda)$, which is defined as

$$\mathcal{L}(z, \lambda) = \phi(z) + \sum_{i=1}^{m} \lambda_i g_i(z) = \phi(z) + \lambda^T g(z), \tag{1.2}$$

where $\lambda \in \mathbb{R}^m$ is a vector of Lagrange multipliers. When a constraint qualification (discussed below) holds at the point z^*, first-order necessary conditions for z^* to be a solution of (1.1) are that there exists a vector of Lagrange multipliers $\lambda^* \in \mathbb{R}^m$ such that the following conditions are satisfied for $(z, \lambda) = (z^*, \lambda^*)$:

$$\mathcal{L}_z(z, \lambda) = \nabla\phi(z) + \nabla g(z)\lambda = 0, \quad g(z) \le 0, \quad \lambda \ge 0, \quad \lambda^T g(z) = 0, \tag{1.3}$$

*Received by the editors November 9, 1998; accepted for publication (in revised form) January 23, 2001; published electronically May 22, 2001.

http://www.siam.org/journals/siopt/12-1/34743.html

†Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439 (wright@mcs.anl.gov), and Department of Computer Science, University of Chicago, Chicago, IL 60637.

where

$$\nabla g(z) = [\nabla g_1(z), \nabla g_2(z), \ldots, \nabla g_m(z)] .$$

The conditions (1.3) are the well-known Karush–Kuhn–Tucker (KKT) conditions. We use $\mathcal{S}_\lambda$ to denote the set of vectors λ^* such that (z^*, λ^*) satisfies (1.3). The primal-dual solution set is defined by

$$\mathcal{S} = \{z^*\} \times \mathcal{S}_\lambda. \tag{1.4}$$

This paper discusses local convergence of PDIP algorithms for (1.1), assuming that the algorithm is implemented on a computer that performs calculations according to the standard model of floating-point arithmetic. Because of our focus on *local* convergence properties, we assume throughout that the current iterate (z, λ) is close enough to the solution set $\mathcal{S}$ that superlinear convergence would occur if exact steps (uncorrupted by finite precision) were taken. In the interests of generality, we weaken an assumption that is often made in the analysis of algorithms for (1.1), namely, that the gradients of the active constraints are linearly independent at the solution. We replace this linear independence constraint qualification (LICQ) with the weaker Mangasarian–Fromovitz constraint qualification (MFCQ) [18]. MFCQ allows constraint gradients to become dependent at the solution, so that the set $\mathcal{S}_\lambda$ of optimal Lagrange multipliers is no longer necessarily a singleton, though it remains bounded. We continue to assume that a strict complementarity (SC) condition holds, that is,

$$g_i(z^*) = 0 \;\Rightarrow\; \lambda_i^* > 0 \quad \text{for some } \lambda^* \in \mathcal{S}_\lambda. \tag{1.5}$$

In the context of rapidly convergent algorithms, the SC condition makes good sense. If SC fails to hold, superlinear convergence of Newton-like algorithms does not occur, except for specially modified algorithms such as those that identify the active constraints explicitly (see Monteiro and Wright [20] and El-Bakry, Tapia, and Zhang [8]).

The major conclusion of the paper is that the effects of roundoff errors on the rapid local convergence of the algorithm are fairly benign. When a standard second-order condition is added to the assumptions already mentioned, the steps produced under floating-point arithmetic approach $\mathcal{S}$ almost as effectively as do exact steps, as long as the distance to the solution set remains significantly greater than the unit roundoff $\mathbf{u}$. The latter condition is hardly restrictive, since the data errors made in storing the problem in a digital computer mean that the solution set is known only to within some multiple of $\mathbf{u}$ in any case.

The conclusions about the effectiveness of the computed steps are not obvious, because all three formulations of the linear system that must be solved to compute the step at each iteration may become highly ill conditioned near the solution. Our analysis would be significantly simpler if we were to make the LICQ assumption because, in this case, one formulation of the linear equations remains well conditioned, and stability of the three standard formulations can be proved by exploiting their relationship to this system of equations.

This work is related to earlier work of the author on finite-precision analysis of interior-point algorithms for linear complementarity problems [24] and linear programming [27, 30]. The existence of second-order effects gives the analysis here a somewhat different flavor, however. In addition, we go into more depth in checking that the computed iterates can continue to satisfy the approximate centrality conditions usually required in primal-dual algorithms, and in deriving expressions for

the rate at which the computed iterates approach the solution set. Related work by Forsgren, Gill, and Shinnerl [9] deals with one formulation of the step equations for the nonlinear programming problem—the so-called augmented form treated here in section 6—but makes assumptions on the pivot sequence that do not always hold in practice. M. H. Wright [23] recently presented an analysis of the condensed form of the step equations discussed in section 5 under the assumption that LICQ holds, and found that the computed steps were more accurate than would be expected from a naive analysis.

For linear programming, the PDIP approach has emerged as the most powerful of the interior-point approaches. The supporting theory is strong, in terms of global and local convergence analysis and complexity theory (see the bibliography of Wright [26]). Implementations yield better results than pure-primal or barrier-function approaches; see Andersen et al. [1]. Strong theory is also available for these algorithms when applied to convex programming, in which $\phi(\cdot)$ and $g_i(\cdot)$, $i = 1, \ldots, m$, are all convex functions; see, for example, Wright and Ralph [31] and Ralph and Wright [21, 22]. The latter paper drops the LICQ assumption in favor of MFCQ, making the local theory stronger in one sense than the corresponding local theory for the sequential quadratic programming (SQP) algorithm. The use of MFCQ complicates the analysis considerably, however; under LICQ, the implicit function theorem can be used to prove a key technical result about the length of the step, while more complicated logic is needed to derive this same result under MFCQ.

A significant by-product of the current paper is to prove the key technical result about the length of the rapidly convergent step (Corollary 4.3) under MFCQ and SC, even when the problem (1.1) is not convex. This allows the local convergence results of Ralph and Wright [31, 21, 22] to be extended to general nonconvex nonlinear problems.

The analysis of this paper could also be applied to the recently proposed stabilized sequential quadratic programming (sSQP) algorithm (see Wright [29] and Hager [15]), in which small penalties on the change in the multiplier estimate λ from one iteration to the next ensure rapid convergence even when LICQ is relaxed to MFCQ. A finite-precision analysis of the sSQP method appears in [29, section 3.2], but only for the augmented form of the step equations. Analysis quite similar to that of the current paper could be applied to show that similar conclusions continue to hold when a condensed form of the step equations is used instead. We omit the details.

The remainder of this paper is structured in the following way. Section 2 contains notation, together with our basic assumptions about (1.1) and some relevant results from the literature. Section 3 discusses the primal-dual interior-point framework, defining the general form of each iteration and the step equations that must be solved at each iteration. Subsection 3.2 proves an important technical result about the relationship between the distance of an interior-point iterate to the solution set $\mathcal{S}$ and a duality measure μ. Section 4 describes perturbed variants of the linear systems that are solved to obtain PDIP steps, and proves our key results about the effect of the perturbations on the accuracy of the steps.

Section 5 focuses on one form of the PDIP step equations: the most compact form in which most of the computational effort goes into factoring a symmetric positive definite matrix, usually by a Cholesky procedure. We trace the effect on step accuracy of errors in evaluation of the functions, formation of the system, and the factorization/solution process. Further, we show the effects of these inaccuracies on the distance that we can move along the steps before the interiority condition is vi-

olated, and on various measures of algorithmic progress. An analogous treatment of the augmented form of the step equations appears in section 6. The conclusions of this section depend on the actual algorithm used to solve the augmented system—it is not sufficient to assume, as in section 5, that any backward-stable procedure is used to factor the matrix. (We note that similar results hold for the full form of the step equations, but we omit the details of this case, which can be found in the technical report [28].) We conclude with a numerical illustration of the main results in section 7 and summarize the paper in section 8.

2. Notation, assumptions, and basic results. We use $\mathcal{B}$ to denote the set of active indices at z^*, that is,

$$\mathcal{B} = \{i = 1, 2, \ldots, m \,|\, g_i(z^*) = 0\}, \tag{2.1}$$

whereas $\mathcal{N}$ denotes its complement

$$\mathcal{N} = \{1, 2, \ldots, m\} \backslash \mathcal{B}. \tag{2.2}$$

The set $\mathcal{B}_+ \subset \mathcal{B}$ is defined as

$$\mathcal{B}_+ = \{i \in \mathcal{B} \,|\, \lambda_i^* > 0 \text{ for some } \lambda^* \text{ satisfying (1.3)}\}. \tag{2.3}$$

The strict complementarity condition (1.5) is equivalent to

$$\mathcal{B}_+ = \mathcal{B}. \tag{2.4}$$

We frequently make reference to submatrices and subvectors corresponding to the index sets $\mathcal{B}$ and $\mathcal{N}$. For example, the quantities $\lambda_{\mathcal{B}}$ and $g_{\mathcal{B}}(z)$ are the vectors containing the components λ_i and $g_i(z)$, respectively, for $i \in \mathcal{B}$, while $\nabla g_{\mathcal{B}}(z)$ is the matrix whose columns are $\nabla g_i(z)$, $i \in \mathcal{B}$.

The Mangasarian–Fromovitz constraint qualification (MFCQ) is satisfied at z^* if there is a vector $\bar{y} \in \mathbb{R}^n$ such that

$$\nabla g_{\mathcal{B}}(z^*)^T \bar{y} < 0. \tag{2.5}$$

The following fundamental result about MFCQ is due to Gauvin [11].

LEMMA 2.1. *Suppose that the first-order conditions* (1.3) *are satisfied at* $z = z^*$. *Then* $\mathcal{S}_\lambda$ *is bounded if and only if the MFCQ condition* (2.5) *is satisfied at* z^*.

This result is crucial because it allows our (local) analysis to place a uniform bound on all λ in a neighborhood of the dual solution set $\mathcal{S}_\lambda$.

The second-order condition used in most of the remainder of the paper is that there is a constant $\xi > 0$ such that

$$w^T \mathcal{L}_{zz}(z^*, \lambda^*) w \geq \xi \|w\|^2 \tag{2.6}$$

for all $\lambda^* \in \mathcal{S}_\lambda$ and all w satisfying

$$\begin{aligned} &\nabla g_i(z^*)^T w = 0 \quad \text{for all } i \in \mathcal{B}_+, \\ &\nabla g_i(z^*)^T w \leq 0 \quad \text{for all } i \in \mathcal{B} \backslash \mathcal{B}_+. \end{aligned} \tag{2.7}$$

When the SC condition (1.5) (alternatively, (2.4)) is satisfied, this direction set is simply null $\nabla g_{\mathcal{B}}(z^*)^T$.

A simple example that satisfies MFCQ but not LICQ at the solution and that satisfies the second-order conditions (2.6), (2.7) and the SC condition is as follows:

$$\min_{z\in\mathbb{R}^2} z_1 \text{ subject to } (z_1-1/3)^2+z_2^2\le 1/9,\ \ (z_1-2/3)^2+z_2^2\le 4/9. \tag{2.8}$$

The solution is $z^*=0$, and the optimal multiplier set is

$$\mathcal{S}_\lambda=\{\lambda\ge 0\,|\,2\lambda_1+4\lambda_2=3\}. \tag{2.9}$$

The gradients of the two constraints at the solution are $(-2/3,0)^T$ and $(-4/3,0)^T$, respectively. They are linearly dependent, but the MFCQ condition (2.5) can be satisfied by choosing $\bar{y}=(1,0)^T$.

We use $\mathbf{u}$ to denote the unit roundoff, which we define as the smallest number such that the following property holds: When x and y are any two floating-point numbers, op denotes $+$, $-$, $\times$, or $/$, and $fl(z)$ denotes the floating-point approximation of a real number z, we have

$$fl(x \text{ op } y)=(x \text{ op } y)(1+\epsilon),\quad |\epsilon|\le \mathbf{u}. \tag{2.10}$$

Modest multiples of $\mathbf{u}$ are denoted by $\delta_{\mathbf{u}}$.

We assume that the problem is scaled so that the values of g and ϕ and their first and second derivatives in the vicinity of the solution set $\mathcal{S}$, and the values (z,λ) themselves, can all be bounded by moderate quantities. When multiplied by $\mathbf{u}$, quantities of this type are absorbed into the notation $\delta_{\mathbf{u}}$ in the analysis below.

Order notation $O(\cdot)$ and $\Theta(\cdot)$ is used as follows: If v (vector or scalar) and ϵ (nonnegative scalar) are two quantities that share a dependence on other variables, we write $v=O(\epsilon)$ if there is a moderate constant β_1 such that $\|v\|\le\beta_1\epsilon$ for all values of ϵ that are interesting in the given context. (The "interesting context" frequently includes cases in which ϵ is either sufficiently small or sufficiently large, but we often use $v=O(\mu)$ to indicate that $\|v\|\le\beta_1\mu$ for all sufficiently small μ that satisfy $\mu\gg\mathbf{u}$ for some β_1; see later discussion.) We write $v=\Theta(\epsilon)$ if there are constants β_1 and β_0 such that $\beta_0\epsilon\le\|v\|\le\beta_1\epsilon$ for all interesting values of ϵ. Similarly, we write $v=O(1)$ if $\|v\|\le\beta_1$, and $v=\Theta(1)$ if $\beta_0\le\|v\|\le\beta_1$.

We use the notation $\delta(z,\lambda)$ to denote the distance from (z,λ) to the primal-dual solution set, that is,

$$\delta(z,\lambda)\stackrel{\text{def}}{=}\min_{(z^*,\lambda^*)\in\mathcal{S}}\|(z,\lambda)-(z^*,\lambda^*)\|. \tag{2.11}$$

It is well known (see, for example, Theorem A.1 of Wright [25]) that this distance can be estimated in terms of known quantities at (z,λ). We state this result formally as follows.

THEOREM 2.2. *Suppose that the first-order conditions* (1.3)*, the MFCQ condition* (2.5)*, and the second-order conditions* (2.6), (2.7) *are satisfied at the solution z^*. Then if $\lambda\ge 0$, we have*

$$\delta(z,\lambda)=\Theta\left(\left\|\begin{bmatrix}\mathcal{L}_z(z,\lambda)\\ \min(\lambda,-g(z))\end{bmatrix}\right\|\right). \tag{2.12}$$

We write the singular value decomposition (SVD) of the matrix $\nabla g_{\mathcal{B}}(z^*)$ of first partial derivatives as follows:

$$\nabla g_{\mathcal{B}}(z^*)=\begin{bmatrix}\hat{U} & \hat{V}\end{bmatrix}\begin{bmatrix}\Sigma & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}U^T\\ V^T\end{bmatrix}=\hat{U}\Sigma U^T, \tag{2.13}$$

where the matrices $\begin{bmatrix} \hat{U} & \hat{V} \end{bmatrix}$ and $\begin{bmatrix} U & V \end{bmatrix}$ are orthogonal and Σ is a diagonal matrix with positive diagonal elements.

Note that the columns of $\hat{U}$ form a basis for the range space of $\nabla g_{\mathcal{B}}(z^*)$, while the columns of $\hat{V}$ form a basis for the null space of $\nabla g_{\mathcal{B}}(z^*)^T$. Similarly, the columns of U form a basis for the range space of $\nabla g_{\mathcal{B}}(z^*)^T$, while the columns of V form a basis for the null space of $\nabla g_{\mathcal{B}}(z^*)$. These four subspaces are key to our analysis, particularly the subspace spanned by the columns of V. For the computational methods used to solve the primal-dual step equations discussed in this paper, the computed step in the $\mathcal{B}$-components of the multipliers—that is, $\Delta\lambda_{\mathcal{B}}$—has a larger error in the range space of V than in the complementary subspace spanned by the columns of U. The errors in the computed primal step Δz, the computed step of the $\mathcal{N}$-components of the multipliers $\lambda_{\mathcal{N}}$, and the computed step in the dual slack variables (defined later) are typically also less significant than the error in $V^T\Delta\lambda_{\mathcal{B}}$. We show, however, that the potentially large error in $V^T\Delta\lambda_{\mathcal{B}}$ does not affect the performance of primal-dual algorithms that use these computed steps until μ becomes similar to $\mathbf{u}^{1/2}$.

When the stronger LICQ condition holds, the matrix V is vacuous, and the SVD (2.13) reduces to $\nabla g_{\mathcal{B}}(z^*) = \hat{U}\Sigma U^T$. Much of the analysis in this paper would simplify considerably under LICQ, in part because $V^T\Delta\lambda_{\mathcal{B}}$—the step component with the large error—is no longer present.

We use $\sigma_{\min}(\cdot)$ to denote the smallest eigenvalue, and $\mathrm{cond}(\cdot)$ to denote the condition number, as measured by the Euclidean norm.

3. Primal-dual interior-point methods.

3.1. Centrality conditions and step equations. Primal-dual interior-point methods are constrained, modified Newton methods applied to a particular form of the KKT conditions (1.3). By introducing a vector $s \in \mathbb{R}^m$ of slacks for the inequality constraint, we can rewrite the nonlinear program as

$$\min_{(z,s)} \phi(z) \quad \text{subject to } g(z) + s = 0,\ s \geq 0,$$

and the KKT conditions (1.3) as

$$\mathcal{L}_z(z,\lambda) = 0, \quad g(z) + s = 0, \quad (\lambda, s) \geq 0, \quad \lambda^T s = 0. \tag{3.1}$$

Motivated by this form of the conditions, we define the mapping $\mathcal{F}(z,\lambda,s)$ by

$$\mathcal{F}(z,\lambda,s) \stackrel{\text{def}}{=} \begin{bmatrix} \nabla\phi(z) + \nabla g(z)\lambda \\ g(z) + s \\ S\Lambda e \end{bmatrix}, \tag{3.2}$$

where the diagonal matrices S and Λ are defined by

$$S \stackrel{\text{def}}{=} \mathrm{diag}(s_1, s_2, \ldots, s_m), \quad \Lambda \stackrel{\text{def}}{=} \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_m)$$

and e is defined as

$$e = (1, 1, \ldots, 1)^T. \tag{3.3}$$

The KKT conditions (3.1) can now be stated equivalently as

$$\mathcal{F}(z,\lambda,s) = 0, \qquad (s,\lambda) \geq 0. \tag{3.4}$$

Primal-dual iterates (z, λ, s) invariably satisfy the strict bound $(s, \lambda) > 0$, while they approach satisfaction of the condition $\mathcal{F}(\cdot) = 0$ in the limit. An important measure of progress is the *duality measure* $\mu(\lambda, s)$, which is defined by

$$\mu(\lambda, s) \stackrel{\text{def}}{=} \lambda^T s/m. \tag{3.5}$$

When μ is used without arguments, we assume that $\mu = \mu(\lambda, s)$, where (z, λ, s) is the current primal-dual iterate. We emphasize that μ is a function of (z, λ, s), rather than a target value explicitly chosen by the algorithm, as is the case in some of the literature.

A typical step $(\Delta z, \Delta\lambda, \Delta s)$ of the primal-dual method satisfies

$$\nabla\mathcal{F}(z, \lambda, s) \begin{bmatrix} \Delta z \\ \Delta\lambda \\ \Delta s \end{bmatrix} = -\mathcal{F}(z, \lambda, s) - \begin{bmatrix} 0 \\ 0 \\ t \end{bmatrix}, \tag{3.6}$$

where t defines the deviation from a pure Newton step for $\mathcal{F}$ (which is also known as a "primal-dual affine-scaling" step). The vector t frequently contains a centering term $\sigma\mu e$, where σ is a centering parameter in the range $[0, 1]$. It sometimes also contains higher-order information, such as the product $\Delta\Lambda_{\text{aff}}\Delta S_{\text{aff}}e$, where $\Delta\Lambda_{\text{aff}}$ and ΔS_{aff} are the diagonal matrices constructed from the components of the pure Newton step (Mehrotra [19]). In any case, the vector t usually goes to zero rapidly as the iterates converge to a solution, so that the steps generated from (3.6) approach pure Newton steps, which in turn ensures rapid local convergence. Throughout this paper, we assume that t satisfies the estimate

$$t = O(\mu^2). \tag{3.7}$$

All our major results continue to hold, with slight modification, if we replace (3.7) by $t = O(\mu^\sigma)$ for some $\sigma \in (1, 2]$. Our essential point remains unchanged; the theoretical superlinear convergence rate promised by this choice of t is not seriously compromised by roundoff errors as long as μ remains significantly larger than the unit roundoff $\mathbf{u}$. To avoid notational clutter, however, we analyze only the case (3.7).

Using the definition (1.2), we can write the system (3.6) explicitly as follows:

$$\begin{bmatrix} \mathcal{L}_{zz}(z, \lambda) & \nabla g(z) & 0 \\ \nabla g(z)^T & 0 & I \\ 0 & S & \Lambda \end{bmatrix} \begin{bmatrix} \Delta z \\ \Delta\lambda \\ \Delta s \end{bmatrix} = - \begin{bmatrix} \mathcal{L}_z(z, \lambda) \\ g(z) + s \\ S\Lambda e + t \end{bmatrix}. \tag{3.8}$$

Block eliminations can be performed on this system to yield more compact formulations. By eliminating Δs, we obtain the *augmented system* form, which is

$$\begin{bmatrix} \mathcal{L}_{zz}(z, \lambda) & \nabla g(z) \\ \nabla g(z)^T & -\Lambda^{-1}S \end{bmatrix} \begin{bmatrix} \Delta z \\ \Delta\lambda \end{bmatrix} = \begin{bmatrix} -\mathcal{L}_z(z, \lambda) \\ -g(z) + \Lambda^{-1}t \end{bmatrix}. \tag{3.9}$$

By eliminating $\Delta\lambda$ from this system, we obtain a system that is sometimes referred to as the *condensed* form (or in the case of linear programming as the *normal equations* form), which is

$$\begin{aligned} &\left[\mathcal{L}_{zz}(z, \lambda) + \nabla g(z)\Lambda S^{-1}\nabla g(z)^T\right]\Delta z \\ &= -\mathcal{L}_z(z, \lambda) - \nabla g(z)\Lambda S^{-1}[g(z) - \Lambda^{-1}t]. \end{aligned} \tag{3.10}$$

We consider primal-dual methods in which each iterate (z, λ, s) satisfies the following properties:

$$\|r_f(z,\lambda)\| \le C\mu, \quad \text{where } r_f(z,\lambda) \stackrel{\text{def}}{=} \mathcal{L}_z(z,\lambda), \tag{3.11a}$$

$$\|r_g(z,s)\| \le C\mu, \quad \text{where } r_g(z,s) \stackrel{\text{def}}{=} g(z) + s, \tag{3.11b}$$

$$(\lambda, s) > 0, \quad \lambda_i s_i \ge \gamma\mu \quad \text{for all } i = 1, 2, \ldots, m \tag{3.11c}$$

for some constants $C > 0$ and $\gamma \in (0, 1)$, where μ is defined as in (3.5). (In much of the succeeding discussion, we omit the arguments from the quantities μ, r_f, and r_g when they are evaluated at the current iterate (z, λ, s).) These conditions ensure that the pairwise products $s_i\lambda_i$, $i = 1, 2, \ldots, m$ are not too disparate and that the first two components of $\mathcal{F}$ in (3.2) can be bounded in terms of the third component. They are sometimes called the *centrality conditions* because they are motivated by the notion of a central path and its neighborhoods. Conditions of the type (3.11) are imposed in most path-following interior-point methods for linear programming (see, for example, [26]). For nonlinear convex programming, examples of methods that require these conditions can be found in Ralph and Wright [31, 21, 22]. In nonlinear programming, we mention Gould et al. [14] (see Algorithm 4.1 and Figure 5.1) and Byrd, Liu, and Nocedal [4]. In the latter paper, (3.11a) and (3.11b) are imposed explicitly, while (3.11c) can be guaranteed by choosing $\epsilon_\mu = (1-\gamma)\mu$. Even when the choice $\epsilon_\mu = \mu$ is made, as in the bulk of the discussion in [4], their other conditions concerning positivity of (s, λ) can be expected to produce iterates that satisfy (3.11c) in practice.

For points (z, λ, s) that satisfy (3.11), we can use μ to estimate the distance $\delta(z, \lambda)$ from (z, λ) to the solution set $\mathcal{S}$ (see (2.11)). These results, which are proved in the following subsection, can be summarized briefly as follows. When the MFCQ condition (2.5) and the second-order conditions (2.6), (2.7) are satisfied, we have that $\delta(z, \lambda) = O(\mu^{1/2})$. When the strict complementarity assumption (1.5) is added, we obtain the stronger estimate $\delta(z, \lambda) = O(\mu)$. We can use these estimates to obtain bounds on the elements of the diagonal matrices S, Λ, $S^{-1}\Lambda$, and $\Lambda^{-1}S$ in the systems above; these bounds are the key to the error analysis of the remainder of the paper.

3.2. Using the duality measure to estimate distance to the solution. The main result of this section, Theorem 3.3, shows that under certain assumptions, the distance $\delta(z, \lambda)$ of a primal-dual iterate (z, λ, s) to the solution set $\mathcal{S}$ can be estimated by the duality measure μ. We start with a technical lemma that proves the weaker estimate $\delta(z, \lambda) = O(\mu^{1/2})$. Note that this result does not assume that the SC condition (1.5) holds.

LEMMA 3.1. *Suppose that z^* is a solution of* (1.1) *at which the MFCQ condition* (2.5) *and the second-order conditions* (2.6), (2.7) *are satisfied. Then for all (z, λ) with $\lambda \ge 0$ for which there is a vector s such that (z, λ, s) satisfies* (3.11), *we have that*

$$\delta(z, \lambda) = O(\mu^{1/2}). \tag{3.12}$$

Proof. We prove the result by showing that

$$\begin{bmatrix} \mathcal{L}_z(z,\lambda) \\ \min(\lambda, -g(z)) \end{bmatrix} = O(\mu^{1/2}) \tag{3.13}$$

and then applying Theorem 2.2. Since $\mathcal{L}_z(z,\lambda) = r_f = O(\mu)$, the first part of the vector satisfies the required estimate. For the second part, we have from (3.11b) that

$$-g(z) = s - r_g = s + O(\mu)$$

and hence that

$$\min(-g_i(z), \lambda_i) = \min(s_i, \lambda_i) + O(\mu). \tag{3.14}$$

Because of (3.5) and (3.11c), we have that $s_i\lambda_i \le m\mu$ and $(\lambda_i, s_i) > 0$. It follows immediately that $\min(\lambda_i, s_i) \le (m\mu)^{1/2}$ for $i = 1, 2, \ldots, m$. Hence, by substitution into (3.14), we obtain

$$\min(-g_i(z), \lambda_i) \le (m\mu)^{1/2} + O(\mu) = O(\mu^{1/2}).$$

We conclude that the second part of the vector in (3.13) is of size $O(\mu^{1/2})$, so the proof is complete. □

The following examples show the upper bound of Lemma 3.1 is indeed achieved and that it is not possible to obtain a lower bound on $\delta(z,\lambda)$ as a strictly increasing nonnegative function of μ. To demonstrate the first claim, consider the problem

$$\min \tfrac{1}{2}z^2 \qquad \text{subject to } -z \le 0.$$

The point $(z,\lambda,s) = (\epsilon,\epsilon,\epsilon)$ satisfies

$$\mathcal{L}_z(z,\lambda) = 0, \quad g(z) + s = 0, \quad s\lambda = \epsilon^2, \quad \mu = \epsilon^2,$$

so that the conditions (3.11) are satisfied. Clearly the distance from the point (z,λ) to the solution set $\mathcal{S} = (0,0)$ is $\sqrt{2}\epsilon = \sqrt{2}\mu^{1/2}$. For the second claim, consider any nonlinear program such that $\mathcal{B} = \{1, 2, \ldots, m\}$ (that is, all constraints active) and strict complementarity (1.5) holds at some multiplier λ^*. Then for appropriate choices of γ and C, the point

$$(z,\lambda,s) = (z^*, \lambda^*, (m\mu)/(e^T\lambda^*)e) \tag{3.15}$$

satisfies the definition (3.5) and the condition (3.11) for any $\mu > 0$. On the other hand, we have $\delta(z,\lambda) = \delta(z^*,\lambda^*) = 0$ by definition, so there are no $\beta > 0$ and $\sigma > 0$ that yield a lower bound estimate of the form $\delta(z,\lambda) \ge \beta\mu^\sigma$.

We now prove an extension of Lemma 5.1 of Ralph and Wright [21], dropping the monotonicity assumption of this earlier result.

LEMMA 3.2. *Suppose that the conditions of Lemma* 3.1 *hold and in addition that the SC condition* (1.5) *is satisfied. Then for all* (z,λ,s) *satisfying* (3.11)*, we have that*

$$i \in \mathcal{B} \Rightarrow s_i = \Theta(\mu), \quad \lambda_i = \Theta(1), \tag{3.16a}$$

$$i \in \mathcal{N} \Rightarrow s_i = \Theta(1), \quad \lambda_i = \Theta(\mu). \tag{3.16b}$$

Proof. By boundedness of $\mathcal{S}$ (Lemma 2.1), we have for all (z,λ,s) sufficiently close to $\mathcal{S}$ that

$$\lambda_i = O(1), \qquad s_i = -g_i(z) + (r_g)_i = O(1). \tag{3.17}$$

Given (z, λ, s) satisfying (3.11), let $P(\lambda)$ be the projection of λ onto the set $\mathcal{S}_\lambda$, and let $\lambda^* \in \mathcal{S}_\lambda$ be some strictly complementary optimal multiplier (for which (1.5) is satisfied). From Lemma 3.1 we obtain

$$\|z - z^*\| = O(\mu^{1/2}). \tag{3.18}$$

Using this observation together with smoothness of $\phi(\cdot)$ and $g(\cdot)$, we have for the gradient of $\mathcal{L}$ that

$$\begin{aligned}
&\mathcal{L}_z(z,\lambda) - \mathcal{L}_z(z^*,\lambda^*) \\
&= \nabla\phi(z) - \nabla\phi(z^*) + \nabla g(z)\lambda - \nabla g(z^*)\lambda^* \\
&= O(\mu^{1/2}) + \nabla g(z)[\lambda - P(\lambda)] + [\nabla g(z) - \nabla g(z^*)]P(\lambda) + \nabla g(z^*)[P(\lambda) - \lambda^*].
\end{aligned}$$

Since $P(\lambda)$ and λ^* are both in $\mathcal{S}_\lambda$, we find from (1.3) that the last term vanishes. From (3.18) and $P(\lambda) = O(1)$, the second-to-last term has size $O(\mu^{1/2})$. For the remaining term, we have $\nabla g(z) = O(1)$, and $\|\lambda - P(\lambda)\| \le \delta(z,\lambda) = O(\mu^{1/2})$. By assembling all these observations and using $\mathcal{L}_z(z^*,\lambda^*) = 0$, we obtain

$$\mathcal{L}_z(z,\lambda) = \mathcal{L}_z(z,\lambda) - \mathcal{L}_z(z^*,\lambda^*) = O(\mu^{1/2}). \tag{3.19}$$

Using again that $\nabla g(z^*)[P(\lambda) - \lambda^*] = 0$, we have from (3.18) that

$$\begin{aligned}
[P(\lambda) - \lambda^*]^T[g(z) - g(z^*)] &= [P(\lambda) - \lambda^*]^T[\nabla g(z^*)^T(z - z^*) + O(\|z - z^*\|^2)] \\
&= O(\|z - z^*\|^2) = O(\mu).
\end{aligned} \tag{3.20}$$

By gathering the estimates (3.12), (3.18), (3.19), and (3.20), we obtain

$$\begin{aligned}
&\begin{bmatrix} z - z^* \\ \lambda - \lambda^* \end{bmatrix}^T \begin{bmatrix} \mathcal{L}_z(z,\lambda) - \mathcal{L}_z(z^*,\lambda^*) \\ -g(z) + g(z^*) \end{bmatrix} \\
&= \begin{bmatrix} z - z^* \\ \lambda - P(\lambda) \end{bmatrix}^T \begin{bmatrix} \mathcal{L}_z(z,\lambda) - \mathcal{L}_z(z^*,\lambda^*) \\ -g(z) + g(z^*) \end{bmatrix} \\
&\quad + [P(\lambda) - \lambda^*]^T[-g(z) + g(z^*)] \\
&= O(\delta(z,\lambda))O(\mu^{1/2}) + O(\mu) = O(\mu).
\end{aligned} \tag{3.21}$$

By substituting from (3.11) and using (3.21), we obtain

$$\begin{bmatrix} z - z^* \\ \lambda - \lambda^* \end{bmatrix}^T \begin{bmatrix} r_f \\ s - r_g - s^* \end{bmatrix} = \begin{bmatrix} z - z^* \\ \lambda - \lambda^* \end{bmatrix}^T \begin{bmatrix} \mathcal{L}_z(z,\lambda) - \mathcal{L}_z(z^*,\lambda^*) \\ -g(z) + g(z^*) \end{bmatrix} = O(\mu),$$

and therefore

$$(\lambda - \lambda^*)^T(s - s^*) = -(z - z^*)^T r_f + (\lambda - \lambda^*)^T r_g + O(\mu).$$

By using the conditions (3.11a), (3.11b), and the definition (3.5), we obtain

$$\begin{aligned}
&-\sum_{i=1}^m \lambda_i^* s_i - \sum_{i=1}^m \lambda_i s_i^* \\
&= -(\lambda^*)^T s - \lambda^T s^* = -\lambda^T s + O(\mu) + O(\|z - z^*\|\|r_f\|) + O(\|\lambda - \lambda^*\|\|r_g\|) = O(\mu).
\end{aligned}$$

Since $(\lambda, s) > 0$ and $(\lambda^*, s^*) \ge 0$, all terms $\lambda_i^* s_i$ and $\lambda_i s_i^*$, $i = 1, 2, \ldots, m$ are nonnegative, so there is a constant $C_1 > 0$ such that

$$0 \le \lambda_i^* s_i \le C_1\mu, \quad 0 \le \lambda_i s_i^* \le C_1\mu \quad \text{for all } i = 1, 2, \ldots, m.$$

For all $i \in \mathcal{B}$, we have $\lambda_i^* > 0$ by our strictly complementary choice of λ^*, and so

$$0 < s_i \leq \frac{C_1}{\lambda_i^*}\mu \leq \frac{C_1}{\min_{i\in\mathcal{B}} \lambda_i^*}\mu \stackrel{\text{def}}{=} C_2\mu. \tag{3.22}$$

On the other hand, we have by boundedness of $\mathcal{S}_\lambda$ and our assumption (3.11c) that

$$s_i \geq \frac{\gamma\mu}{\lambda_i} \geq \gamma_{\min}\mu \quad \text{for all } i = 1, 2, \ldots, m \tag{3.23}$$

for some constant $\gamma_{\min} > 0$. By combining (3.22) and (3.23), we have that

$$s_i = \Theta(\mu), \quad \text{for all } i \in \mathcal{B}.$$

For the $\lambda_\mathcal{B}$ component, we have that

$$s_i\lambda_i \geq \gamma\mu \;\; \Rightarrow \;\; \lambda_i \geq \frac{\gamma\mu}{s_i} \geq \frac{\gamma}{C_2} \quad \text{for all } i \in \mathcal{B}.$$

By combining this bound with (3.17), we obtain that

$$\lambda_i = \Theta(1) \quad \text{for all } i \in \mathcal{B}.$$

This completes the proof of (3.16a). We omit the proof of (3.16b), which is similar. ∎

Next, we show that when the strict complementarity assumption is added to the assumptions of Lemma 3.1, the upper bound on the distance to the solution set in (3.12) can actually be improved to $O(\mu)$.

THEOREM 3.3. *Suppose that z^* is a solution of* (1.1) *at which the MFCQ condition* (2.5)*, the second-order conditions* (2.6)*,* (2.7)*, and the SC condition* (1.5) *are satisfied. Then for all (z, λ) with $\lambda \geq 0$ for which there is a vector s such that (z, λ, s) satisfies* (3.11)*, we have that*

$$\delta(z, \lambda) = O(\mu). \tag{3.24}$$

Proof. From (3.11a), we have directly that $r_f = O(\mu)$. We have from (3.11) and (3.16a) that

$$g_i(z) = -s_i + (r_g)_i = O(\mu), \quad \lambda_i = \Theta(1), \quad \lambda_i > 0 \quad \text{for all } i \in \mathcal{B},$$

so that

$$\min(-g_i(z), \lambda_i) = -g_i(z) = O(\mu) \quad \text{for all } i \in \mathcal{B}, \tag{3.25}$$

whenever μ is sufficiently small. For the remaining components, we have

$$\min(-g_i(z), \lambda_i) = \lambda_i = O(\mu) \quad \text{for all } i \in \mathcal{N}. \tag{3.26}$$

By substituting (3.11a), (3.25), and (3.26) into (2.12), we obtain the result. ∎

Similar conclusions to Lemma 3.2 and Theorem 3.3 can be reached in the case of linear programming algorithms. The second-order conditions (2.6), (2.7) are not relevant for this class of problems, and the SC assumption (1.5) holds for every linear program that has a solution.

4. Accuracy of PDIP steps: General results. By partitioning the constraint index set $\{1, 2, \ldots, m\}$ into active indices $\mathcal{B}$ and inactive indices $\mathcal{N}$, we can express the system (3.9) without loss of generality as follows:

$$\begin{bmatrix} \mathcal{L}_{zz}(z,\lambda) & \nabla g_{\mathcal{B}}(z) & \nabla g_{\mathcal{N}}(z) \\ \nabla g_{\mathcal{B}}(z)^T & -D_{\mathcal{B}} & 0 \\ \nabla g_{\mathcal{N}}(z)^T & 0 & -D_{\mathcal{N}} \end{bmatrix} \begin{bmatrix} \Delta z \\ \Delta\lambda_{\mathcal{B}} \\ \Delta\lambda_{\mathcal{N}} \end{bmatrix} = \begin{bmatrix} -\mathcal{L}_z(z,\lambda) \\ -g_{\mathcal{B}}(z) + \Lambda_{\mathcal{B}}^{-1} t_{\mathcal{B}} \\ -g_{\mathcal{N}}(z) + \Lambda_{\mathcal{N}}^{-1} t_{\mathcal{N}} \end{bmatrix}, \tag{4.1}$$

where $D_{\mathcal{B}}$ and $D_{\mathcal{N}}$ are positive diagonal matrices defined by

$$D_{\mathcal{B}} = \Lambda_{\mathcal{B}}^{-1} S_{\mathcal{B}}, \qquad D_{\mathcal{N}} = \Lambda_{\mathcal{N}}^{-1} S_{\mathcal{N}}. \tag{4.2}$$

When the SC condition (1.5) is satisfied, we have from Lemma 3.2 that the diagonals of $D_{\mathcal{B}}$ have size $\Theta(\mu)$ while those of $D_{\mathcal{N}}$ have size $\Theta(\mu^{-1})$. By eliminating $\Delta\lambda_{\mathcal{N}}$ from (4.1), we obtain the following intermediate stage between (3.9) and (3.10):

$$\begin{aligned} &\begin{bmatrix} H(z,\lambda) & \nabla g_{\mathcal{B}}(z) \\ \nabla g_{\mathcal{B}}(z)^T & -D_{\mathcal{B}} \end{bmatrix} \begin{bmatrix} \Delta z \\ \Delta\lambda_{\mathcal{B}} \end{bmatrix} \\ &= \begin{bmatrix} -\mathcal{L}_z(z,\lambda) - \nabla g_{\mathcal{N}}(z) D_{\mathcal{N}}^{-1} [g_{\mathcal{N}}(z) - \Lambda_{\mathcal{N}}^{-1} t_{\mathcal{N}}] \\ -g_{\mathcal{B}}(z) + \Lambda_{\mathcal{B}}^{-1} t_{\mathcal{B}} \end{bmatrix}, \end{aligned} \tag{4.3}$$

where we have defined

$$H(z,\lambda) \stackrel{\text{def}}{=} \mathcal{L}_{zz}(z,\lambda) + \nabla g_{\mathcal{N}}(z) D_{\mathcal{N}}^{-1} \nabla g_{\mathcal{N}}(z)^T. \tag{4.4}$$

In this section, we start by proving a key result about the solutions of perturbed forms of the system (4.3). Subsequently, we use this result as the foundation for proving results about the three alternative formulations (3.8), (3.9), and (3.10) of the PDIP step equations. The principal reason for our focus on (4.3) is that the proof of the main result can be derived from fairly standard linear algebra arguments. Gould [13, section 6] obtains a system similar to (4.3) for the Newton equations for the primal log-barrier function, and notes that the matrix approaches a nonsingular limit when certain optimality conditions, including LICQ, are satisfied. Because we replace LICQ by MFCQ, the matrix in (4.3) may approach a singular limit in our case.

We note that the form (4.3) is also relevant to the stabilized sequential quadratic programming (sSQP) method of Wright [29] and Hager [15]; that is, slight modifications to the results of this paper can be used to show that the condensed and augmented formulations of the step equations for the sSQP algorithm yield good steps even in the presence of roundoff errors and cancellation. We omit further details in this paper.

Errors in the step equations arise from cancellation and roundoff errors in evaluating both the matrix and right-hand side and from roundoff errors that arise in the factorization/solution process. We discuss these sources of error further and quantify them in the next section. In this section, we consider the following perturbed version of (4.3):

$$\begin{bmatrix} H(z,\lambda) + \tilde{E}_{11} & \nabla g_{\mathcal{B}}(z) + \tilde{E}_{12} \\ \nabla g_{\mathcal{B}}(z)^T + \tilde{E}_{21} & -D_{\mathcal{B}} + \tilde{E}_{22} \end{bmatrix} \begin{bmatrix} w \\ y \end{bmatrix} = \begin{bmatrix} r_1 \\ \nabla g_{\mathcal{B}}(z^*)^T r_3 + r_4 \end{bmatrix}. \tag{4.5}$$

Here, $\tilde{E}$ is the perturbation matrix (appropriately partitioned and not assumed to be symmetric) and r_1, r_3, and r_4 represent components of a general right-hand side.

Note the partitioning of the second right-hand side component into a component $\nabla g_{\mathcal{B}}(z^*)^T r_3$ in the range space of $\nabla g_{\mathcal{B}}(z^*)^T$ and a remainder term r_4. When LICQ is satisfied, the range space of $\nabla g_{\mathcal{B}}(z^*)^T$ spans the full space, so we can choose r_4 to be zero. Under MFCQ, however, we have in general that r_4 must be nonzero. The main interest of the results below is in isolating the component of the solution of (4.5) that is sensitive to r_4.

To make the results applicable to a wider class of linear systems, we do not impose the assumptions that were needed in the preceding section to ensure that the matrices $D_{\mathcal{B}}$ and $D_{\mathcal{N}}$ defined by (4.2) have diagonals of the appropriate size. Instead, we *assume* that the diagonals have the given size, and derive the application to the linear systems of interest (those that arise in primal-dual interior-point methods) as a special case.

Our results in this and later sections make extensive use of the SVD (2.13) of $\nabla g_{\mathcal{B}}(z^*)$. They also make assumptions about the size of the smallest singular value of this matrix, and about the size of the smallest eigenvalue of $\hat{V}^T \mathcal{L}_{zz}(z^*, \lambda^*)\hat{V}$, the two-sided projection of the Lagrangian Hessian onto the active constraint manifold.

THEOREM 4.1. *Let (z, λ) be an approximate primal-dual solution of* (1.1) *with $\delta(z, \lambda) = O(\mu)$, and suppose the diagonal matrices $D_{\mathcal{B}}$ and $D_{\mathcal{N}}^{-1}$ defined by* (4.2) *have all their diagonal elements of size $\Theta(\mu)$. Suppose that the perturbation submatrices in* (4.5) *satisfy*

$$\tilde{E}_{11} = \delta_{\mathbf{u}}/\mu + O(\mu), \quad \tilde{E}_{21}, \tilde{E}_{12}, \tilde{E}_{22} = \delta_{\mathbf{u}} \tag{4.6}$$

and that the following conditions hold for some $\beta > 0$:

$$\mathbf{u}/\mu \ll 1, \quad \mathbf{u} \ll 1, \tag{4.7a}$$

$$\sigma_{\min}(\Sigma) \geq \beta \max(\mu^{1/3}, \mathbf{u}/\mu), \tag{4.7b}$$

$$\sigma_{\min}(\hat{V}^T \mathcal{L}_{zz}(z^*, \lambda^*)\hat{V}) \geq \beta \max(\mu^{1/3}, \mathbf{u}/\mu) \quad \text{for all } \lambda^* \in \mathcal{S}_\lambda. \tag{4.7c}$$

Then if β is sufficiently large (in a sense to be specified in the proof), the step (w, y) computed from (4.5) *satisfies*

$$(U^T y, \hat{V}^T w, \hat{U}^T w) = O(\|r_1\| + \|r_3\| + \|r_4\|),$$
$$V^T y = O(\|r_1\| + \|r_3\| + \|r_4\|/\mu).$$

Proof. If we define

$$y_U = U^T y, \quad y_V = V^T y, \quad w_{\hat{U}} = \hat{U}^T w, \quad w_{\hat{V}} = \hat{V}^T w,$$

we have

$$y = U y_U + V y_V, \qquad w = \hat{U} w_{\hat{U}} + \hat{V} w_{\hat{V}}.$$

Using this notation, we can rewrite (4.5) as

$$\begin{bmatrix} \hat{U}^T M_{11} \hat{U} & \hat{U}^T M_{11} \hat{V} & \hat{U}^T M_{12} U & \hat{U}^T M_{12} V \\ \hat{V}^T M_{11} \hat{U} & \hat{V}^T M_{11} \hat{V} & \hat{V}^T M_{12} U & \hat{V}^T M_{12} V \\ U^T M_{21} \hat{U} & U^T M_{21} \hat{V} & U^T M_{22} U & U^T M_{22} V \\ V^T M_{21} \hat{U} & V^T M_{21} \hat{V} & V^T M_{22} U & V^T M_{22} V \end{bmatrix} \begin{bmatrix} w_{\hat{U}} \\ w_{\hat{V}} \\ y_U \\ y_V \end{bmatrix} = \begin{bmatrix} \hat{U}^T r_1 \\ \hat{V}^T r_1 \\ U^T \nabla g_{\mathcal{B}}(z^*)^T r_3 + U^T r_4 \\ V^T \nabla g_{\mathcal{B}}(z^*)^T r_3 + V^T r_4 \end{bmatrix}, \tag{4.8}$$

where we have defined

$$M_{11} = H(z,\lambda) + \tilde{E}_{11}, \quad M_{12} = \nabla g_{\mathcal{B}}(z) + \tilde{E}_{12}, \tag{4.9}$$
$$M_{21} = \nabla g_{\mathcal{B}}(z)^T + \tilde{E}_{21}, \quad M_{22} = -D_{\mathcal{B}} + \tilde{E}_{22},$$

and $H(\cdot,\cdot)$ is defined in (4.4). From (2.13), we have

$$V^T \nabla g_{\mathcal{B}}(z^*)^T = 0, \quad U^T \nabla g_{\mathcal{B}}(z^*)^T = \Sigma \hat{U}^T.$$

The fact that V^T annihilates $\nabla g_{\mathcal{B}}(z^*)^T$ is crucial, because it causes the term with r_3 to disappear from the last component of the right-hand side of (4.8), which becomes

$$\begin{bmatrix} \hat{U}^T r_1 \\ \hat{V}^T r_1 \\ \Sigma \hat{U}^T r_3 + U^T r_4 \\ V^T r_4 \end{bmatrix}. \tag{4.10}$$

From the definitions (4.9) and (4.4), the perturbation bound (4.6), our assumptions that $D_{\mathcal{N}}^{-1} = O(\mu)$ and $\delta(z,\lambda) = O(\mu)$, compactness of $\mathcal{S}$, and the fact that $\mathcal{L}_{zz}$ is Lipschitz continuous, we have that

$$M_{11} = \mathcal{L}_{zz}(z^*,\lambda^*) + \delta_{\mathbf{u}}/\mu + O(\mu) \tag{4.11}$$

for some $\lambda^* \in \mathcal{S}_\lambda$. Using these same facts, we have likewise that

$$M_{21} = \nabla g_{\mathcal{B}}(z^*)^T + \delta_{\mathbf{u}} + O(\mu),$$

so by substituting from (2.13), we have that

$$U^T M_{21} \hat{U} = \Sigma + \delta_{\mathbf{u}} + O(\mu), \quad U^T M_{21} \hat{V} = \delta_{\mathbf{u}} + O(\mu), \tag{4.12a}$$
$$V^T M_{21} \hat{U} = \delta_{\mathbf{u}} + O(\mu), \quad V^T M_{21} \hat{V} = \delta_{\mathbf{u}} + O(\mu). \tag{4.12b}$$

Similarly, from the definition of M_{12}, we have

$$\hat{U}^T M_{12} U = \Sigma + \delta_{\mathbf{u}} + O(\mu), \quad \hat{U}^T M_{12} V = \delta_{\mathbf{u}} + O(\mu), \tag{4.13a}$$
$$\hat{V}^T M_{12} U = \delta_{\mathbf{u}} + O(\mu), \quad \hat{V}^T M_{12} V = \delta_{\mathbf{u}} + O(\mu). \tag{4.13b}$$

For the M_{22} block, we have from (4.9) and (4.6) that

$$U^T M_{22} U = -U^T D_{\mathcal{B}} U + \delta_{\mathbf{u}} = O(\mu) + \delta_{\mathbf{u}}, \tag{4.14a}$$
$$U^T M_{22} V = O(\mu) + \delta_{\mathbf{u}}, \quad V^T M_{22} U = O(\mu) + \delta_{\mathbf{u}}, \tag{4.14b}$$
$$V^T M_{22} V = -V^T D_{\mathcal{B}} V + \delta_{\mathbf{u}} = \tilde{M}_{VV} + \delta_{\mathbf{u}}, \tag{4.14c}$$

where $\tilde{M}_{VV} \stackrel{\text{def}}{=} -V^T D_{\mathcal{B}} V$ has all its singular values of size $\Theta(\mu)$, so that

$$\tilde{M}_{VV}^{-1} = \Theta(\mu^{-1}). \tag{4.15}$$

Using these estimates together with (4.10), we can rewrite (4.8) as

$$\left\{ \begin{bmatrix} Q & 0 \\ 0 & \tilde{M}_{VV} \end{bmatrix} + \begin{bmatrix} \hat{E}_{11} & \hat{E}_{12} \\ \hat{E}_{21} & \hat{E}_{22} \end{bmatrix} \right\} \left[\begin{array}{c} w_{\hat{U}} \\ w_{\hat{V}} \\ \hline y_U \\ y_V \end{array}\right] = \left[\begin{array}{c} \hat{U}^T r_1 \\ \hat{V}^T r_1 \\ \hline \Sigma \hat{U}^T r_3 + U^T r_4 \\ V^T r_4 \end{array}\right], \tag{4.16}$$

where

$$Q = \begin{bmatrix} \hat{U}^T\mathcal{L}_{zz}(z^*,\lambda^*)\hat{U} & \hat{U}^T\mathcal{L}_{zz}(z^*,\lambda^*)\hat{V} & \Sigma \\ \hat{V}^T\mathcal{L}_{zz}(z^*,\lambda^*)\hat{U} & \hat{V}^T\mathcal{L}_{zz}(z^*,\lambda^*)\hat{V} & 0 \\ \Sigma & 0 & 0 \end{bmatrix} + \begin{bmatrix} \delta_{\mathbf{u}}/\mu + O(\mu) & \delta_{\mathbf{u}}/\mu + O(\mu) & \delta_{\mathbf{u}} + O(\mu) \\ \delta_{\mathbf{u}}/\mu + O(\mu) & \delta_{\mathbf{u}}/\mu + O(\mu) & 0 \\ \delta_{\mathbf{u}} + O(\mu) & 0 & 0 \end{bmatrix} \tag{4.17}$$

$$\stackrel{\text{def}}{=} \begin{bmatrix} N_{UU} & N_{UV} & \bar{\Sigma}_1 \\ N_{VU} & N_{VV} & 0 \\ \bar{\Sigma}_2 & 0 & 0 \end{bmatrix}, \tag{4.18}$$

while

$$\hat{E}_{11} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \delta_{\mathbf{u}} + O(\mu) \\ 0 & \delta_{\mathbf{u}} + O(\mu) & \delta_{\mathbf{u}} + O(\mu) \end{bmatrix}, \tag{4.19}$$

and

$$\hat{E}_{12}, \hat{E}_{21} = \delta_{\mathbf{u}} + O(\mu) = O(\mu), \quad \hat{E}_{22} = \delta_{\mathbf{u}}. \tag{4.20}$$

For purposes of specifying the required conditions on β in (4.7b) and (4.7c), we define κ to be a constant such that expressions of size $\delta_{\mathbf{u}}$ and $O(\mu)$ that arise in the perturbation terms in the coefficient matrix in (4.16) can be bounded by $\kappa\mathbf{u}$ and $\kappa\mu$, respectively. For example, we suppose that the perturbations in $\bar{\Sigma}_1$, $\bar{\Sigma}_2$, and N_{VV} can be bounded as

$$\|\bar{\Sigma}_1 - \Sigma\| \le \kappa(\mu + \mathbf{u}), \quad \|\bar{\Sigma}_2 - \Sigma\| \le \kappa(\mu + \mathbf{u}), \tag{4.21a}$$

$$\|N_{VV} - \hat{V}^T\mathcal{L}_{zz}(z^*,\lambda^*)\hat{V}\| \le \kappa(\mathbf{u}/\mu + \mu) \tag{4.21b}$$

and that

$$\|\hat{E}_{11}\| \le \kappa(\mathbf{u}+\mu), \ \ \|\hat{E}_{12}\| \le \kappa(\mathbf{u}+\mu), \ \ \|\hat{E}_{21}\| \le \kappa(\mathbf{u}+\mu), \ \ \|\hat{E}_{22}\| \le \kappa\mathbf{u}. \tag{4.22}$$

From (4.21a) and (4.7b), we have that

$$\|\bar{\Sigma}_1 - \Sigma\| \le \kappa \max(\mu^{1/3}, \mathbf{u}/\mu) \le (\kappa/\beta)\sigma_{\min}(\Sigma) \le (\kappa/\beta)\|\Sigma\|.$$

It is therefore easy to show that if β can be chosen large enough that $\beta > 2\kappa$ (while still satisfying (4.7b) and (4.7c)), then

$$\|\bar{\Sigma}_1\| \le 2\|\Sigma\|, \quad \|\bar{\Sigma}_1^{-1}\| \le 2\|\Sigma^{-1}\|. \tag{4.23}$$

Similarly, we can show that

$$\|\bar{\Sigma}_2\| \le 2\|\Sigma\|, \quad \|\bar{\Sigma}_2^{-1}\| \le 2\|\Sigma^{-1}\|, \tag{4.24}$$

$$\|N_{VV}\| \le 2\|\hat{V}^T\mathcal{L}_{zz}(z^*,\lambda^*)\hat{V}\|, \quad \|N_{VV}^{-1}\| \le 2\|(\hat{V}^T\mathcal{L}_{zz}(z^*,\lambda^*)\hat{V})^{-1}\|. \tag{4.25}$$

Note, too, that because of Lipschitz continuity of $\mathcal{L}_{zz}$ and compactness of $\mathcal{S}$, and the bounds (4.7a), the norms of N_{UU}, N_{UV}, N_{VU}, N_{VV}, and Σ are all $O(1)$. Hence the matrix Q is itself invertible, and we have

$$Q^{-1} = \begin{bmatrix} 0 & 0 & \bar{\Sigma}_2^{-1} \\ 0 & N_{VV}^{-1} & -N_{VV}^{-1} N_{VU} \bar{\Sigma}_2^{-1} \\ \bar{\Sigma}_1^{-1} & -\bar{\Sigma}_1^{-1} N_{UV} N_{VV}^{-1} & -\bar{\Sigma}_1^{-1}(N_{UU} - N_{UV} N_{VV}^{-1} N_{VU})\bar{\Sigma}_2^{-1} \end{bmatrix}. \tag{4.26}$$

Noting that

$$(Q + \hat{E}_{11})^{-1} = (I + Q^{-1}\hat{E}_{11})^{-1} Q^{-1}, \tag{4.27}$$

we examine the size of $Q^{-1}\hat{E}_{11}$. Note first from (4.7b) and (4.7c) together with (4.23), (4.24), and (4.25) that

$$\|\bar{\Sigma}_1^{-1}\| \le \frac{2}{\beta}(\mathbf{u}/\mu)^{-1}, \quad \|\bar{\Sigma}_2^{-1}\| \le \frac{2}{\beta}(\mathbf{u}/\mu)^{-1}, \quad \|N_{VV}^{-1}\| \le \frac{2}{\beta}(\mathbf{u}/\mu)^{-1}, \tag{4.28a}$$

$$\|\bar{\Sigma}_1^{-1}\| \le \frac{2}{\beta}\mu^{-1/3}, \quad \|\bar{\Sigma}_2^{-1}\| \le \frac{2}{\beta}\mu^{-1/3}, \quad \|N_{VV}^{-1}\| \le \frac{2}{\beta}\mu^{-1/3}. \tag{4.28b}$$

By forming the product of (4.26) with (4.19) and using the bounds in (4.28), we can show that the norm of $Q^{-1}\hat{E}_{11}$ can be made less than $1/2$ provided that β in (4.7b), (4.7c) is sufficiently large. The $(3,3)$ block of $Q^{-1}\hat{E}_{11}$, for instance, has the form

$$-\bar{\Sigma}_1^{-1} N_{UV} N_{VV}^{-1}(\delta_{\mathbf{u}} + O(\mu)) + \bar{\Sigma}_1^{-1}(N_{UU} - N_{UV} N_{VV}^{-1} N_{VU})\bar{\Sigma}_2^{-1}(\delta_{\mathbf{u}} + O(\mu)).$$

Because of (4.22), its norm can be bounded by a quantity of the form

$$C\kappa \left(\|\bar{\Sigma}_1^{-1}\| \, \|N_{VV}^{-1}\| + \|\bar{\Sigma}_1^{-1}\| \, \|\bar{\Sigma}_2^{-1}\| \, \|N_{VV}^{-1}\| + \|\bar{\Sigma}_1^{-1}\| \, \|\bar{\Sigma}_2^{-1}\| \right) ((\mathbf{u}/\mu)\mu + \mu)$$

(for some C that depends on $\|\mathcal{L}_{zz}(z^*, \lambda^*)\|$), which in turn because of (4.28) is bounded by the following quantity:

$$8C\kappa \left(\frac{1}{\beta^2}\mu^{2/3} + \frac{1}{\beta^3}\mu^{1/3} \right) + 8C\kappa \left(\frac{1}{\beta^2}\mu^{1/3} + \frac{1}{\beta^3} \right).$$

Provided that β is large enough that this and the other blocks of $Q^{-1}\hat{E}_{11}$ can be bounded appropriately, we have that $\|Q^{-1}\hat{E}_{11}\| \le 1/2$, and therefore from (4.27) we have

$$\|(Q + \hat{E}_{11})^{-1}\| = 2\|Q^{-1}\|.$$

Our conclusion is that for β satisfying the conditions outlined in this paragraph, the inverse of the $(1,1)$ block of the matrix in (4.16) can be bounded in terms of $\|Q^{-1}\|$, which because of (4.23), (4.24), (4.25), and (4.26) can in turn be bounded by a finite quantity that depends only on the problem data and not on μ.

Returning to (4.16) and using (4.20), we have that

$$\begin{aligned} \begin{bmatrix} w_{\hat{U}} \\ w_{\hat{V}} \\ y_U \end{bmatrix} &= -(Q + \hat{E}_{11})^{-1}\hat{E}_{12} y_V + (Q + \hat{E}_{11})^{-1} \begin{bmatrix} \hat{U}^T r_1 \\ \hat{V}^T r_1 \\ \Sigma \hat{U}^T r_3 + U^T r_4 \end{bmatrix} \\ &= O(\|\hat{E}_{12}\| \|y_V\|) + O(\|r_1\| + \|r_3\| + \|r_4\|) \\ &= O(\mu)\|y_V\| + O(\|r_1\| + \|r_3\| + \|r_4\|). \end{aligned} \tag{4.29}$$

Meanwhile, for the second block row of (4.16), we obtain

$$y_V = -(\tilde{M}_{VV} + \hat{E}_{22})^{-1}\hat{E}_{21}\begin{bmatrix} w_{\hat{U}} \\ w_{\hat{V}} \\ y_U \end{bmatrix} + (\tilde{M}_{VV} + \hat{E}_{22})^{-1}V^T r_4. \tag{4.30}$$

Since from (4.15), (4.20), and (4.7a) we have

$$(\tilde{M}_{VV} + \hat{E}_{22})^{-1} = (I + \tilde{M}_{VV}^{-1}\hat{E}_{22})^{-1}\tilde{M}_{VV}^{-1} = (I + \delta_{\mathbf{u}}/\mu)\tilde{M}_{VV}^{-1} = O(\mu^{-1}),$$

it follows from (4.30) and (4.20) that

$$y_V = O(\mu^{-1})O(\mu)\left\|\begin{bmatrix} w_{\hat{U}} \\ w_{\hat{V}} \\ y_U \end{bmatrix}\right\| + O(\mu^{-1})O(\|r_4\|).$$

By substituting from (4.29), we obtain

$$\|y_V\| = O(\mu)\|y_V\| + O(\|r_1\| + \|r_3\| + \|r_4\|) + O(\|r_4\|/\mu),$$

and therefore

$$\|y_V\| = O(\|r_1\| + \|r_3\| + \|r_4\|/\mu),$$

as claimed. The estimate for $(w_{\hat{U}}, w_{\hat{V}}, y_U)$ is obtained by substituting into (4.29). □

The conditions (4.7) need a little explanation. For the typical value $\mathbf{u} = 10^{-16}$, the minimum value of the quantity $\max(\mu^{1/3}, \mathbf{u}/\mu)$ is 10^{-4}, achieved at μ^{-12}. Moreover, we have $\max(\mu^{1/3}, \mathbf{u}/\mu) \leq 10^{-2}$ only for μ in the range $[10^{-14}, 10^{-6}]$. It would seem, then, that the problem would need to be quite well conditioned for (4.7b) and (4.7c) to hold and that μ may have to become quite small before the results apply. We note, however, that the purpose of the bounds (4.7b) and (4.7c) is to ensure that the inverse of $Q + \hat{E}_{11}$ can be bounded independently of μ, and that for this purpose they are quite conservative. That is, we would expect to find that $\|(Q + \hat{E}_{11})^{-1}\|$ is not too much larger than the norm of the inverse of the corresponding exact matrix (the first term on the right-hand side of (4.17)) for μ not much less than the smallest eigenvalues of Σ and $\hat{V}^T\mathcal{L}_{zz}(z^*, \lambda^*)\hat{V}$.

The requirement that $\mathbf{u}/\mu$ and μ both be small in (4.7) may not seem to sit well with expressions such as $O(\mu)$ and $O(\mu^2)$, which crop up repeatedly in the analysis and which assert that certain bounds hold "for all sufficiently small μ." As noted in the preceding paragraph, this requirement implies that the analysis holds for μ in a certain range, or "window," of values. Similar windows are used in the analysis of S. Wright [24, 27, 30] and M. H. Wright [23], and numerical experience indicates that such a window does indeed exist in most practical cases. We expect the same to be true of the problem and algorithms discussed in this paper.

At this point, we assemble the assumptions that are made in the remainder of the paper into a single catch-all assumption.

Assumption 4.1.

(a) z^* is a solution of (1.1), so that the condition (1.3) holds. The MFCQ condition (2.5), the second-order conditions (2.6), (2.7), and the SC condition (1.5) are satisfied at this solution. The current iterate (z, λ, s) of the PDIP algorithm satisfies the conditions (3.11), and the right-hand side modification t satisfies (3.7).

(b) The quantities μ, $\mathbf{u}$ (2.10), $\mathcal{L}_{zz}(z^*, \lambda^*)$, Σ, and $\hat{V}$ (2.13) satisfy the conditions (4.7).

From our observations following (4.2), we have under this assumption that

$$D_{\mathcal{B}} = O(\mu), \quad D_{\mathcal{B}}^{-1} = O(\mu^{-1}), \quad D_{\mathcal{N}} = O(\mu^{-1}), \quad D_{\mathcal{N}}^{-1} = O(\mu). \tag{4.31}$$

Our next result considers a perturbed form of the system (4.1) with a general right-hand side. By eliminating one component to obtain the form (4.3), we can apply Theorem 4.1 to obtain estimates of the dependence of the solution on the right-hand side components.

THEOREM 4.2. *Suppose that Assumption* 4.1 *holds. Consider the linear system*

$$\begin{bmatrix} \mathcal{L}_{zz}(z,\lambda) + E_{11} & \nabla g_{\mathcal{B}}(z) + E_{12} & \nabla g_{\mathcal{N}}(z) + E_{13} \\ \nabla g_{\mathcal{B}}(z)^T + E_{21} & -D_{\mathcal{B}} + E_{22} & E_{23} \\ \nabla g_{\mathcal{N}}(z)^T + E_{31} & E_{32} & -D_{\mathcal{N}} + E_{33} \end{bmatrix} \begin{bmatrix} w \\ y \\ q \end{bmatrix} = \begin{bmatrix} r_5 \\ \nabla g_{\mathcal{B}}(z^*)^T r_6 + r_7 \\ r_8 \end{bmatrix}, \tag{4.32}$$

where

$$E_{11} = \delta_{\mathbf{u}}/\mu, \quad E_{33} = \delta_{\mathbf{u}}/\mu^2, \tag{4.33a}$$

$$E_{12}, E_{21}, E_{22} = \delta_{\mathbf{u}}, \quad E_{13}, E_{31}, E_{23}, E_{32} = \delta_{\mathbf{u}}/\mu. \tag{4.33b}$$

Then the step (w, y, q) *satisfies the following estimates:*

$$\begin{aligned} (U^T y, w) &= O(\|r_5\| + \|r_6\| + \|r_7\| + \mu\|r_8\|), \\ V^T y &= O(\|r_5\| + \|r_6\| + \|r_7\|/\mu + (\delta_{\mathbf{u}}/\mu + O(\mu))\|r_8\|), \\ q &= O(\mu)\,[\|r_5\| + \|r_6\| + \|r_8\|] + (\delta_{\mathbf{u}}/\mu + O(\mu))\|r_7\|. \end{aligned}$$

Proof. From (4.31) and the assumed bound (4.33a) on the size of E_{33}, we have that

$$\begin{aligned} &(-D_{\mathcal{N}} + E_{33})^{-1} \\ &= -(I - D_{\mathcal{N}}^{-1} E_{33})^{-1} D_{\mathcal{N}}^{-1} = (I + O(\mu)\delta_{\mathbf{u}}/\mu^2) O(\mu) = O(\mu). \end{aligned} \tag{4.34}$$

By eliminating q from (4.32), we obtain the reduced system

$$\begin{bmatrix} H(z,\lambda) + \tilde{E}_{11} & \nabla g_{\mathcal{B}}(z) + \tilde{E}_{12} \\ \nabla g_{\mathcal{B}}(z)^T + \tilde{E}_{21} & -D_{\mathcal{B}} + \tilde{E}_{22} \end{bmatrix} \begin{bmatrix} w \\ y \end{bmatrix} = \begin{bmatrix} r_5 + O(\mu)\|r_8\| \\ \nabla g_{\mathcal{B}}(z^*)^T r_6 + r_7 + \delta_{\mathbf{u}}\|r_8\| \end{bmatrix},$$

where from (4.7) and (4.4), we obtain

$$\begin{aligned} \tilde{E}_{11} &= E_{11} - (\nabla g_{\mathcal{N}}(z) + E_{13})(-D_{\mathcal{N}} + E_{33})^{-1}(\nabla g_{\mathcal{N}}(z)^T + E_{31}) - \nabla g_{\mathcal{N}}(z) D_{\mathcal{N}}^{-1} \nabla g_{\mathcal{N}}(z)^T \\ &= \delta_{\mathbf{u}}/\mu + O(\mu), \\ \tilde{E}_{12} &= E_{12} - (\nabla g_{\mathcal{N}}(z) + E_{13})(-D_{\mathcal{N}} + E_{33})^{-1} E_{32} = \delta_{\mathbf{u}} + O(1)O(\mu)\delta_{\mathbf{u}}/\mu = \delta_{\mathbf{u}}, \\ \tilde{E}_{21} &= E_{21} - E_{23}(-D_{\mathcal{N}} + E_{33})^{-1}(\nabla g_{\mathcal{N}}(z)^T + E_{31}) = \delta_{\mathbf{u}} + (\delta_{\mathbf{u}}/\mu)O(\mu)O(1) = \delta_{\mathbf{u}}, \\ \tilde{E}_{22} &= E_{22} - E_{23}(-D_{\mathcal{N}} + E_{33})^{-1} E_{32} = \delta_{\mathbf{u}} + (\delta_{\mathbf{u}}/\mu)^2 O(\mu) = \delta_{\mathbf{u}}. \end{aligned}$$

These perturbation matrices satisfy the assumptions of Theorem 4.1, which can be applied to give

$$(U^T y, \hat{V}^T w, \hat{U}^T w) = O(\|r_5\| + \|r_6\| + \|r_7\| + \mu\|r_8\|), \tag{4.35a}$$

$$V^T y = O(\|r_5\| + \|r_6\| + \|r_7\|/\mu) + (\delta_{\mathbf{u}}/\mu + O(\mu))\|r_8\|. \tag{4.35b}$$

From the last block row of (4.32), and using (4.7), (4.34), (4.35), we obtain

$$\begin{aligned} q &= (-D_{\mathcal{N}} + E_{33})^{-1}\left[r_8 - (\nabla g_{\mathcal{N}}(z)^T + E_{31})w - E_{32}y\right] \\ &= O(\mu)\left[\|r_8\| + \|w\| + (\delta_{\mathbf{u}}/\mu)\|y\|\right] \\ &= O(\mu)\left[\|r_5\| + \|r_6\| + \|r_7\| + \|r_8\|\right] \\ &\qquad +\delta_{\mathbf{u}}\left[\|r_5\| + \|r_6\| + \|r_7\|/\mu + (\delta_{\mathbf{u}}/\mu + O(\mu))\|r_8\|\right] \\ &= O(\mu)\left[\|r_5\| + \|r_6\| + \|r_8\|\right] + (\delta_{\mathbf{u}}/\mu + O(\mu))\|r_7\|. \qquad \square \end{aligned}$$

An estimate for the solution of the exact system (3.8) follows almost immediately from this result. This is the key technical result used by Ralph and Wright [21, 22] to prove superlinear convergence of PDIP algorithms for convex programming problems. The result below, however, does not require a convexity assumption.

COROLLARY 4.3. *Suppose that Assumption* 4.1(a) *holds. Then the (exact) solution* $(\Delta z, \Delta\lambda, \Delta s)$ *of the system* (3.8) *satisfies*

$$(\Delta z, \Delta\lambda, \Delta s) = O(\mu). \tag{4.36}$$

Proof. Note first that Assumption 4.1(b) holds trivially in this case for μ sufficiently small, because our assumption of exact computations is equivalent to setting $\mathbf{u} = 0$. We prove the result by identifying the system (4.1) with (4.32) and then applying Theorem 4.2.

For the right-hand side, we note first that, because of smoothness of g, Taylor's theorem, the definition (2.1) of $\mathcal{B}$, and Theorem 3.3,

$$\begin{aligned} g_{\mathcal{B}}(z) &= g_{\mathcal{B}}(z^*) + \nabla g_{\mathcal{B}}(z^*)^T(z - z^*) + O(\|z - z^*\|^2) \\ &= \nabla g_{\mathcal{B}}(z^*)^T(z - z^*) + O(\mu^2). \end{aligned} \tag{4.37}$$

We now identify the right-hand side of (4.1) with (4.32) by setting

$$\begin{aligned} r_5 &= -\mathcal{L}_z(z, \lambda), \\ r_6 &= (z - z^*), \\ r_7 &= -\nabla g_{\mathcal{B}}(z^*)^T(z - z^*) - g_{\mathcal{B}}(z) + \Lambda_{\mathcal{B}}^{-1} t_{\mathcal{B}}, \\ r_8 &= -g_{\mathcal{N}}(z) + \Lambda_{\mathcal{N}}^{-1} t_{\mathcal{N}}. \end{aligned}$$

The sizes of these vectors can be estimated by using (3.11), Lemma 3.2, (4.37), Theorem 3.3, and the assumption (3.7) on the size of t to obtain

$$r_5 = O(\mu), \quad r_6 = O(\mu), \quad r_7 = O(\mu^2), \quad r_8 = O(1). \tag{4.38}$$

(By choosing r_6 and r_7 in this way, we ensure that the terms involving $\|r_7\|/\mu$ in the estimates of the solution components in Theorem 4.2 are not grossly larger than the other terms in these expressions.) We complete the identification of (4.1) with (4.32) by setting all the perturbation matrices $E_{11}, E_{12}, \ldots, E_{33}$ to zero and by identifying

the solution vector components Δz, $\Delta\lambda_{\mathcal{B}}$, and $\Delta\lambda_{\mathcal{N}}$ with w, y, and q, respectively. By directly applying Theorem 4.2, substituting the estimates (4.38), and setting $\delta_{\mathbf{u}} = 0$ (since we are assuming exact computations), we have that

$$(U^T\Delta\lambda_{\mathcal{B}}, \Delta z) = O(\mu), \quad V^T\Delta\lambda_{\mathcal{B}} = O(\mu), \quad \Delta\lambda_{\mathcal{N}} = O(\mu).$$

To show that the remaining solution component Δs of (3.8) is also of size $O(\mu)$, we write the second block row in (3.8) as

$$\Delta s = -(g(z) + s) - \nabla g(z)^T\Delta z,$$

from which the desired estimate follows immediately by substituting from (3.11b) and $\Delta z = O(\mu)$. ∎

The next result uses Theorem 4.2 to compare perturbed and exact solutions of the system of the system (4.1).

COROLLARY 4.4. *Suppose that Assumption* 4.1 *holds. Let* (w, y, q) *be obtained from the following perturbed version of* (3.9)*:*

$$\begin{bmatrix} \mathcal{L}_{zz}(z,\lambda) + E_{11} & \nabla g_{\mathcal{B}}(z) + E_{12} & \nabla g_{\mathcal{N}}(z) + E_{13} \\ \nabla g_{\mathcal{B}}(z)^T + E_{21} & -D_{\mathcal{B}} + E_{22} & E_{23} \\ \nabla g_{\mathcal{N}}(z)^T + E_{31} & E_{32} & -D_{\mathcal{N}} + E_{33} \end{bmatrix} \begin{bmatrix} w \\ y \\ q \end{bmatrix} = \begin{bmatrix} -\mathcal{L}_z(z,\lambda) + f_1 \\ -g_{\mathcal{B}}(z) + \Lambda_{\mathcal{B}}^{-1}t_{\mathcal{B}} + f_2 \\ -g_{\mathcal{N}}(z) + \Lambda_{\mathcal{N}}^{-1}t_{\mathcal{N}} + f_3 \end{bmatrix}, \tag{4.39}$$

where E_{ij}, $i, j = 1, 2, 3$, *satisfy the conditions* (4.33) *and* f_1, f_2, *and* f_3 *are all of size* $\delta_{\mathbf{u}}$. *Then if* $(\Delta z, \Delta\lambda, \Delta s)$ *is the (exact) solution of the system* (3.8)*, we have*

$$\begin{aligned} (\Delta z - w, U^T(\Delta\lambda_{\mathcal{B}} - y)) &= \delta_{\mathbf{u}}, \\ V^T(\Delta\lambda_{\mathcal{B}} - y) &= \delta_{\mathbf{u}}/\mu, \\ \Delta\lambda_{\mathcal{N}} - q &= \delta_{\mathbf{u}}. \end{aligned}$$

Proof. By combining (4.39) with (4.1), we obtain

$$\begin{aligned} &\begin{bmatrix} \mathcal{L}_{zz}(z,\lambda) + E_{11} & \nabla g_{\mathcal{B}}(z) + E_{12} & \nabla g_{\mathcal{N}}(z) + E_{13} \\ \nabla g_{\mathcal{B}}(z)^T + E_{21} & -D_{\mathcal{B}} + E_{22} & E_{23} \\ \nabla g_{\mathcal{N}}(z)^T + E_{31} & E_{32} & -D_{\mathcal{N}} + E_{33} \end{bmatrix} \begin{bmatrix} w - \Delta z \\ y - \Delta\lambda_{\mathcal{B}} \\ q - \Delta\lambda_{\mathcal{N}} \end{bmatrix} \\ &\qquad = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} - \begin{bmatrix} E_{11} & E_{12} & E_{13} \\ E_{21} & E_{22} & E_{23} \\ E_{31} & E_{32} & E_{33} \end{bmatrix} \begin{bmatrix} \Delta z \\ \Delta\lambda_{\mathcal{B}} \\ \Delta\lambda_{\mathcal{N}} \end{bmatrix}. \end{aligned} \tag{4.40}$$

From the bounds on the perturbations E in (4.33) and the result of Corollary 4.3, we have for the right-hand side of this expression that

$$\begin{aligned} \begin{bmatrix} r_5 \\ r_7 \\ r_8 \end{bmatrix} &\stackrel{\text{def}}{=} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} - \begin{bmatrix} E_{11} & E_{12} & E_{13} \\ E_{21} & E_{22} & E_{23} \\ E_{31} & E_{32} & E_{33} \end{bmatrix} \begin{bmatrix} \Delta z \\ \Delta\lambda_{\mathcal{B}} \\ \Delta\lambda_{\mathcal{N}} \end{bmatrix} \\ &= \begin{bmatrix} \delta_{\mathbf{u}} + (\delta_{\mathbf{u}}/\mu)\mu + \delta_{\mathbf{u}}\mu + (\delta_{\mathbf{u}}/\mu)\mu \\ \delta_{\mathbf{u}} + \delta_{\mathbf{u}}\mu + \delta_{\mathbf{u}}\mu + (\delta_{\mathbf{u}}/\mu)\mu \\ \delta_{\mathbf{u}} + (\delta_{\mathbf{u}}/\mu)\mu + (\delta_{\mathbf{u}}/\mu)\mu + (\delta_{\mathbf{u}}/\mu^2)\mu \end{bmatrix} = \begin{bmatrix} \delta_{\mathbf{u}} \\ \delta_{\mathbf{u}} \\ \delta_{\mathbf{u}}/\mu \end{bmatrix}. \end{aligned} \tag{4.41}$$

Using these estimates, we can simply apply Theorem 4.2 to (4.40) (with $r_6 = 0$) to obtain the result. □

For later reference, we show how the estimates of Corollary 4.4 can be modified when the perturbations have a special form. Suppose that

$$E_{23} = 0, \quad E_{33} = \delta_{\mathbf{u}}/\mu, \quad f_2 = U f_2^U + O(\mu^2), \text{ where } f_2^U = \delta_{\mathbf{u}}, \tag{4.42}$$

where U is the matrix from (2.13). Instead of setting $r_6 = 0$ as in the proof above, we set

$$r_6 = \hat{U}\Sigma f_2^U = \delta_{\mathbf{u}}$$

(using (2.13) to obtain an r_6 for which $\nabla g_{\mathcal{B}}(z^*)^T r_6 = U f_2^U$). By modifying (4.41) to account for the remaining perturbations, we can identify (4.40) with (4.32) by setting

$$\begin{aligned} \begin{bmatrix} r_5 \\ r_7 \\ r_8 \end{bmatrix} &\stackrel{\text{def}}{=} \begin{bmatrix} f_1 \\ f_2 - U f_2^U \\ f_3 \end{bmatrix} - \begin{bmatrix} E_{11} & E_{12} & E_{13} \\ E_{21} & E_{22} & E_{23} \\ E_{31} & E_{32} & E_{33} \end{bmatrix} \begin{bmatrix} \Delta z \\ \Delta\lambda_{\mathcal{B}} \\ \Delta\lambda_{\mathcal{N}} \end{bmatrix} \\ &= \begin{bmatrix} \delta_{\mathbf{u}} + (\delta_{\mathbf{u}}/\mu)\mu + \delta_{\mathbf{u}}\mu + (\delta_{\mathbf{u}}/\mu)\mu \\ O(\mu^2) + \delta_{\mathbf{u}}\mu + \delta_{\mathbf{u}}\mu \\ \delta_{\mathbf{u}} + (\delta_{\mathbf{u}}/\mu)\mu + (\delta_{\mathbf{u}}/\mu)\mu + (\delta_{\mathbf{u}}/\mu^2)\mu \end{bmatrix} = \begin{bmatrix} \delta_{\mathbf{u}} \\ O(\mu^2) \\ \delta_{\mathbf{u}}/\mu \end{bmatrix}. \end{aligned} \tag{4.43}$$

Using these modified right-hand side estimates, we can apply Theorem 4.2 to obtain the following improved bound on one of the components:

$$V^T(\Delta\lambda_{\mathcal{B}} - y) = O(\mu). \tag{4.44}$$

The bounds on the other components remain unchanged.

We emphasize that the conditions (3.11), and in particular (3.11c), are critical to the results of this and all the remaining sections of the paper. These conditions enable Lemma 3.2, which in turn enable us to assert that the diagonals of $D_{\mathcal{B}}$ all have size $\Theta(\mu)$ while those of $D_{\mathcal{N}}$ all have size $\Theta(\mu^{-1})$ (see (4.31)). This neat classification of the diagonals of D into two categories drives all subsequent analyses. The motivation for conditions like (3.11) in the literature for path-following methods (with exact steps) is not unrelated: It allows us to obtain bounds on the steps and to show that we can move a significant distance along this direction while ensuring that (3.11) continues to be satisfied at the new iterate. (See, for example, [26, Chapters 5 and 6] and its bibliography for the case of linear programming and [31, 21, 22] for the case of nonlinear convex programming.) In the analysis above, we obtain bounds on the *errors* (rather than the steps themselves) when perturbation terms of a certain structure appear in the matrix and right-hand side.

Many practical implementations of path-following methods for linear programming do not explicitly check that the condition (3.11c) is satisfied by the calculated iterates (see, for example, [19] and [5]). However, the heuristics for "stepping back" from the boundary of the nonnegative orthant by a small but significant quantity are motivated by this condition, and it is observed to hold in practice on all but the most recalcitrant problems.

5. The condensed system. Here we consider an algorithm in which the condensed linear system (3.10) is formed and solved to obtain Δz, and the remaining step components $\Delta\lambda$ and Δs are recovered from (3.8). We obtain expressions for the errors in the calculated step $(\widehat{\Delta z}, \widehat{\Delta\lambda}, \widehat{\Delta s})$ and discuss the effects of these errors on

certain measures of step quality. We also derive conditions under which the Cholesky factorization applied to (3.10) is guaranteed to run to completion.

Formally, the complete procedure can be described as follows.

procedure condensed
given the current iterate (z, λ, s)
form the coefficient matrix and right-hand side for (3.10);
solve (3.10) using a backward stable algorithm to obtain Δz;
set $\Delta\lambda = D^{-1}[g(z) - \Lambda^{-1}t + \nabla g(z)^T \Delta z]$;
set $\Delta s = -(g(z) + s) - \nabla g(z)^T \Delta z$.

We have used the definition (4.2) of the matrix D. For convenience, we restate the system (3.10) here as follows:

$$\left[\mathcal{L}_{zz}(z,\lambda) + \nabla g(z) D^{-1} \nabla g(z)^T\right] \Delta z = -\mathcal{L}_z(z,\lambda) - \nabla g(z) D^{-1}[g(z) - \Lambda^{-1}t]. \tag{5.1}$$

Note that this procedure requires evaluation of $D^{-1} = S^{-1}\Lambda$, rather than D itself.

5.1. Quantifying the errors. When implemented in finite-precision arithmetic, solution of (5.1) gives rise to errors of three types:

- cancellation in evaluation of the matrix and right-hand side;
- roundoff errors in evaluation of the matrix and right-hand side;
- roundoff errors that accumulate during the process of factoring the matrix and using triangular substitutions to obtain the solution.

Cancellation may be an issue in the evaluation of the nonlinear functions $\mathcal{L}_{zz}(z,\lambda)$, $\mathcal{L}_z(z,\lambda)$, $g(z)$, and $\nabla g(z)$, because intermediate terms computed during the additive evaluation of these quantities may exceed the size of the final result (see Golub and Van Loan [12, p. 61]). The intermediate terms generally contain rounding error (which occurs whenever real numbers are represented in finite precision). Cancellation becomes a significant phenomenon whenever we take a difference of two nearly equal quantities, since the error in the computed result due to roundoff in the two arguments may be large relative to the size of the result. If, as we can reasonably assume, intermediate quantities in the calculations of our right-hand sides remain bounded, the absolute size of the errors in the result is $\delta_{\mathbf{u}}$. In the case of $\mathcal{L}_z(z,\lambda)$ and $g_{\mathcal{B}}(z)$, the final result in exact arithmetic has size $O(\mu)$, so that the error $\delta_{\mathbf{u}}$ takes on a large relative significance for small values of μ. This fact causes the error bound in some components of the solution to be larger than in others, as we see in (5.6c) below. In summary, the computed versions of the quantities discussed above differ from their exact values in the following way:

$$\text{computed } \mathcal{L}_{zz}(z,\lambda) \leftarrow \mathcal{L}_{zz}(z,\lambda) + \bar{F}, \tag{5.2a}$$

$$\text{computed } \mathcal{L}_z(z,\lambda) \leftarrow \mathcal{L}_z(z,\lambda) + \bar{f}, \tag{5.2b}$$

$$\text{computed } \nabla g(z) \leftarrow \nabla g(z) + F = \begin{bmatrix} \nabla g_{\mathcal{B}}(z) \\ \nabla g_{\mathcal{N}}(z) \end{bmatrix} + \begin{bmatrix} F_{\mathcal{B}} \\ F_{\mathcal{N}} \end{bmatrix}, \tag{5.2c}$$

$$\text{computed } g(z) \leftarrow g(z) + f = \begin{bmatrix} g_{\mathcal{B}}(z) \\ g_{\mathcal{N}}(z) \end{bmatrix} + \begin{bmatrix} f_{\mathcal{B}} \\ f_{\mathcal{N}} \end{bmatrix}, \tag{5.2d}$$

where $\bar{F}$, $\bar{f}$, F, and f are all of size $\delta_{\mathbf{u}}$. Earlier discussion of cancellation in similar contexts can be found in the papers of S. Wright [24, 27, 30] and M. H. Wright [23].

The second source of error is evaluation of the matrix D^{-1}. From the model (2.10) of floating-point arithmetic and the estimates (3.16) of Lemma 3.2, we have

that

$$\text{computed } D_{\mathcal{B}}^{-1} \leftarrow (D_{\mathcal{B}} + G_{\mathcal{B}})^{-1}, \quad G_{\mathcal{B}} = \mu\delta_{\mathbf{u}}, \tag{5.3a}$$

$$\text{computed } D_{\mathcal{N}}^{-1} \leftarrow (D_{\mathcal{N}} + G_{\mathcal{N}})^{-1}, \quad G_{\mathcal{N}} = \delta_{\mathbf{u}}/\mu, \tag{5.3b}$$

where $G_{\mathcal{B}}$ and $G_{\mathcal{N}}$ are both diagonal matrices that can be composed into a single diagonal matrix G.

Third, we account for the error in forming the matrix and right-hand side of (5.1) from the computed quantities described in the last two paragraphs. Since we are now dealing with floating-point numbers, the model (2.10) applies; that is, any additional errors that arise during the combination of these floating-point quantities have size $\mathbf{u}$ relative to the size of the result of the calculation. Since the norm of the coefficient matrix is of size $O(\mu^{-1})$ while the right-hand side has size $O(1)$ (see (3.11)), we represent these errors by a matrix $\hat{F}$ of size $\delta_{\mathbf{u}}/\mu$ and a vector $\hat{f}$ of size $\delta_{\mathbf{u}}$.

Finally, we account for the error that arises in the application of a backward-stable method to solve (5.1). Specifically, we assume that the method yields a computed solution that is the exact solution of a nearby problem whose data contains relative perturbations of size $\mathbf{u}$. The absolute sizes of these terms would therefore be $\delta_{\mathbf{u}}/\mu$ in the case of the matrix and $\delta_{\mathbf{u}}$ in the case of the right-hand side. Since these errors are the same size as those discussed in the preceding paragraph, we incorporate them into the matrix $\hat{F}$ and the vector $\hat{f}$.

Summarizing, we find that the computed solution $\widehat{\Delta z}$ of (5.1) satisfies the following system:

$$\begin{aligned}\Big[\mathcal{L}_{zz}(z,\lambda) + \bar{F} + (\nabla g(z) + F)(D+G)^{-1}(\nabla g(z) + F)^T + \hat{F}\Big]\,\widehat{\Delta z} \\ = -\mathcal{L}_z(z,\lambda) - \bar{f} - (\nabla g(z) + F)(D+G)^{-1}[g(z) + f - \Lambda^{-1}t] + \hat{f},\end{aligned} \tag{5.4}$$

where the perturbation terms $\bar{F}$, F, $\hat{F}$, G, $\bar{f}$, $\hat{f}$, and f are described in the paragraphs above. By "unfolding" this system and using the partitioning of F, G, and f defined in (5.2) and (5.3), we find that $\widehat{\Delta z}$ also satisfies the following system for some vectors y and q:

$$\begin{bmatrix} \mathcal{L}_{zz}(z,\lambda) + \bar{F} + \hat{F} & \nabla g_{\mathcal{B}}(z) + F_{\mathcal{B}} & \nabla g_{\mathcal{N}}(z) + F_{\mathcal{N}} \\ \nabla g_{\mathcal{B}}(z)^T + F_{\mathcal{B}}^T & -D_{\mathcal{B}} - G_{\mathcal{B}} & 0 \\ \nabla g_{\mathcal{N}}(z)^T + F_{\mathcal{N}}^T & 0 & -D_{\mathcal{N}} - G_{\mathcal{N}} \end{bmatrix} \begin{bmatrix} \widehat{\Delta z} \\ y \\ q \end{bmatrix} = \begin{bmatrix} -\mathcal{L}_z(z,\lambda) - \bar{f} + \hat{f} \\ -g_{\mathcal{B}}(z) + \Lambda_{\mathcal{B}}^{-1}t_{\mathcal{B}} - f_{\mathcal{B}} \\ -g_{\mathcal{N}}(z) + \Lambda_{\mathcal{N}}^{-1}t_{\mathcal{N}} - f_{\mathcal{N}} \end{bmatrix}. \tag{5.5}$$

This system has precisely the form of (4.39) (in particular, the perturbation matrices satisfy the appropriate bounds). Hence, by a direct application of Corollary 4.4, we conclude that

$$\Delta z - \widehat{\Delta z} = \delta_{\mathbf{u}}, \tag{5.6a}$$

$$U^T(\Delta\lambda_{\mathcal{B}} - y) = \delta_{\mathbf{u}}, \tag{5.6b}$$

$$V^T(\Delta\lambda_{\mathcal{B}} - y) = \delta_{\mathbf{u}}/\mu. \tag{5.6c}$$

We return now to the recovery of the remaining solution components $\widehat{\Delta\lambda}$ and $\widehat{\Delta s}$ in the procedure **condensed**. We have from Assumption 4.1 together with (3.11b),

Lemma 3.2, (4.36), (5.6a), (5.3a), (4.7), and (4.31) that

$$g_{\mathcal{B}}(z) = r_g(z,s)_{\mathcal{B}} - s_{\mathcal{B}} = O(\mu), \quad \Lambda_{\mathcal{B}}^{-1} = \Theta(1), \quad \widehat{\Delta z} = \Delta z + \delta_{\mathbf{u}} = O(\mu), \tag{5.7a}$$

$$(D_{\mathcal{B}} + G_{\mathcal{B}})^{-1} = (I + D_{\mathcal{B}}^{-1} G_{\mathcal{B}}) D_{\mathcal{B}}^{-1} = (I + \delta_{\mathbf{u}})^{-1} O(\mu^{-1}) = O(\mu^{-1}). \tag{5.7b}$$

Since $t = O(\mu^2)$, we have from our model (2.10) that the floating-point version of the calculation of $\widehat{\Delta\lambda}_{\mathcal{B}}$ in the procedure **condensed** satisfies the following:

$$\widehat{\Delta\lambda}_{\mathcal{B}} = (D_{\mathcal{B}} + G_{\mathcal{B}})^{-1} \left[g_{\mathcal{B}}(z) + f_{\mathcal{B}} - \Lambda_{\mathcal{B}}^{-1} t_{\mathcal{B}} + (\nabla g_{\mathcal{B}}(z) + F_{\mathcal{B}})^T \widehat{\Delta z} + \mu \delta_{\mathbf{u}} \right] + \delta_{\mathbf{u}}.$$

(The final term $\delta_{\mathbf{u}}$ arises from (2.10) because our best estimate of the quantity in the brackets at this point of the analysis is $O(\mu)$, so from (5.7b) the result has size $O(1)$.) Meanwhile, we have from the second block row of (5.5) that

$$y = (D_{\mathcal{B}} + G_{\mathcal{B}})^{-1} \left[g_{\mathcal{B}}(z) + f_{\mathcal{B}} - \Lambda_{\mathcal{B}}^{-1} t_{\mathcal{B}} + (\nabla g_{\mathcal{B}}(z) + F_{\mathcal{B}})^T \widehat{\Delta z} \right].$$

By a direct comparison of these two expressions, and using $(D_{\mathcal{B}} + G_{\mathcal{B}})^{-1} = O(\mu)$, we find that

$$\widehat{\Delta\lambda}_{\mathcal{B}} - y = \delta_{\mathbf{u}}. \tag{5.8}$$

By combining (5.8) with (5.6b) and (5.6c), we find that

$$U^T(\Delta\lambda_{\mathcal{B}} - \widehat{\Delta\lambda}_{\mathcal{B}}) = \delta_{\mathbf{u}}, \quad V^T(\Delta\lambda_{\mathcal{B}} - \widehat{\Delta\lambda}_{\mathcal{B}}) = \delta_{\mathbf{u}}/\mu. \tag{5.9}$$

For the "nonbasic" part $\widehat{\Delta\lambda}_{\mathcal{N}}$, we have from (3.11b), Lemma 3.2, (4.36), (5.6a), (5.3b), (4.7), and (4.31) that

$$g_{\mathcal{N}}(z) = O(1), \quad \Lambda_{\mathcal{N}}^{-1} = O(\mu^{-1}), \quad \widehat{\Delta z} = O(\mu), \tag{5.10a}$$

$$(D_{\mathcal{N}} + G_{\mathcal{N}})^{-1} = (I + D_{\mathcal{N}}^{-1} G_{\mathcal{N}})^{-1} D_{\mathcal{N}}^{-1} = D_{\mathcal{N}}^{-1} + \mu\delta_{\mathbf{u}} = O(\mu). \tag{5.10b}$$

By using $t_{\mathcal{N}} = O(\mu^2)$ and applying the model (2.10) to the appropriate step in the procedure **condensed**, we obtain

$$\widehat{\Delta\lambda}_{\mathcal{N}} = (D_{\mathcal{N}} + G_{\mathcal{N}})^{-1} \left[g_{\mathcal{N}}(z) + f_{\mathcal{N}} - \Lambda_{\mathcal{N}}^{-1} t_{\mathcal{N}} + (\nabla g_{\mathcal{N}}(z) + F_{\mathcal{N}})^T \widehat{\Delta z} + \delta_{\mathbf{u}} \right] + \mu\delta_{\mathbf{u}}.$$

By comparing this expression with the corresponding exact formula, which is

$$\Delta\lambda_{\mathcal{N}} = D_{\mathcal{N}}^{-1} \left[g_{\mathcal{N}}(z) - \Lambda_{\mathcal{N}}^{-1} t_{\mathcal{N}} + \nabla g_{\mathcal{N}}(z)^T \Delta z \right],$$

and by using the bounds (5.10) and the fact that $f_{\mathcal{N}}$ and $F_{\mathcal{N}}$ have size $\delta_{\mathbf{u}}$, we obtain

$$\begin{aligned}
\widehat{\Delta\lambda}_{\mathcal{N}} - \Delta\lambda_{\mathcal{N}} &= \mu\delta_{\mathbf{u}} + (D_{\mathcal{N}} + G_{\mathcal{N}})^{-1}[f_{\mathcal{N}} + \delta_{\mathbf{u}}] \\
&\quad + \ [(D_{\mathcal{N}} + G_{\mathcal{N}})^{-1} - D_{\mathcal{N}}^{-1}][g_{\mathcal{N}}(z) - \Lambda_{\mathcal{N}}^{-1} t_{\mathcal{N}}] \\
&\quad + \ (D_{\mathcal{N}} + G_{\mathcal{N}})^{-1}(\nabla g_{\mathcal{N}}(z) + F_{\mathcal{N}})^T \widehat{\Delta z} - D_{\mathcal{N}}^{-1} \nabla g_{\mathcal{N}}(z)^T \Delta z \\
&= \mu\delta_{\mathbf{u}} + (D_{\mathcal{N}} + G_{\mathcal{N}})^{-1}[\nabla g_{\mathcal{N}}(z)^T(\widehat{\Delta z} - \Delta z) + F_{\mathcal{N}}^T \widehat{\Delta z}] \\
&\quad + \ [(D_{\mathcal{N}} + G_{\mathcal{N}})^{-1} - D_{\mathcal{N}}^{-1}] \nabla g_{\mathcal{N}}(z)^T \Delta z \\
&= \mu\delta_{\mathbf{u}}. \qquad (5.11)
\end{aligned}$$

Finally, for the recovered step $\widehat{\Delta s}$, we have from the last step of procedure **condensed**, together with (3.11b), (5.2d), (5.7b), and (2.10), that

$$\widehat{\Delta s} = -(g(z) + f + s) - (\nabla g(z) + F)^T \widehat{\Delta z} + \delta_{\mathbf{u}},$$

where the final term accounts for the rounding error (2.10) that arises from accumulating the terms in the sum, which are all bounded. By substituting the expression for the exact Δs together with the estimates (5.2d) and (5.7b) on the sizes of the perturbation terms, we obtain

$$\begin{aligned} \widehat{\Delta s} &= -(g(z) + s) - \nabla g(z)^T \Delta z - f - \nabla g(z)^T (\widehat{\Delta z} - \Delta z) - F^T \widehat{\Delta z} + \mu \delta_{\mathbf{u}} \\ &= \Delta s + \delta_{\mathbf{u}}. \end{aligned} \tag{5.12}$$

We summarize the results obtained so far in the following theorem.

THEOREM 5.1. *Suppose that Assumption* 4.1 *holds. Then when the step* $(\widehat{\Delta z}, \widehat{\Delta \lambda}, \widehat{\Delta s})$ *is calculated in a finite-precision environment by using the procedure* **condensed** *(and where, in particular, a backward stable method is used to solve the linear system for the* $\widehat{\Delta z}$ *component), we have that*

$$(\Delta z - \widehat{\Delta z}, U^T(\Delta \lambda_{\mathcal{B}} - \widehat{\Delta \lambda}_{\mathcal{B}}), \Delta s - \widehat{\Delta s}) = \delta_{\mathbf{u}}, \tag{5.13a}$$

$$V^T(\Delta \lambda_{\mathcal{B}} - \widehat{\Delta \lambda}_{\mathcal{B}}) = \delta_{\mathbf{u}}/\mu, \tag{5.13b}$$

$$\Delta \lambda_{\mathcal{N}} - \widehat{\Delta \lambda}_{\mathcal{N}} = \mu \delta_{\mathbf{u}}. \tag{5.13c}$$

This theorem extends the result of M. H. Wright [23] for accuracy of the computed solution of the condensed system by relaxing the LICQ assumption to MFCQ. When LICQ holds, the matrix V is vacuous, so the absolute error in all components is of size at most $\delta_{\mathbf{u}}$. The higher accuracy (5.13c) of the components $\widehat{\Delta \lambda}_{\mathcal{N}}$ (also noted in [23]) does not contribute significantly to the progress that can be made along the inexact direction $(\widehat{\Delta z}, \widehat{\Delta \lambda}, \widehat{\Delta s})$, in the sense of section 5.3.

We return briefly to the case discussed immediately after Corollary 4.4, in which the perturbations have the special form (4.42), using these results to show that the bound (5.13b) can be strengthened when $f_{\mathcal{B}}$ satisfies

$$V^T f_{\mathcal{B}} = O(\mu^2). \tag{5.14}$$

This case is of interest when the cancellation errors in computing $g_{\mathcal{B}}(z)$ are smaller than the estimate we made following (5.2d), possibly because of use of higher-precision arithmetic or the fact that the computation did not require differencing of quantities whose size is large relative to the final result. When (5.14) holds, we see by comparing (4.39) with (5.5) that

$$E_{23} = 0, \quad E_{33} = G_{\mathcal{N}} = \delta_{\mathbf{u}}/\mu, \quad f_2 = UU^T f_{\mathcal{B}} + O(\mu^2), \text{ where } f_{\mathcal{B}} = \delta_{\mathbf{u}}.$$

Therefore, we deduce from (4.44) that (5.6c) can be replaced by

$$V^T(\Delta \lambda_{\mathcal{B}} - y) = O(\mu).$$

Using (5.8) and $\mu \gg \delta_{\mathbf{u}}$, we can therefore replace (5.13b) in this case by

$$V^T(\Delta \lambda_{\mathcal{B}} - \widehat{\Delta \lambda}_{\mathcal{B}}) = O(\mu). \tag{5.15}$$

5.2. Termination of the Cholesky algorithm. In deriving the estimate (5.6), we have assumed that a backward stable algorithm is used to solve (5.1). Because of (2.6), (2.7), and the SC condition, and the estimates of the sizes of the diagonals of D (from (4.2) and Lemma 3.2), it is easy to show that the matrix in (5.1) is positive definite for all sufficiently small μ. The Cholesky algorithm is therefore an obvious candidate for solving this system. However, the condition number of the matrix in (5.1) usually approaches ∞ as $\mu \downarrow 0$, raising the possibility that the Cholesky algorithm may break down when μ is small. A simple argument, which we now sketch, suffices to show that successful completion of the Cholesky algorithm can be expected under the assumptions we have used in our analysis so far.

We state first the following technical result. Since it is similar to one proved by Debreu [6, Theorem 3], its proof is omitted.

LEMMA 5.2. *Suppose that M and A are two matrices with the properties that M is symmetric and*

$$A^T x = 0 \;\;\Rightarrow\;\; x^T M x \geq \alpha \|M\| \|x\|^2$$

for some constant $\alpha > 0$. Then for all μ such that

$$0 < \mu < \bar{\mu} \stackrel{\text{def}}{=} \min\left(\frac{\alpha\|A\|^2}{4\|M\|}, \frac{\|A\|}{\alpha\|M\|}\right),$$

we have that

$$x^T(M + \mu^{-1} A A^T)x \geq \frac{\alpha}{2}\|x\|^2 \quad \text{for all } x.$$

We apply this result to (5.1) by setting

$$\begin{aligned} M &= \mathcal{L}_{zz}(z,\lambda) + \nabla g_{\mathcal{N}}(z) D_{\mathcal{N}}^{-1} \nabla g_{\mathcal{N}}(z)^T = \mathcal{L}_{zz}(z,\lambda) + O(\mu), \\ A &= \mu^{1/2} \nabla g_{\mathcal{B}}(z) D_{\mathcal{B}}^{-1/2} \end{aligned}$$

(where again we use (4.2) and Lemma 3.2 to derive the order estimates). The conditions (2.6), (2.7), and strict complementarity ensure that this choice of M and A satisfies the assumptions of Lemma 5.2. The result then implies that the smallest singular value of the matrix in (5.1) is positive and of size $\Theta(1)$ for all values of μ below a threshold that is also of size $\Theta(1)$. Since $D = O(\mu^{-1})$, the largest eigenvalue of this matrix is of size $O(\mu^{-1})$, so we have

$$\text{cond}(\mathcal{L}_{zz}(z,\lambda) + \nabla g(z) D^{-1} \nabla g(z)^T) = O(\mu^{-1}). \tag{5.16}$$

(An estimate similar to this is derived by M. H. Wright [23, Theorem 3.2] under the LICQ assumption.) It is known from a result of Wilkinson (cited by Golub and Van Loan [12, p. 147]) that the Cholesky algorithm runs to completion if $q_n \delta_{\mathbf{u}} \text{cond}(\cdot) \leq 1$, where q_n is a modest quantity that depends polynomially on the dimension n of the matrix. By combining this result with (5.16), we conclude that for the matrix in (5.1), we can expect completion of the Cholesky algorithm whenever $\mu \gg \delta_{\mathbf{u}}$. That is, no new assumptions need to be added to those made in deriving the results of earlier sections.

We note that this situation differs a little from the case of linear programming where, because second-order conditions are not applicable, it is usually necessary to modify the Cholesky procedure to ensure that it runs to completion (see [30]).

5.3. Local convergence with computed steps. We begin this section by showing how the quantities r_f, r_g, and μ change along the computed step $(\widehat{\Delta z}, \widehat{\Delta\lambda}, \widehat{\Delta s})$ obtained from the finite-precision implementation of the procedure **condensed**. We compare these with the changes that can be expected along the exact direction $(\Delta z, \Delta\lambda, \Delta s)$. We then consider the effects of these perturbations on an algorithm of the type in which the iterates are expected to satisfy the conditions (3.11). Rapidly convergent variants of these algorithms for linear programming problems usually allow the values of C and γ in these conditions to be relaxed, so that a near-unit step can be taken. We address the following question: If similar relaxations are allowed in an algorithm for nonlinear programming, are near-unit steps still possible when the steps contain perturbations of the type considered above?

We show in particular that for the computed search direction, the maximum step length that can be taken without violating the nonnegativity conditions on λ and s satisfies

$$1 - \hat{\alpha}_{\max} = \delta_{\mathbf{u}}/\mu + O(\mu), \tag{5.17}$$

while the reductions in pairwise products, μ, r_f, and r_g, satisfy

$$(\lambda_i + \alpha\widehat{\Delta\lambda}_i)(s_i + \alpha\widehat{\Delta s}_i) = (1-\alpha)\lambda_i s_i + \delta_{\mathbf{u}} + O(\mu^2), \quad i = 1, 2, \ldots, m, \tag{5.18a}$$

$$\mu(\lambda + \alpha\widehat{\Delta\lambda}, s + \alpha\widehat{\Delta s}) = (1-\alpha)\mu + \delta_{\mathbf{u}} + O(\mu^2), \tag{5.18b}$$

$$r_f(z + \alpha\widehat{\Delta z}, \lambda + \alpha\widehat{\Delta\lambda}) = (1-\alpha)r_f + \delta_{\mathbf{u}} + O(\mu^2), \tag{5.18c}$$

$$r_g(z + \alpha\widehat{\Delta z}, s + \alpha\widehat{\Delta s}) = (1-\alpha)r_g + \delta_{\mathbf{u}} + O(\mu^2). \tag{5.18d}$$

The corresponding maximum steplength for the *exact* direction satisfies

$$1 - \alpha_{\max} = O(\mu), \tag{5.19}$$

while the reductions in r_f, r_g, and μ satisfy

$$(\lambda_i + \alpha\Delta\lambda_i)(s_i + \alpha\Delta s_i) = (1-\alpha)\lambda_i s_i + O(\mu^2), \quad i = 1, 2, \ldots, m, \tag{5.20a}$$

$$\mu(\lambda + \alpha\Delta\lambda, s + \alpha\Delta s) = (1-\alpha)\mu + O(\mu^2), \tag{5.20b}$$

$$r_f(z + \alpha\Delta z, \lambda + \alpha\Delta\lambda) = (1-\alpha)r_f + O(\mu^2), \tag{5.20c}$$

$$r_g(z + \alpha\Delta z, s + \alpha\Delta s) = (1-\alpha)r_g + O(\mu^2). \tag{5.20d}$$

Our proof of the estimates (5.17) and (5.18) is tedious but not completely straightforward, and we have included it in the appendix.

It is clear from (5.17) and (5.18) that the direction $(\widehat{\Delta z}, \widehat{\Delta\lambda}, \widehat{\Delta s})$ makes good progress toward the solution set $\mathcal{S}$. If the actual steplength α is close to its maximum value $\hat{\alpha}_{\max}$, in the sense that

$$\hat{\alpha}_{\max} - \alpha = \delta_{\mathbf{u}}/\mu + O(\mu), \tag{5.21}$$

we have by direct substitution in (5.17) and (5.18) that

$$\begin{aligned}
\mu(\lambda + \alpha\widehat{\Delta\lambda}, s + \alpha\widehat{\Delta s}) &= \delta_{\mathbf{u}} + O(\mu^2),\\
r_f(z + \alpha\widehat{\Delta z}, \lambda + \alpha\widehat{\Delta\lambda}) &= \delta_{\mathbf{u}} + O(\mu^2),\\
r_g(z + \alpha\widehat{\Delta z}, s + \alpha\widehat{\Delta s}) &= \delta_{\mathbf{u}} + O(\mu^2).
\end{aligned}$$

These formulae suggest that finite precision does not have an observable effect on the quadratic convergence rate of the underlying algorithm until μ drops below about $\sqrt{\mathbf{u}}$. Stopping criteria for interior-point methods usually include a condition such as $\mu \leq 10^4\mathbf{u}$ or $\mu \leq \sqrt{\mathbf{u}}$ (see, for example, [5]), so that μ is not allowed to become so small that the assumption $\mu \gg \mathbf{u}$ made in (4.7) is violated.

In making this back-of-the-envelope assessment, however, we have not taken into account the approximate centrality conditions (3.11), which must continue to hold (possibly in a relaxed form) at the new iterate. These conditions play a central role both in the analysis above and in the convergence analysis of the underlying "exact" algorithms, and also appear to be important in practice. Typically (see, for example, Ralph and Wright [21]), the conditions (3.11) are relaxed by allowing a modest increase in C and a modest decrease in γ on the rapidly convergent steps. We show in the next result that enforcement of these relaxed conditions is not inconsistent with taking a step length α that is close to $\hat{\alpha}_{\max}$, so that rapid convergence can still be observed even in the presence of finite-precision effects.

THEOREM 5.3. *Suppose Assumption* 4.1 *holds. Then when the step* $(\widehat{\Delta z}, \widehat{\Delta\lambda}, \widehat{\Delta s})$ *is calculated in a finite-precision environment by using the procedure* **condensed**, *there is a constant* $\hat{C}$ *such that for all* $\tau \in [0, 1/2]$ *and all* α *satisfying*

$$\alpha \in [0, 1 - \hat{C}\tau^{-1}(\mathbf{u}/\mu + \mu)], \tag{5.22}$$

the following relaxed form of the approximate centrality conditions holds:

$$r_f(z + \alpha\widehat{\Delta z}, \lambda + \alpha\widehat{\Delta\lambda}) \leq C(1+\tau)\mu(\lambda + \alpha\widehat{\Delta\lambda}, s + \alpha\widehat{\Delta s}), \tag{5.23a}$$

$$r_g(z + \alpha\widehat{\Delta z}, s + \alpha\widehat{\Delta s}) \leq C(1+\tau)\mu(\lambda + \alpha\widehat{\Delta\lambda}, s + \alpha\widehat{\Delta s}), \tag{5.23b}$$

$$(\lambda_i + \alpha\widehat{\Delta\lambda}_i)(s_i + \alpha\widehat{\Delta s}_i) \geq \gamma(1-\tau)\mu(\lambda + \alpha\widehat{\Delta\lambda}, s + \alpha\widehat{\Delta s}) \quad \text{for all } i = 1, 2, \ldots, m, \tag{5.23c}$$

where C *is the constant from conditions* (3.11). *Moreover, when we set* α *to its upper bound in* (5.22), *we find that*

$$\delta(z + \alpha\widehat{\Delta z}, \lambda + \alpha\widehat{\Delta\lambda}) \leq \tau^{-1}(\delta_{\mathbf{u}} + O(\mu^2)). \tag{5.24}$$

Proof. From (3.11) and (5.18), we have that

$$\begin{aligned}
&\|r_f(z + \alpha\widehat{\Delta z}, \lambda + \alpha\widehat{\Delta\lambda})\| \\
&= (1-\alpha)\|r_f\| + \delta_{\mathbf{u}} + O(\mu^2) \\
&\leq C(1-\alpha)\mu + \delta_{\mathbf{u}} + O(\mu^2) \\
&= C(1+\tau)(1-\alpha)\mu - C\tau(1-\alpha)\mu + \delta_{\mathbf{u}} + O(\mu^2) \\
&= C(1+\tau)\mu(\lambda + \alpha\widehat{\Delta\lambda}, s + \alpha\widehat{\Delta s}) - C\tau(1-\alpha)\mu + \delta_{\mathbf{u}} + O(\mu^2).
\end{aligned}$$

We deduce that the required condition (5.23a) will hold provided that

$$\delta_{\mathbf{u}} + O(\mu^2) \leq C\tau(1-\alpha)\mu.$$

Since by definition we have that $\delta_{\mathbf{u}} + O(\mu^2) \leq \bar{C}(\mathbf{u} + \mu^2)$ for some positive constant $\bar{C}$, we find that a sufficient condition for the required inequality is that

$$(1-\alpha) \geq (\bar{C}/C)\tau^{-1}(\mathbf{u}/\mu + \mu),$$

which is equivalent to (5.22) for an obvious definition of $\hat{C}$. Identical logic can be applied to $\|r_g\|$ to yield a similar condition on α.

For the condition (5.23c), we have from (3.11) and (5.18) that

$$\begin{aligned}
&(\lambda_i + \alpha\widehat{\Delta\lambda}_i)(s_i + \alpha\widehat{\Delta s}_i) \\
&= (1-\alpha)\lambda_i s_i + \delta_{\mathbf{u}} + O(\mu^2) \\
&\geq (1-\alpha)\gamma\mu + \delta_{\mathbf{u}} + O(\mu^2) \\
&= \gamma(1-\tau)(1-\alpha)\mu + \gamma\tau(1-\alpha)\mu + \delta_{\mathbf{u}} + O(\mu^2) \\
&= \gamma(1-\tau)\mu(\lambda + \alpha\widehat{\Delta\lambda}, s + \alpha\widehat{\Delta s}) + \gamma\tau(1-\alpha)\mu + \delta_{\mathbf{u}} + O(\mu^2).
\end{aligned}$$

Hence, the condition (5.23c) holds provided that

$$\gamma\tau(1-\alpha)\mu + \delta_{\mathbf{u}} + O(\mu^2) \geq 0.$$

Similar logic can be applied to this inequality to derive a bound of the type (5.22), after a possible adjustment of $\hat{C}$.

Finally, we obtain (5.24) by substituting $\alpha = 1 - \hat{C}\tau^{-1}(\mathbf{u}/\mu + \mu)$ into (5.18) and applying Theorem 3.3. (Note that, despite the relaxation of the centrality conditions (5.23), the result of Theorem 3.3 still holds; we simply modify the proof to replace C by $(3/2)C$ in (3.11a) and (3.11b) and γ by $\gamma/2$ in (3.11c).) □

6. The augmented system. In this section, we consider the case in which the augmented system (3.9) (equivalently, (4.1)) is solved to obtain $(\Delta z, \Delta\lambda)$, while the remaining step component Δs is recovered from (3.8). The formal specification for this procedure is as follows.

procedure augmented
given the current iterate (z, λ, s)
form the coefficient matrix and right-hand side for (4.1);
solve (4.1) to obtain $(\Delta z, \Delta\lambda)$;
set $\Delta s = -(g(z) + s) - \nabla g(z)^T \Delta z$.

Much of our work in analyzing the augmented system form (4.1) has already been performed in section 4; the main error result is simply Corollary 4.4. However, we can apply this result only if the floating-point errors made in evaluating and solving this system satisfy the assumptions of this corollary. In particular, we need to show that the perturbation matrices E_{ij}, $i, j = 1, 2, 3$, in (4.39) satisfy the estimates (4.33).

This task is not completely straightforward. Unlike the condensed and full-system cases, it is not simply a matter of assuming that a backward-stable algorithm has been used to solve the system (4.1). The reason is that the largest terms in the coefficient matrix in (3.9)—the diagonal elements in the matrix $D_{\mathcal{N}}$—have size $O(\mu^{-1})$. The usual analysis of backward-stable algorithms represents the floating-point errors as a perturbation of the entire coefficient matrix whose size is bounded by $\delta_{\mathbf{u}}$ times the matrix norm—in this case, $\delta_{\mathbf{u}}/\mu$. However, Corollary 4.4 requires some elements of the perturbation matrix to be smaller than this estimate; in particular, the submatrices E_{12}, E_{21}, and E_{22} need to be of size $\delta_{\mathbf{u}}$. Therefore, we need to look closely at the particular algorithms used to solve (4.1) to see whether they satisfy the following condition.

Condition 6.1. The solution obtained by applying the algorithm in question to the system (4.1) in floating-point arithmetic is the exact solution of a perturbed system

in which the perturbations of the coefficient matrix satisfy the estimates (4.33), while the right-hand side is unperturbed.

We focus on diagonal pivoting methods, which take a symmetric matrix T and produce a factorization of the form

$$PTP^T = LYL^T, \tag{6.1}$$

where P is a permutation matrix, L is unit lower triangular, and Y is block diagonal, with a combination of 1×1 and symmetric 2×2 blocks. The best-known methods of this class are due to Bunch and Parlett [3] and Bunch and Kaufman [2], while Duff et al. [7] and Fourer and Mehrotra [10] have described sparse variants. These algorithms differ in their selection criteria for the 1×1 and 2×2 pivot blocks. In our case, the presence of the diagonal elements of size $\Theta(\mu^{-1})$ (from the submatrix $D_{\mathcal{N}} = \Lambda_{\mathcal{N}}^{-1} S_{\mathcal{N}}$) and their place in these pivot blocks are crucial to the result.

We start by stating a general result of Higham [17] concerning backward stability that applies to all diagonal pivoting schemes. We then examine the Bunch–Kaufman scheme, showing that the large diagonals appear only as 1×1 pivots and that this algorithm satisfies Condition 6.1. (In [17, Theorem 4.2], Higham actually proves that the Bunch–Kaufman scheme is backward stable in the normwise sense, but this result is not applicable to our context, for the reasons mentioned above.)

Next, we briefly examine the Bunch–Parlett method, showing that it starts out by selecting all the large diagonal elements in turn as 1×1 pivots, before going on to factor the remaining matrix, whose elements are all $O(1)$ in size. This method also satisfies Condition 6.1. We then examine the sparse diagonal pivoting approaches of Duff et al. [7] and Fourer and Mehrotra [10], which may not satisfy Condition 6.1 because of the possible presence of 2×2 pivots in which one of the diagonals has size $\Theta(\mu^{-1})$. These algorithms can be modified in simple ways to overcome this difficulty, possibly at the expense of higher density in the L factor. We then mention Gaussian elimination with pivoting and refer to previous results in the literature to show that this approach satisfies Condition 6.1. Finally, we state a result like Theorem 5.3 about convergence of a finite-precision implementation of an algorithm based on the augmented system form.

Higham [17, Theorem 4.1] proves the following result.

THEOREM 6.1. *Let T be an $\bar{n} \times \bar{n}$ symmetric matrix, and let $\hat{x}$ be the computed solution to the linear system $Tx = b$ produced by a method that yields a factorization of the form* (6.1)*, with any diagonal pivoting strategy. Assume that, during recovery of the solution, the subsystems that involve the 2×2 diagonal blocks are solved via Gaussian elimination with partial pivoting. Then we have that*

$$(T + \Delta T)\hat{x} = b, \quad |\Delta T| \leq \delta_{\mathbf{u}}(|T| + P^T|\hat{L}||\hat{Y}||\hat{L}^T|P) + \delta_{\mathbf{u}}^2, \tag{6.2}$$

where $\hat{L}$ and $\hat{Y}$ are the computed factors, and $|A|$ denotes the matrix formed from A by replacing all its elements by their absolute values.

In Higham's result, the coefficient of $\mathbf{u}$ in the $\delta_{\mathbf{u}}$ term is actually a linear polynomial in the dimension of the system. The partial pivoting strategy for the 2×2 systems can actually be replaced by any method for which the computed solution of $Ry = d$ satisfies $(R + \Delta R)\hat{y} = d$, where R is the 2×2 matrix in question and $|\Delta R| \leq \delta_{\mathbf{u}}|R|$. This property was also key in an earlier paper of S. Wright [27], who derived a result similar to Theorem 6.1 in the context of the augmented systems that arise from interior-point methods for linear programming.

All the procedures below have the property that the growth in the maximum element size in the remaining submatrix is bounded by a modest quantity at each individual step of the factorization. (In the case of Bunch–Kaufman and Bunch–Parlett, this bound averages about 2.6 per elimination step; see Golub and Van Loan [12, section 4.4.4].) As with Gaussian elimination with partial pivoting, exponential element growth is possible, so that L and Y in (6.1) contain much larger elements than the original matrix T. Such behavior is, however, quite rare and is confined to pathological cases and certain special problem classes. In our analysis below, we make the safe assumption that catastrophic growth of this kind does not occur.

6.1. The Bunch–Kaufman procedure. At each iteration, the Bunch–Kaufman procedure chooses either a 1×1 or a 2×2 pivot by examining at most two columns of the remaining matrix, that is, the part of the matrix that remains to be factored at this stage of the process. It makes use of quantities χ_i defined by

$$\chi_i = \max_{j \,|\, j \neq i} |T_{ij}|,$$

where in this case T denotes the remaining matrix. We define the pivot selection strategy for the first step of the factorization process. The entire algorithm is obtained by applying this procedure recursively to the remaining submatrix.

set $\nu = (1 + \sqrt{17})/8$;
calculate χ_1, and store the index r for which $\chi_1 = |T_{r1}|$;
if $|T_{11}| \geq \nu \chi_1$
    choose T_{11} as a 1×1 pivot;
else
    calculate χ_r;
    if $\chi_r |T_{11}| \geq \nu \chi_1^2$
        choose T_{11} as a 1×1 pivot;
    else if $|T_{rr}| \geq \nu \chi_r$
        choose T_{rr} as a 1×1 pivot;
    else
        choose a 2×2 pivot with diagonals T_{11} and T_{rr};
    end if
end if.

For each choice of pivot, the permutation matrix P_1 is chosen so that the desired 1×1 or 2×2 pivot is in the upper left of the matrix $P_1 T P_1^T$. If one writes

$$P_1 T P_1^T = \begin{bmatrix} R & C^T \\ C & \hat{T} \end{bmatrix},$$

where R is the chosen pivot, the first step of the factorization yields

$$P_1 T P_1^T = \begin{bmatrix} I & \\ CR^{-1} & I \end{bmatrix} \begin{bmatrix} R & \\ & \bar{T} \end{bmatrix} \begin{bmatrix} I & R^{-1}C^T \\ & I \end{bmatrix}, \tag{6.3}$$

where $\bar{T} = \hat{T} - CR^{-1}C^T$ is the matrix remaining after this stage of the factorization.

At the first step of the factorization, the quantities χ_1 and χ_r (if calculated) both have size $O(1)$, since the large elements of this matrix occur only on the diagonal. Since a 2×2 pivot is chosen only if

$$|T_{11}| < \nu \chi_1 \quad \text{and} \quad |T_{rr}| < \nu \chi_r,$$

it follows immediately that both diagonals in a 2×2 pivot must be $O(1)$. Hence, the pivot chosen by this procedure is one of three types:

(6.4a) 1×1 pivot of size $O(1)$;

(6.4b) 2×2 pivot in which both diagonals have size $O(1)$;

(6.4c) 1×1 pivot of size $\Theta(\mu^{-1})$.

In fact, the pivots are one of the types (6.4) at *all* stages of the factorization, not just the first stage. The reason is that the updated matrix $\bar{T}$ in (6.3) has the same essential form as the original matrix T—its elements are all of size $O(1)$ except for some large diagonal elements of size $\Theta(\mu^{-1})$. We demonstrate this claim by showing that the update $CR^{-1}C^T$ that is applied to the remaining matrix in (6.3) is a matrix whose elements are of size at most $O(1)$, regardless of the type of pivot, so that it does not disturb the essential structure of the remaining matrix. When the pivots are of type (6.4a) and (6.4b), the standard argument of Bunch and Kaufman [2] can be applied to show that the norm of $CR^{-1}C^T$ is at most a modest multiple of $\|C\|$. We know that $\|C\| = O(1)$, since C consists only of off-diagonal elements, so we conclude that $\|CR^{-1}C^T\| = O(1)$ in this case as claimed. For the other pivot type (6.4c), we have $R = \Theta(\mu^{-1})$ and $C = O(1)$, so the elements of $CR^{-1}C^T$ have size $O(\mu)$, and the claim holds in this case, too.

In the rest of this subsection, we show by using Theorem 6.1 that Condition 6.1 holds for the Bunch–Kaufman algorithm. In fact, we prove a stronger result: When T in Theorem 6.1 is the matrix (4.1), the perturbation matrix ΔT contains elements of size $\delta_{\mathbf{u}}$, except in those diagonal locations corresponding to the elements of $D_{\mathcal{N}}$, where they may be as large as $\delta_{\mathbf{u}}/\mu$. Given the bound on $|\Delta T|$ in (6.2), we need only to show that $P^T|\hat{L}||\hat{Y}||\hat{L}|^T P$ has the desired structure. In fact, it suffices to show that the exact factor product $P^T|L||Y||L|^T P$ has the structure in question, since the difference between these two products is just $\delta_{\mathbf{u}}$ in size.

We demonstrate this claim inductively, using a refined version of the arguments from Higham [17, section 4.3] for some key points and omitting some details. For simplicity, and without loss of generality, we assume that $P = I$.

When $\bar{n} = 1$ (that is, T is 1×1), we have that $L = 1$ and $Y = T$, and the result holds trivially. When $\bar{n} = 2$, there are three cases to consider. If the matrix contains no elements of size $\Theta(\mu^{-1})$, then the analysis for general matrices can be used to show that $|L||Y||L|^T = O(1)$, as required. If either or both diagonal elements have size $\Theta(\mu^{-1})$, then both pivots are 1×1, and the factors have the form

$$L = \begin{bmatrix} 1 & 0 \\ T_{21}/T_{11} & 1 \end{bmatrix}, \qquad Y = \begin{bmatrix} T_{11} & 0 \\ 0 & T_{22} - T_{21}^2/T_{11} \end{bmatrix}. \tag{6.5}$$

Two cases arise.

(i) A diagonal of size $O(1)$ is accepted as the first pivot and moved (if necessary) to the $(1,1)$ position. We then have

$$|T_{11}| \geq \nu\chi_1 = \nu\chi_2 = \nu|T_{21}|,$$

and therefore $|T_{21}/T_{11}| \leq 1/\nu$ and hence $|T_{21}^2/T_{11}| \leq |T_{21}|/\nu = O(1)$. If the $(2,2)$ diagonal is also $O(1)$, we have that $L = O(1)$ and $Y = O(1)$, and we are done. Otherwise, $T_{22} = \Theta(\mu^{-1})$, and so the $(2,2)$ element of Y satisfies this same estimate. We conclude from (6.5) that $|L||Y||L|^T$ also has an $\Theta(\mu^{-1})$ element in the $(2,2)$ position and $O(1)$ elements elsewhere.

(ii) A diagonal of size $\Theta(\mu^{-1})$ is accepted as the first pivot and moved (if necessary) to the $(1,1)$ position. We then have

$$|T_{21}/T_{11}| = O(\mu), \qquad |T_{21}^2/T_{11}| = O(\mu).$$

It follows from (6.5) that

$$|L||Y||L|^T = \begin{bmatrix} |T_{11}| & |T_{21}| \\ |T_{21}| & |T_{22}| + O(\mu) \end{bmatrix},$$

which obviously has the desired structure.

We now assume that our claim holds for some dimension $\bar{n} \geq 2$, and we prove that it continues to hold for dimension $\bar{n}+1$. Using the notation of (6.3) (assuming that $P_1 = I$), and denoting the factorization of the Schur complement $\bar{T}$ in (6.3) by $\bar{T} = \bar{L}\bar{Y}\bar{L}^T$, we have that

$$T = LYL^T = \begin{bmatrix} I & \\ CR^{-1} & \bar{L} \end{bmatrix} \begin{bmatrix} R & \\ & \bar{Y} \end{bmatrix} \begin{bmatrix} I & R^{-1}C^T \\ & \bar{L}^T \end{bmatrix}. \tag{6.6}$$

It follows that

$$|L||Y||L|^T = \begin{bmatrix} |R| & |R||R^{-1}C^T| \\ |CR^{-1}||R| & |CR^{-1}||R||R^{-1}C^T| + |\bar{L}||\bar{Y}||\bar{L}|^T \end{bmatrix}. \tag{6.7}$$

Since, as we mentioned above, the norm of $CR^{-1}C^T$ is at most $O(1)$, the Schur complement $\bar{T} = \hat{T} - CR^{-1}C^T$ has size $O(1)$ except for large $\Theta(\mu^{-1})$ elements in the same locations as in the original matrix. Hence, by our inductive hypothesis, $|\bar{L}||\bar{Y}||\bar{L}|^T$ has a similar structure, and we need to show only that the effects of the first step of the factorization (6.3) do not disturb the desired structure.

For the case in which R is a pivot of type either (6.4a) and (6.4b), Higham [17, section 4.3] shows all elements of both $|CR^{-1}||R|$ and $|CR^{-1}||R||R^{-1}C^T|$ are bounded by a modest multiple of either χ_1 (if T_{11} was selected as the pivot because it passed the test $|T_{11}| \geq \nu\chi_1$) or $(\chi_1 + \chi_r)$, where r is the "other" column considered during the selection process. In our case, this observation implies that both $|CR^{-1}||R|$ and $|CR^{-1}||R||R^{-1}C^T|$ have size $O(1)$. By combining these observations with those of the preceding paragraph, we conclude that for pivots of types (6.4a) and (6.4b), "large" elements of the matrix in (6.7) occur only in the diagonal locations originally occupied by $D_{\mathcal{N}}$.

For the remaining case—pivots of type (6.4c)—we have that C has size $O(1)$ while R^{-1} has size $O(\mu)$. Therefore, $|CR^{-1}||R|$ has size $O(1)$ and $|CR^{-1}||R||R^{-1}C^T|$ has size $O(\mu)$, while $|R|$, which occupies the $(1,1)$ position in the matrix (6.7), just as it did in the original matrix T, has size $\Theta(\mu^{-1})$. We conclude that the desired structure holds in this case as well.

We conclude from this discussion that Condition 6.1 holds for the Bunch–Kaufman procedure. We show later that the perturbations arising from other sources, namely, roundoff and cancellation in the evaluation of the matrix and right-hand side, also satisfy the conditions of Corollary 4.4, so this result can be used to bound the error in the computed steps.

Finally, we note that it is quite possible for pivots of types (6.4a) and (6.4b) to be chosen while diagonal elements of size $\Theta(\mu^{-1})$ still remain in the submatrix. Therefore, a key assumption of the analysis of Forsgren, Gill, and Shinnerl [9, Theorem 4.4]—namely, that all the diagonals of size $\Theta(\mu^{-1})$ are chosen as 1×1 pivots before any of the other diagonals are chosen—may not be satisfied by the Bunch–Kaufman procedure.

6.2. The Bunch–Parlett procedure. The Bunch–Parlett procedure is conceptually simpler but more expensive to implement than Bunch–Kaufman, since it requires $O(n^2)$ (rather than $O(n)$) comparisons at each step of the factorization. The pivot selection strategy is as follows.

set $\nu = (1+\sqrt{17})/8$;
calculate $\chi_{\text{off}} = |T_{rs}| = \max_{i \neq j} |T_{ij}|$, $\chi_{\text{diag}} = |T_{pp}| = \max_i |T_{ii}|$;
if $\chi_{\text{diag}} \geq \nu \chi_{\text{off}}$
    choose T_{pp} as the 1×1 pivot;
else
    choose the 2×2 pivot whose off-diagonal element is T_{rs};
end if.

The elimination procedure then follows as in (6.3).

It is easy to show that the Bunch–Parlett procedure starts by selecting all the diagonals of size $\Theta(\mu^{-1})$ in turn as 1×1 pivots. (Because of this property, it satisfies the key assumption of [9] mentioned at the end of the preceding section.) The update $CR^{-1}C^T$ generated by each of these pivot steps has size only $O(\mu)$, so the matrix that remains after this phase of the factorization contains only $O(1)$ elements. The remaining pivots are then a combination of types (6.4a) and (6.4b).

By using the arguments of the preceding subsection in a slightly simplified form, we can show that Condition 6.1 holds for this procedure as well.

6.3. Sparse diagonal pivoting. For large instances of (1.1), the Bunch–Kaufman and Bunch–Parlett procedures are usually inefficient because they do not try to maintain sparsity in the lower triangular factor L. Sparse variants of these algorithms, such as those of Duff et al. [7] and Fourer and Mehrotra [10], use pivot selection strategies that combine stability considerations with Markowitz-like estimates of the amount of fill-in that a candidate pivot will cause in the remaining matrix.

At each stage of the factorization, both algorithms examine a roster of possible 1×1 and 2×2 pivots, starting with those that would create the least fill-in. As soon as a pivot is found that meets the stability criteria described below, it is accepted. Both algorithms prefer to use 1×1 pivots where possible.

For candidate 1×1 pivots, Duff et al. [7, p. 190] use the following stability criterion:

$$|R^{-1}| \|C\|_\infty \leq \rho, \tag{6.8}$$

where the notation R and C is from (6.3) and $\rho \in [2, \infty)$ is some user-selected parameter that represents the tolerable growth factor at each stage of the factorization. For a 2×2 pivot, the criterion is

$$|R^{-1}| \begin{bmatrix} \|C_{\cdot,1}\|_\infty \\ \|C_{\cdot,2}\|_\infty \end{bmatrix} \leq \begin{bmatrix} \rho \\ \rho \end{bmatrix}, \tag{6.9}$$

where $C_{\cdot,1}$ and $C_{\cdot,2}$ are the two columns of C. The stability criteria of Fourer and Mehrotra [10] are similar.

As they stand, the stability tests (6.8) and (6.9) do not necessarily restrict the choice of pivots to the three types (6.4). If a 1×1 pivot of size $\Theta(\mu^{-1})$ is ever considered for structural reasons, it will pass the test (6.8) (the left-hand side of this expression will have size $O(\mu)$) and therefore will be accepted as a pivot. However, it is possible that 2×2 pivots in which one or both diagonals have size $\Theta(\mu^{-1})$ may pass the test (6.9) and may therefore be accepted. Although the test (6.9) ensures that the size of the update $CR^{-1}C^T$ is modest (so that the update $\bar{T} = \hat{T} - CR^{-1}C^T$ does

not disturb the large-diagonal structure of $\hat{T}$), there is no obvious assurance that the matrix $|L||Y||L|^T$ in (6.7) mirrors the structure of $|T|$, in terms of having the large diagonal elements in the same locations. The terms $|CR^{-1}||R|$ and $|CR^{-1}||R||R^{-1}C^T|$ in (6.7) may not have size $O(1)$, as they do for pivots of the three types (6.4) arising from the Bunch–Kaufman and Bunch–Parlett selection procedures.

The Fourer–Mehrotra algorithm does, however, rule out the possibility of a 2×2 pivot in which *both* diagonals are of size $\Theta(\mu^{-1})$. It considers a 2×2 candidate only if one of its diagonal elements has previously been considered as a 1×1 pivot but failed the stability test. However, if either of the diagonals had been subjected to the test (6.8), they would have been accepted, as noted in the preceding paragraph, so this situation cannot occur.

If the sparse algorithms are modified to ensure that all pivots have one of the three types (6.4), and all continue to satisfy the stability tests (6.8) or (6.9), then simple arguments (simpler than those of section 6.1!) can be applied to show that Condition 6.1 is satisfied. One possible modification that achieves the desired effect is to require that a 2×2 pivot be allowed only if *both* its diagonals have previously been considered as 1×1 pivots but failed the stability test (6.8).

6.4. Gaussian elimination. Another possibility for solving the system (4.1) is to ignore its symmetry and apply a Gaussian elimination algorithm, with row and/or column pivoting to preserve sparsity and prevent excessive element growth. Such a strategy satisfies Condition 6.1. In [24], the author uses a result of Higham [16] to show that the effects of the large diagonal elements are essentially confined to the columns in which they appear. Assuming that the pivot sequence is chosen to prevent excessive element growth in the remaining matrix, and using the notation of (4.32) and (4.33), we can account for the effects of roundoff error in Gaussian elimination with perturbations in the coefficient matrix that satisfy the following estimates:

$$E_{11}, E_{21}, E_{31}, E_{12}, E_{22}, E_{32} = \delta_{\mathbf{u}}, \quad E_{13}, E_{23}, E_{33} = \delta_{\mathbf{u}}/\mu.$$

These certainly satisfy the conditions (4.33), so Condition 6.1 holds.

6.5. Local convergence with the computed steps. We can now state a formal result to show that when the evaluation errors are taken into account as well as the roundoff errors from the factorization/solution procedure discussed above, the accuracies of the computed steps obtained from the procedure **augmented**, implemented in finite precision, satisfy the same estimates as for the corresponding steps obtained from the procedure **condensed**. The result is analogous to Theorem 5.1.

THEOREM 6.2. *Suppose Assumption* 4.1 *holds. Then when the step* $(\widehat{\Delta z}, \widehat{\Delta\lambda}, \widehat{\Delta s})$ *is calculated in a finite-precision environment by using the procedure* **augmented**, *where the algorithm used to solve* (4.1) *satisfies Condition* 6.1, *we have*

$$(\Delta z - \widehat{\Delta z}, U^T(\Delta\lambda_{\mathcal{B}} - \widehat{\Delta\lambda}_{\mathcal{B}}), \Delta s - \widehat{\Delta s}) = \delta_{\mathbf{u}}, \tag{6.10a}$$

$$V^T(\Delta\lambda_{\mathcal{B}} - \widehat{\Delta\lambda}_{\mathcal{B}}) = \delta_{\mathbf{u}}/\mu, \tag{6.10b}$$

$$\Delta\lambda_{\mathcal{N}} - \widehat{\Delta\lambda}_{\mathcal{N}} = \delta_{\mathbf{u}}. \tag{6.10c}$$

Proof. The proof follows from Corollary 4.4 when we show that the perturbations to (4.1) from all sources—evaluation of the matrix and right-hand side as well as the factorization/solution procedure—satisfy the bounds required by this earlier result.

Because of Condition 6.1, perturbations arising from the factorization/solution procedure satisfy the bounds (4.33). The expressions (5.2) show that the errors arising from evaluation of $\mathcal{L}_{zz}(z,\lambda)$, $\mathcal{L}_z(z,\lambda)$, $\nabla g(z)$, and $g(z)$ are all of size $\delta_{\mathbf{u}}$, and hence they too satisfy the required bounds. Similarly to (5.3), evaluation of $D_{\mathcal{B}}$ and $D_{\mathcal{N}}$ yields errors of relative size $\delta_{\mathbf{u}}$, that is,

$$\text{computed } D_{\mathcal{B}} \leftarrow D_{\mathcal{B}} + G_{\mathcal{B}}, \quad G_{\mathcal{B}} = \mu\delta_{\mathbf{u}}, \tag{6.11a}$$

$$\text{computed } D_{\mathcal{N}} \leftarrow D_{\mathcal{N}} + G_{\mathcal{N}}, \quad G_{\mathcal{N}} = \delta_{\mathbf{u}}/\mu, \tag{6.11b}$$

where $G_{\mathcal{B}}$ and $G_{\mathcal{N}}$ are diagonal matrices.

We now obtain all the estimates in (6.10) by a direct application of Corollary 4.4, with the exception of the estimate for $(\Delta s - \widehat{\Delta s})$. Since the expressions for recovering Δs are identical in procedures **condensed** and **augmented**, we can apply expression (5.12) from section 5.1 to deduce that the desired estimate holds for this component as well. □

The only difference between the error estimates of Theorem 5.1 for the condensed system and those obtained above for the augmented system is that the $\widehat{\Delta\lambda}_{\mathcal{N}}$ components are slightly less accurate in the augmented case. If we work through the analysis of section 5.3 with the estimate (6.10c) replacing (5.13c), we find that the main results are unaffected. Therefore, we conclude this section by stating without proof a result similar to Theorem 5.3.

THEOREM 6.3. *Suppose that all the assumptions of Theorem* 5.3 *hold, except that the step* $(\widehat{\Delta z}, \widehat{\Delta\lambda}, \widehat{\Delta s})$ *is calculated by using the procedure* **augmented** *with a factorization/solution algorithm that satisfies Condition* 6.1. *Then the conclusions of Theorem* 5.3 *hold.*

7. Numerical illustration. We illustrate the results of sections 5 and 6 using the two-variable example (2.8). Consider a simple algorithm that takes steps satisfying (3.8) with t set rather arbitrarily to $t = \mu^2 e$. (The search directions thus used are like those generated in the later stages of a practical primal-dual algorithm such as Mehrotra's algorithm [19].) We start this algorithm from the point

$$z_0 = (1/30, 1/9)^T, \quad \lambda_0 = (1, 1/5)^T, \quad s_0 = (1/10, 1/2).$$

(It is easy to check that the conditions (3.11) are satisfied at this point for a modest value of C.) At each step we calculated $\hat{\alpha}_{\max}$, defined in section 5.3, and took an actual step of $.99\hat{\alpha}_{\max}$.

We programmed the method in Matlab, using double-precision arithmetic. In our first experiment, we solved the formulation (4.1) of the linear equations using Matlab's standard Gaussian elimination solver for general systems of linear equations, which was analyzed in section 6.4. From Theorem 6.2, the estimates (6.10) apply to this case.

Results are tabulated in Table 1. Note first the size of the component $\|V^T\widehat{\Delta\lambda}_{\mathcal{B}}\|$, which grows as μ decreases below $\mathbf{u}^{1/2}$, in accordance with (6.10b). (We cannot tabulate the difference $\|V^T(\widehat{\Delta\lambda}_{\mathcal{B}} - \Delta\lambda_{\mathcal{B}}\|)$ because of course we do not know the true step $(\Delta z, \Delta\lambda, \Delta s)$, but since the true step has size $O(\mu)$ (Corollary 4.3), the error is dominated by the term $V^T\widehat{\Delta\lambda}_{\mathcal{B}}$ in any case.) As predicted by (5.17), the maximum step $\hat{\alpha}_{\max}$ becomes significantly smaller than 1 as μ is decreased below $\mathbf{u}^{1/2}$. As indicated by (5.18), however, good progress still can be made along this direction (in the sense of reducing μ and the norms of the residuals r_f and r_g) almost until μ

TABLE 1
Details of iteration sequence for PDIP applied to (2.8)*, with steps computed by solving the augmented system.*

iter	$\log \mu$	$\log \|\widehat{\Delta z}\|$	$\log \|U^T \widehat{\Delta\lambda}_{\mathcal{B}}\|$	$\log \|V^T \widehat{\Delta\lambda}_{\mathcal{B}}\|$	$\hat{\alpha}_{\max}$	λ^T
0	-1.0	-0.9	-1.9	-1.9	.9227	(1.00,.20)
1	-2.7	-1.5	-1.3	-1.2	.9193	(0.99,.19)
⋮						
5	-9.4	-6.7	-6.3	-4.6	1.0	(1.04,.23)
6	-11.4	-8.7	-8.3	-5.9	1.0	(1.04, .23)
7	-13.4	-10.7	-10.3	-3.8	.9999	(1.04,.23)
8	-15.4	-12.7	-12.3	-1.2	.9439	(1.04,.23)
9	-17.1	-13.9	-13.4	-0.6	.9723	(1.10,.20)

TABLE 2
Details of iteration sequence for PDIP applied to (2.8)*, with steps computed by solving the condensed system.*

iter	$\log \mu$	$\log \|\widehat{\Delta z}\|$	$\log \|U^T \widehat{\Delta\lambda}_{\mathcal{B}}\|$	$\log \|V^T \widehat{\Delta\lambda}_{\mathcal{B}}\|$	$\hat{\alpha}_{\max}$	λ^T
0	-1.0	-0.9	-1.9	-1.9	.9227	(1.00,.20)
1	-2.7	-1.5	-1.3	-1.2	.9193	(0.99,.19)
⋮						
5	-9.4	-6.7	-6.3	-4.6	1.0	(1.04,.23)
6	-11.4	-8.7	-8.3	-5.7	1.0	(1.04, .23)
7	-13.4	-10.7	-10.3	-8.3	1.0	(1.04,.23)
8	-15.4	-12.7	-12.4	-10.3	1.0	(1.04,.23)
9	-17.4	-14.7	-13.3	-12.3	1.0	(1.04,.23)

reaches the level of **u**. In fact, between iterations 5 and 8 we see the reduction factor of 100 that we would expect by moving a distance of .99 along a direction that is close to the pure Newton direction. The component with the large error—$V^T \widehat{\Delta\lambda}_{\mathcal{B}}$—does not interfere significantly with rapid convergence, but only causes the λ iterates to move tangentially to $\mathcal{S}_\lambda$. This effect may be noted in the final iterate where the value of λ changes significantly. In some cases, however, when the current λ is near the edge of the set $\mathcal{S}_\lambda$, this error may result in a severe curtailment of the step length.

Next, we performed the same experiment using the condensed formulation (3.10) of the linear system, as described in section 5. Results are shown in Table 2. The main difference with Table 1 is that Table 2 shows no increase in the value $\|V^T \widehat{\Delta\lambda}_{\mathcal{B}}\|$ as μ approaches unit roundoff; this component appears to decrease at the same rate as the other step components. This observation can be explained by our analysis of the case in which the cancellation error term $f_{\mathcal{B}}$ incurred in the evaluation of $g_{\mathcal{B}}(z)$ satisfies (5.14). We calculated the product $V^T(g_{\mathcal{B}}(z) + f_{\mathcal{B}})$ (the product of V with our computed version of $g_{\mathcal{B}}(z)$) and found it to be exactly zero on iterations 7, 8, and 9. Therefore, using Taylor's theorem, (2.13), and Theorem 3.3, we have

$$V^T f_{\mathcal{B}} = -V^T g_{\mathcal{B}}(z) = -V^T \nabla g_{\mathcal{B}}(z^*)(z - z^*) + O(\|z - z^*\|^2) = O(\mu^2).$$

Hence, (5.15) together with Corollary 4.3 shows that $V^T \widehat{\Delta\lambda}_{\mathcal{B}} = O(\mu)$, which is consistent with the results in Table 2. Note too that because of the higher accuracy in the $V^T \widehat{\Delta\lambda}_{\mathcal{B}}$ component, the maximum step length stays very close to 1 during the last few iterations. By comparing Tables 1 and 2, however, we can verify that the convergence of μ to zero, and of the iterates to the solution set, is not materially affected by the presence or absence of the large error in $V^T \widehat{\Delta\lambda}_{\mathcal{B}}$.

TABLE 3
Details of iteration sequence for PDIP applied to (2.8), (7.1), *with steps computed from the condensed system.*

iter	$\log \mu$	$\log \|\widehat{\Delta z}\|$	$\log \|U^T \widehat{\Delta\lambda}_{\mathcal{B}}\|$	$\log \|V^T \widehat{\Delta\lambda}_{\mathcal{B}}\|$	$\hat{\alpha}_{\max}$	λ^T
0	-1.0	-0.9	-2.1	-2.3	.9161	(1.00,.20)
1	-2.7	-1.5	-1.3	-1.4	.8872	(0.99,.20)
⋮						
5	-7.6	-5.7	-5.7	-4.2	.9999	(.93,.29)
6	-9.5	-7.7	-7.7	-6.3	1.0	(.93, .29)
7	-11.5	-9.7	-9.7	-4.3	.9999	(.93, .29)
8	-13.5	-11.7	-11.5	-2.6	.9960	(.93,.29)
9	-15.3	-13.5	-11.7	-0.6	.7386	(.93,.29)

To show that the lack of cancellation effects in Table 2 cannot be assumed in general, we modified problem (2.8) slightly, changing the second constraint to

$$g_2(z) \stackrel{\text{def}}{=} \frac{2}{3\sqrt{5}}(z_1 - \sqrt{5})^2 + z_2^2 - \frac{2\sqrt{5}}{3} \leq 0. \tag{7.1}$$

The primal and dual solutions remain unchanged, and we ran the condensed-equations version of the algorithm from the same starting point as above. Results are shown in Table 3. We observed that $g_{\mathcal{B}}(z)$ did not escape cancellation errors in this instance and, as in Table 1, we observe significant errors in $V^T \widehat{\Delta\lambda}_{\mathcal{B}}$ that do not materially affect the convergence of the algorithm to the solution set.

8. Summary and conclusions. In this paper, we have analyzed the finite-precision implementation of a PDIP method whose convergence rate is theoretically superlinear. We have made the standard assumptions that appear in most analyses of local convergence of nonlinear programming algorithms and path-following algorithms, with one significant exception: The assumption of linearly independent active constraint gradients is replaced by the weaker MFCQ which is equivalent to boundedness of the set of optimal Lagrange multipliers. Because of this assumption, it is possible that all reasonable formulations of the step equations—the linear system that needs to be solved to obtain the search direction—are ill conditioned, so it is not obvious that the numerical errors that occur when this system is solved in finite precision do not eventually render the computed search direction useless. We show that although the error in the computed step may indeed become large as μ decreases, most of the error is restricted to a subspace that does not matter, namely, the null space of the matrix $\nabla g_{\mathcal{B}}(z^*)$ of first derivatives of the active constraints. Although this error causes the computed iterates to "slip" in a tangential direction to the optimal Lagrange multiplier set, it does not interfere with rapid convergence of the iterates to the primal-dual solution set.

We found that the centrality conditions (3.11), which are usually applied in path-following methods, played a crucial role in the analysis, since they enabled us to establish the estimates (3.16) in Lemma 3.2 concerning the sizes of the basic and nonbasic components of s and λ near the solution set. The analysis of section 4, culminating in Corollary 4.4, finds bounds on the errors induced in step components by certain structured perturbations of the step equations. We show in the same section that the exact step is $O(\mu)$, allowing the local convergence analysis of Ralph and Wright [22] to be extended from convex programs to nonlinear programs.

In sections 5 and 6 we apply the general results of section 4 to the two most obvious ways of formulating and solving the step equations; namely, as a "condensed" system involving just the primal variables z, or as an "augmented" system involving both z and the Lagrange multipliers λ. In each case, the errors introduced in finite-precision implementation have the structure of the perturbations analyzed in section 4, so the error bounds obtained in Corollary 4.4 apply. In section 5.3 (whose analysis also applies to the computed solutions analyzed in section 6), we show that the potentially large error component discussed above does not interfere appreciably with the near-linear decrease of the quantities μ, r_f, and r_g to zero along the computed steps, indicating that until μ becomes quite close to $\mathbf{u}$, the convergence behavior predicted by the analysis of the "exact" algorithm will be observed in the finite-precision implementation. We conclude in section 7 with a numerical illustration of our major observations on a simple problem with two variables and two constraints, first introduced in section 2.

Appendix A. Justification of the estimates (5.17) and (5.18). To prove (5.17), we use analysis similar to that of S. Wright [30]. From the definition (3.5) of μ, and the centrality condition (3.11c), we have that

$$\lambda_i s_i = \Theta(\mu) \quad \text{for all } i = 1, 2, \ldots, m.$$

Hence, from the third block row of (3.8) and the assumption (3.7) on the size of t, we have that

$$\frac{\Delta\lambda_i}{\lambda_i} + \frac{\Delta s_i}{s_i} = -1 - \frac{t_i}{s_i\lambda_i} = -1 + O(\mu) \qquad \text{for all } i = 1, 2, \ldots, m. \tag{A.1}$$

We have from Lemma 3.2 and (4.36) that $\Delta\lambda_i/\lambda_i = O(\mu)$ for all $i \in \mathcal{B}$. Hence, by using (3.16a) from (3.2) together with (A.1), we obtain

$$\Delta s_i = -s_i + O(\mu^2) \qquad \text{for all } i \in \mathcal{B}. \tag{A.2}$$

For the computed step components $\widehat{\Delta s}_{\mathcal{B}}$, we have by combining (5.13a) with (A.2) that

$$\widehat{\Delta s}_i = -s_i + \delta_{\mathbf{u}} + O(\mu^2) \qquad \text{for all } i \in \mathcal{B}. \tag{A.3}$$

Therefore, if $s_i + \alpha\widehat{\Delta s}_i = 0$ for some $i \in \mathcal{B}$ and some $\alpha \in [0, 1]$, we have by using (3.16a) again that

$$\begin{aligned} & s_i + \alpha(-s_i + \delta_{\mathbf{u}} + O(\mu^2)) = 0 \\ & \Rightarrow (1-\alpha)s_i = \delta_{\mathbf{u}} + O(\mu^2) \\ & \Rightarrow (1-\alpha) = \delta_{\mathbf{u}}/\mu + O(\mu) \qquad \text{for any } i \in \mathcal{B}. \end{aligned} \tag{A.4}$$

Meanwhile, for $i \in \mathcal{N}$, we have from Lemma 3.2, (4.36), and (5.13a) that

$$s_i + \alpha\widehat{\Delta s}_i > 0 \qquad \text{for all } i \in \mathcal{N} \text{ and all } \alpha \in [0, 1], \tag{A.5}$$

so the components $\widehat{\Delta s}_{\mathcal{N}}$ do not place a limit on the step length bound $\hat{\alpha}_{\max}$. For the components $\widehat{\Delta\lambda}_{\mathcal{N}}$, we have by using Lemma 3.2, (4.36), (5.13c), and (A.1) that

$$\widehat{\Delta\lambda}_i = -\lambda_i + \mu\delta_{\mathbf{u}} + O(\mu^2) \qquad \text{for all } i \in \mathcal{N}.$$

Therefore, if $\lambda_i + \alpha\widehat{\Delta\lambda}_i = 0$ for some $i \in \mathcal{N}$ and some $\alpha \in [0,1]$, we have by arguing as in (A.4) that

$$1 - \alpha = \delta_{\mathbf{u}} + O(\mu). \tag{A.6}$$

Finally, for $i \in \mathcal{B}$, we have from Lemma 3.2 that $\lambda_i = \Theta(1)$, while from (4.36), (5.13a), and (5.13b), we have that

$$\Delta\lambda_i = O(\mu), \qquad \widehat{\Delta\lambda}_i = O(\mu) + \delta_{\mathbf{u}}/\mu \qquad \text{for all } i \in \mathcal{B}. \tag{A.7}$$

Therefore, we have for $\mu \gg \mathbf{u}$ that

$$\lambda_i + \alpha\widehat{\Delta\lambda}_i > 0 \qquad \text{for all } i \in \mathcal{B} \text{ and all } \alpha \in [0,1]. \tag{A.8}$$

By combining the observations (A.4), (A.5), (A.6), and (A.8), we conclude that there is a value $\hat{\alpha}_{\max}$ satisfying

$$\hat{\alpha}_{\max} \in [0,1], \qquad 1 - \hat{\alpha}_{\max} = \delta_{\mathbf{u}}/\mu + O(\mu)$$

such that

$$(\lambda, s) + \alpha(\widehat{\Delta\lambda}, \widehat{\Delta s}) > 0 \qquad \text{for all } \alpha \in [0, \hat{\alpha}_{\max}],$$

proving the claim (5.17). By making various simplifications to the analysis above, it is easy to show that (5.19) holds as well.

We now prove the claims (5.18) concerning the changes in the feasibility and duality measures along the computed step.

From (1.2), (3.11a), and the first block row of (3.8), we have

$$\begin{aligned}
&r_f(z + \alpha\widehat{\Delta z}, \lambda + \alpha\widehat{\Delta\lambda}) \\
&= \mathcal{L}_z(z + \alpha\widehat{\Delta z}, \lambda + \alpha\widehat{\Delta\lambda}) \\
&= \mathcal{L}_z(z,\lambda) + \alpha\mathcal{L}_{zz}(z,\lambda)\widehat{\Delta z} + \alpha\nabla g(z)\widehat{\Delta\lambda} + O(\alpha^2\|\widehat{\Delta z}\|^2) \\
&= (1-\alpha)\mathcal{L}_z(z,\lambda) + \alpha\mathcal{L}_{zz}(z,\lambda)(\widehat{\Delta z} - \Delta z) + \alpha\nabla g_{\mathcal{B}}(z)(\widehat{\Delta\lambda}_{\mathcal{B}} - \widehat{\Delta\lambda}_{\mathcal{B}}) \\
&\qquad + \alpha\nabla g_{\mathcal{N}}(z)(\widehat{\Delta\lambda}_{\mathcal{N}} - \Delta\lambda_{\mathcal{N}}) + O(\alpha^2\|\widehat{\Delta z}\|^2).
\end{aligned} \tag{A.9}$$

From (4.36) and (5.13a), we have $\widehat{\Delta z} = \delta_{\mathbf{u}} + O(\mu)$, so for $\mu \gg \mathbf{u}$ and $\alpha \in [0,1]$, we have

$$\alpha^2\|\widehat{\Delta z}\|^2 = O(\mu^2). \tag{A.10}$$

From the definition (2.13) of the SVD of $\nabla g_{\mathcal{B}}(z^*)$, Theorem 3.3, and (5.13a), we have that

$$\begin{aligned}
\nabla g_{\mathcal{B}}(z)(\widehat{\Delta\lambda}_{\mathcal{B}} - \Delta\lambda_{\mathcal{B}}) &= \nabla g_{\mathcal{B}}(z^*)(\widehat{\Delta\lambda}_{\mathcal{B}} - \Delta\lambda_{\mathcal{B}}) + O(\|z - z^*\|\|\widehat{\Delta\lambda}_{\mathcal{B}} - \Delta\lambda_{\mathcal{B}}\|) \\
&= \hat{U}\Sigma U^T(\widehat{\Delta\lambda}_{\mathcal{B}} - \Delta\lambda_{\mathcal{B}}) + O(\mu)\delta_{\mathbf{u}}/\mu \\
&= \delta_{\mathbf{u}}.
\end{aligned} \tag{A.11}$$

Note that the larger error (5.13b) in the component $V^T(\widehat{\Delta\lambda}_{\mathcal{B}} - \Delta\lambda_{\mathcal{B}})$, which is present when MFCQ is satisfied but not when LICQ is satisfied, does not enter into the

estimate (A.11). By substituting this estimate into (A.9) together with estimates for $\widehat{\Delta z} - \Delta z$ and $\widehat{\Delta\lambda}_{\mathcal{N}} - \Delta\lambda_{\mathcal{N}}$ from (5.13), we obtain that

$$r_f(z + \alpha\widehat{\Delta z}, \lambda + \alpha\widehat{\Delta\lambda}) = (1-\alpha)r_f + \delta_{\mathbf{u}} + O(\mu^2),$$

verifying our claim (5.18c). The potentially large error (5.13b) does not affect rapid decrease of the r_f component along the computed search direction.

For the second feasibility measure r_g, we have from (3.11b), the second block row of (3.8), and the estimates (5.13a) and (A.10) that

$$\begin{aligned}
&r_g(z + \alpha\widehat{\Delta z}, s + \alpha\widehat{\Delta s}) \\
&= g(z + \alpha\widehat{\Delta z}) + s + \alpha\widehat{\Delta s} \\
&= g(z) + \alpha\nabla g(z)^T\widehat{\Delta z} + s + \alpha\widehat{\Delta s} + O(\alpha^2\|\widehat{\Delta z}\|^2) \\
&= (1-\alpha)(g(z) + s) + \alpha\nabla g(z)^T(\widehat{\Delta z} - \Delta z) + \alpha(\widehat{\Delta s} - \Delta s) + O(\mu^2) \\
&= (1-\alpha)r_g + \delta_{\mathbf{u}} + O(\mu^2),
\end{aligned}$$

verifying (5.18d).

To examine the change in μ, we look at the change in each pairwise product $\lambda_i s_i$, $i = 1, 2, \ldots, m$. We have

$$\begin{aligned}
&(\lambda_i + \alpha\widehat{\Delta\lambda}_i)(s_i + \alpha\widehat{\Delta s}_i) \\
&= \lambda_i s_i + \alpha(s_i\widehat{\Delta\lambda}_i + \lambda_i\widehat{\Delta s}_i) + \alpha^2\widehat{\Delta s}_i\widehat{\Delta\lambda}_i \\
&= \lambda_i s_i + \alpha(s_i\Delta\lambda_i + \lambda_i\Delta s_i) + \alpha s_i(\widehat{\Delta\lambda}_i - \Delta\lambda_i) + \alpha\lambda_i(\widehat{\Delta s}_i - \Delta s_i) \\
&\qquad + \alpha^2\widehat{\Delta\lambda}_i\widehat{\Delta s}_i.
\end{aligned} \tag{A.12}$$

From the last block row in (3.8), the estimate $t = O(\mu^2)$ (3.7), and the estimate (4.36) of the exact step, we have

$$\lambda_i s_i + \alpha(s_i\Delta\lambda_i + \lambda_i\Delta s_i) = (1-\alpha)\lambda_i s_i + O(\mu^2). \tag{A.13}$$

From (4.36) and (5.13), we have

$$\widehat{\Delta\lambda}_i\widehat{\Delta s}_i = (\delta_{\mathbf{u}}/\mu + O(\mu))(O(\mu) + \delta_{\mathbf{u}}) = \delta_{\mathbf{u}} + O(\mu^2), \tag{A.14}$$

since $\mu \gg \mathbf{u}$. For $i \in \mathcal{B}$, we have from Lemma 3.2, (5.13a), and (5.13b) that

$$s_i(\widehat{\Delta\lambda}_i - \Delta\lambda_i) = O(\mu)\delta_{\mathbf{u}}/\mu = \delta_{\mathbf{u}} \quad \text{for all } i \in \mathcal{B}. \tag{A.15}$$

For $i \in \mathcal{N}$, we have from Lemma 3.2 and (5.13c) that

$$s_i(\widehat{\Delta\lambda}_i - \Delta\lambda_i) = \mu\delta_{\mathbf{u}} \quad \text{for all } i \in \mathcal{N}. \tag{A.16}$$

For the remaining term $\lambda_i(\widehat{\Delta s}_i - \Delta s_i)$, we have from Lemma 3.2 and (5.13a) that

$$\lambda_i(\widehat{\Delta s}_i - \Delta s_i) = \delta_{\mathbf{u}} \quad \text{for all } i = 1, 2, \ldots, m. \tag{A.17}$$

By substituting (A.13)–(A.17) into (A.12), we obtain

$$(\lambda_i + \alpha\widehat{\Delta\lambda}_i)(s_i + \alpha\widehat{\Delta s}_i) = (1-\alpha)\lambda_i s_i + \delta_{\mathbf{u}} + O(\mu^2) \quad \text{for all } i = 1, 2, \ldots, m. \tag{A.18}$$

Therefore, by summing over i and using (3.5), we obtain (5.18b).

Acknowledgments. Many thanks are due to an anonymous referee for close and careful readings of various versions of the paper and for many helpful suggestions.

REFERENCES

[1] E. D. Andersen, J. Gondzio, C. Mészáros, and X. Xu, *Implementation of interior-point methods for large scale linear programming*, in Interior Point Methods in Mathematical Programming, T. Terlaky, ed., Kluwer Academic Publishers, Norwell, MA, 1996, pp. 189–252.

[2] J. Bunch and L. Kaufman, *Some stable methods for calculating inertia and solving symmetric linear systems*, Math. Comp., 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] R. H. Byrd, G. Liu, and J. Nocedal, *On the Local Behavior of an Interior-Point Method for Nonlinear Programming*, Technical report 98/02, Optimization Technology Center, Argonne National Laboratory, Argonne, IL, Northwestern University, Evanston, IL, 1998.

[5] J. Czyzyk, S. Mehrotra, M. Wagner, and S. J. Wright, *PCx: An interior-point code for linear programming*, Optim. Methods Softw., 11/12 (1999), pp. 397–430.

[6] G. Debreu, *Definite and semidefinite quadratic forms*, Econometrica, 20 (1952), pp. 295–300.

[7] I. S. Duff, N. I. M. Gould, J. K. Reid, J. A. Scott, and K. Turner, *The factorization of sparse symmetric indefinite matrices*, IMA J. Numer. Anal., 11 (1991), pp. 181–204.

[8] A. El-Bakry, R. A. Tapia, and Y. Zhang, *On convergence rate of newton interior-point algorithms in the absence of strict complementarity*, Comput. Optim. Appl., 6 (1996), pp. 157–167.

[9] A. Forsgren, P. Gill, and J. Shinnerl, *Stability of symmetric ill-conditioned systems arising in interior methods for constrained optimization*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 187–211.

[10] R. Fourer and S. Mehrotra, *Solving symmetric indefinite systems in an interior-point method for linear programming*, Math. Program., 62 (1993), pp. 15–39.

[11] J. Gauvin, *A necessary and sufficient regularity condition to have bounded multipliers in nonconvex programming*, Math. Program., 12 (1977), pp. 136–138.

[12] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, 1996.

[13] N. I. M. Gould, *On the accurate determination of search directions for simple differentiable penalty functions*, IMA J. Numer. Anal., 6 (1986), pp. 357–372.

[14] N. I. M. Gould, D. Orban, A. Sartanaer, and P. Toint, *Superlinear Convergence of Primal-Dual Interior-Point Algorithms for Nonlinear Programming*, Technical report TR/PA/00/20, CERFACS, 2000.

[15] W. W. Hager, *Stabilized sequential quadratic programming*, Comput. Optim. Appl., 12 (1999), pp. 253–273.

[16] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.

[17] N. J. Higham, *Stability of the diagonal pivoting method with partial pivoting*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 52–65.

[18] O. L. Mangasarian and S. Fromovitz, *The Fritz-John necessary optimality conditions in the presence of equality and inequality constraints*, J. Math. Anal. Appl., 17 (1967), pp. 37–47.

[19] S. Mehrotra, *On the implementation of a primal-dual interior point method*, SIAM J. Optim., 2 (1992), pp. 575–601.

[20] R. D. C. Monteiro and S. J. Wright, *Local convergence of interior-point algorithms for degenerate monotone LCP*, Comput. Optim. Appl., 3 (1994), pp. 131–155.

[21] D. Ralph and S. J. Wright, *Superlinear convergence of an interior-point method for monotone variational inequalities*, in Complementarity and Variational Problems: State of the Art, M. C. Ferris and J. S. Pang, eds., SIAM, Philadelphia, 1997, pp. 345–385.

[22] D. Ralph and S. J. Wright, *Superlinear convergence of an interior-point method despite dependent constraints*, Math. Oper. Res., 25 (2000), pp. 179–194.

[23] M. H. Wright, *Ill-conditioning and computational error in interior methods for nonlinear programming*, SIAM J. Optim., 9 (1998), pp. 84–111.

[24] S. J. Wright, *Stability of linear equations solvers in interior-point methods*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1287–1307.

[25] S. J. Wright, *Modifying SQP for Degenerate Problems*, Preprint ANL/MCS-P699-1097, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1997, revised 2000.

[26] S. J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, 1996.

[27] S. J. Wright, *Stability of augmented system factorizations in interior-point methods*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 191–222.

[28] S. J. Wright, *Effects of Finite-Precision Arithmetic on Interior-Point Methods for Nonlin-*

ear Programming, Preprint ANL/MCS-P705-0198, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1998.

[29] S. J. Wright, *Superlinear convergence of a stabilized SQP method to a degenerate solution*, Comput. Optim. Appl., 11 (1998), pp. 253–275.

[30] S. J. Wright, *Modified Cholesky factorizations in interior-point algorithms for linear programming*, SIAM J. Optim., 9 (1999), pp. 1159–1191.

[31] S. J. Wright and D. Ralph, *A superlinear infeasible-interior-point algorithm for monotone nonlinear complementarity problems*, Math. Oper. Res., 21 (1996), pp. 815–838.

SIAM J. OPTIM.
Vol. 12, No. 1, pp. 79–108

THE ORDERED SUBSETS MIRROR DESCENT OPTIMIZATION METHOD WITH APPLICATIONS TO TOMOGRAPHY*

AHARON BEN-TAL†, TAMAR MARGALIT†, AND ARKADI NEMIROVSKI†

Abstract. We describe an optimization problem arising in reconstructing three-dimensional medical images from positron emission tomography (PET). A mathematical model of the problem, based on the maximum likelihood principle, is posed as a problem of minimizing a convex function of several million variables over the standard simplex. To solve a problem of these characteristics, we develop and implement a new algorithm, ordered subsets mirror descent, and demonstrate, theoretically and computationally, that it is well suited for solving the PET reconstruction problem.

Key words. positron emission tomography, maximum likelihood, image reconstruction, convex optimization, mirror descent

AMS subject classifications. 90C25, 90C90, 92C55

PII. S1052623499354564

1. Introduction. The goal of this paper is to develop a *practical* algorithm for an extremely large-scale convex optimization problem arising in nuclear medicine—that of reconstructing images from data acquired by positron emission tomography (PET).

The PET technique is described in section 2, and the corresponding mathematical optimization problem is given in section 3. The specific characteristics of the problem rule out most advanced optimization methods, and as a result we focus on gradient-type methods. Specifically, we develop an accelerated version of the mirror descent (MD) method [Nem78]. The acceleration is based on the *incremental gradient* idea [Ber95], [Ber96], [Ber97], [Luo91], [Luo94], [Tse98], also known as the *ordered subsets* (OS) technique in the medical imaging literature [Hud94], [Man95], [Kam98]. The MD method is described in section 4. The accelerated version, ordered subsets mirror descent (OSMD), is studied in section 5 in particular for a specific setup of OSMD, suitable for the PET reconstruction problem. In section 6 we report the results of testing the OSMD algorithm on several realistic cases, and also compare it to the classical SD method. Our conclusion from these tests is that OSMD is a reliable and efficient algorithm for PET reconstruction, which compares favorably with the best currently commercially used methods.

2. PET. PET is a powerful, noninvasive, medical diagnostic imaging technique for measuring the metabolic activity of cells in the human body. It has been in clinical use since the early 1990s. PET imaging is unique in that it shows the *chemical functioning* of organs and tissues, while other imaging techniques—such as X-ray, computerized tomography (CT), and magnetic resonance imaging (MRI)—show *anatomic structures*. PET is the only method that can detect and display metabolic changes in tissue; distinguish normal tissue from diseased tissue, such as in cancer; differentiate

*Received by the editors March 29, 1999; accepted for publication (in revised form) January 8, 2001; published electronically July 2, 2001.
http://www.siam.org/journals/siopt/12-1/35456.html
†MINERVA Optimization Center, Faculty of Industrial Engineering and Management, Technion–Israel Institute of Technology, Haifa, Israel (abental@ie.technion.ac.il, tammar@ie.technion.ac.il, nemirovs@ie.technion.ac.il). This research is part of the PARAPET Project supported by the EUC Esprit Programme.

viable from dead or dying tissue; show regional blood flow; and determine the distribution and fate of drugs in the body. It is useful clinically in patients with certain conditions affecting the brain and the heart as well as in patients with certain types of cancer. Because of its accuracy, effectiveness, and cost efficiency, PET is becoming indispensable for the diagnosis of disease and treatment of patients.

2.1. The physical principles of PET. A PET scan involves the use of a small amount of a radioactive material which has the property of emitting positrons (positively charged electrons). Such a substance is referred to as *positron emitter*. One of the prime reasons for the importance of PET in medical research and practice is the existence of positron-emitting isotopes of elements such as carbon, oxygen, and fluorine. These isotopes can be attached or tagged to biochemical compounds such as glucose, ammonia, water, etc., to form radioactive tracers that will mimic their stable counterparts biologically (i.e., the radio-tracer element does not modify the biochemical behavior of the molecule). The choice of the biochemical compound and the radioactive tracer depends on the particular medical information being sought. When these radioactive drugs (or "radio-pharmaceuticals") are administered to a patient, either by injection or by inhalation of gas, they distribute within the body according to the physiologic pathways associated with their stable counterparts.

The scan begins after a delay ranging from seconds to minutes to allow for the radio-tracer transport to the organ of interest. Then, the radio-isotope decays to a more stable atom by emitting a positron from its nucleus. The emitted positron loses most of its kinetic energy after traveling only a few millimeters in living tissue. It is then highly susceptible to interaction with an electron, an event that annihilates both particles. The mass of the two particles is converted into 1.02 million electron volts of energy, divided equally between two gamma rays.

The two gamma rays fly off the point of annihilation in nearly opposite directions along a line with a completely random orientation (i.e., uniformly distributed in space). They penetrate the surrounding tissue and are recorded outside the patient by a PET scanner consisting of circular arrays (*rings*) of gamma radiation detectors.

Since the two gamma rays are emitted simultaneously and travel in almost exactly opposite directions, their source can be established with high accuracy. This is achieved by grouping the radiation detectors in pairs. Two opposing detectors register a signal only if both sense high-energy photons within a short ($\sim 10^{-8}$sec) timing window. Detection of two events at the same time is referred to as *coincidence event*. Each detector is in coincidence with a number of detectors opposite so as to cover a *field of view* (FOV) about half as large in diameter as the diameter of the detector array.

A coincidence event is assigned to a *line of response* (LOR) connecting the two relevant detectors. In the two-dimensional case, an LOR is identified by the angle ϕ and the distance s from the scanner axis (the center of the FOV). A certain pair of detectors is identified by the LOR joining their centers, and is sometimes referred to as a *bin*. The total number of coincidence events detected by a specific pair of detectors approximates the line integral of the radio-tracer concentration along the relevant LOR. Considering the total number of coincidence events detected by all pairs of detectors with the same angle ϕ, we get a parallel set of such line integrals, known as a *parallel projection* set or shortly, as a projection.

The measured data set is the collection of numbers of coincidences counted by different pairs of detectors, or equivalently, the number of counts in all bins that intersect the FOV. Based on the measured data, a mathematical algorithm, applied

by a computer, tries to reconstruct the spatial distribution of the radioactivity within the body. The principle of image reconstruction by computerized tomography is that an object can be reproduced from a set of its projections taken at different angles. The validity of such a reconstruction depends, of course, on the number of counts collected. The number of projections is a parameter of the scanner, and it determines the size of the mathematical reconstruction problem.

Note that there are several factors affecting quantitative accuracy of the measured data (e.g., detector efficiency, attenuation, scatter, random events, etc.). Therefore, the total number of counts is typically much smaller than the total number of emissions.

The final result of the scan study is usually presented as a set of two-dimensional (2D) images (known as *slices*), which together compose the three-dimensional (3D) mapping of the tracer distribution within the body.

3. The optimization problem. For consistent data, i.e., free of noise and measurement errors, there is a unique analytic solution of the 2D inversion problem of recovering a 2D image from the set of its one-dimensional (1D) projections. This solution was derived by Radon in 1917 and later became the basis for computerized tomography. The method, named *filtered back-projection* (FBP), was first applied for 2D PET image reconstruction by Shepp and Logan in 1974 [She74].

The images obtained by the FBP method as well as other analytical methods, which are based on inverse transforms, tend to be "streaky" and noisy. To address the problem of noise, the study of statistical (iterative) reconstruction techniques has received much attention in the past few years. Iterative methods allow incorporation of physical constraints and a priori knowledge not contained in the measured projections, e.g., the Poisson nature of the emission process.

The formulation of the PET reconstruction problem as a maximum likelihood (ML) problem rather than as an inverse problem was initially suggested by Rockmore and Mackovski in 1976 [Roc76]. It became feasible when Shepp and Vardi in 1982 [She82] and Vardi, Shepp, and Kaufman in 1985 [Var85] showed how the expectation maximization (EM) algorithm could be used for the ML computation.

3.1. Mathematical model and the ML problem. The goal of ML estimation, as applied to emission tomography, is to find the expected number of annihilations by maximizing the probability of the set of observations, i.e., the detected coincidence events.

The mathematical model is based on the realistic assumption that photon counts follow a Poisson process. To simplify the computations, we form a finite parameter space by imposing a grid of boxes (*voxels*) over the emitting object. Let $X(j)$ denote the number of radioactive events emitted from voxel j. It is assumed that $X(j), j = 1, \ldots, n$, are independent Poisson-distributed random variables with unknown means λ_j,

$$X(j) \sim \text{Poisson}(\lambda_j).$$

Let p_{ij} be the probability that an emission from voxel j will be detected in bin i. Note that p_{ij} defines a transition matrix (likelihood matrix) assumed to be known from the geometry of the detector array. The probability to detect an event emitted from voxel j is

$$p_j = \sum_{i=1}^{m} p_{ij}. \tag{1}$$

The number of events emitted from voxel j and detected in bin i is defined by $X(i,j) = p_{ij}X(j)$. By a Bernoulli thinning process with the probabilities p_{ij}, for different j and i, $\{X(i,j)\}$ are also independent Poisson random variables. Let $Y(i)$ denote the total number of events detected by bin i, i.e.,

$$Y(i) = \sum_j p_{ij}X(i,j); \tag{2}$$

then $Y(i)$ is also a Poisson random variable, with the mean

$$\mu_i = \sum_j p_{ij}\lambda_j, \tag{3}$$

and $Y(i)$'s are independent of each other. A more accurate model of the observations would be

$$\mu_i = \sum_j m_i p_{ij}\lambda_j + r_i + s_i,$$

where, r_i, and s_i are known values for random and scatter coincidences and m_i are known attenuation coefficients, but we will use the simplified model. We denote by y_i the observations, namely the realizations of the random variables $Y(i)$.

The problem of PET image reconstruction can be formulated in the context of an incomplete data problem: the *complete data* (but unobserved) are the number of counts emitted from each voxel ($X(j)$); the *incomplete data* (observed) are counts of photons collected in various bins (y_i); and the parameter to be estimated is the expected number of counts emitted from each voxel (λ_j). Thus, the reconstruction problem is equivalent to a parameter estimation problem, and a *maximum likelihood* function can be formulated. In general, the likelihood function can be defined as the joint probability density of the measured data known up to the unobservable parameters to be estimated. Maximizing this likelihood function with respect to the unobservable parameters yields the parameters with which the data are most consistent.

According to (2) and (3) the vector of observed data $y = (y_1, \ldots, y_m)^T$ has the following likelihood function:

$$\begin{aligned} L(\lambda) = p(Y = y|\lambda) &= \prod_{i=1}^{m} e^{-\mu_i} \frac{\mu_i^{y_i}}{y_i!} \\ &= \prod_{i=1}^{m} \left(\exp[-\textstyle\sum_{j=1}^{n} \lambda_j p_{ij}] \frac{[\sum_{j=1}^{n} \lambda_j p_{ij}]^{y_i}}{y_i!} \right). \end{aligned} \tag{4}$$

The maximum likelihood estimate of λ is the vector $\overline{\lambda}$ maximizing $L(\lambda)$ or equivalently its logarithm:

$$\ln L(\lambda) = -\sum_{j=1}^{n} \lambda_j p_j + \sum_{i=1}^{m} y_i \ln(\sum_{j=1}^{n} \lambda_j p_{ij}) - \text{constant}. \tag{5}$$

Note that the function $\ln L(\lambda)$ is concave [She82]. Therefore, we can write the following convex minimization problem with nonnegativity constraints:

$$F(\lambda) \equiv \sum_{j=1}^{n} p_j \lambda_j - \sum_{i=1}^{m} y_i \ln \left(\sum_{j=1}^{n} p_{ij}\lambda_j \right) \to \min \mid \ \lambda \geq 0. \tag{6}$$

The optimal solution to the problem is the ML estimate of the (discretized) density of the tracer.

Problem (6) is an extremely large-scale convex optimization program: the design dimension n (the number of voxels) normally is $128^3 = 2,097,152$, while the number m of bins (i.e., the number of log-terms in the objective) can vary from 6,000,000 to 20,000,000, depending on the type of the tomograph. On a 450 MHz Pentium III computer with 200 Mb RAM, a single computation of the value and the gradient of the objective (i.e., multiplication of given vectors once by the matrix $P = ||p_{ij}||$ and once by P^T) takes from 15 to 45 minutes, depending on m.

The huge sizes of the PET image reconstruction problem impose severe restrictions on the type of optimization techniques which could be used to solve (6):

A. With the design dimension of order of $n = 10^6$, the only option is to use methods whose computational effort per iteration is linear in n. Even with this complexity per iteration, the overall number of iterations should be at most few tens—otherwise the running time of the method will be too large for actual clinical applications.

B. The objective in (6) is not defined on the whole $\mathbf{R}^n$ and may blow up to ∞ as λ approaches a "bad" boundary point of the nonnegative orthant (e.g., the origin); moreover, (6) is a constrained problem, however simple the constraint might look.

Observation A rules out basically all advanced optimization methods, like interior point ones (or other Newton-based optimization techniques): in spite of the fast convergence in terms of iteration counts, these techniques (at least in their "theoretically valid" forms) will "never" finish even the first iteration. In principle, it could be possible to use quasi-Newton techniques. Such an approach, however, would require resolving difficulties coming from B, without a clear reward for the effort: to the best of our knowledge, in the case when the number of iterations is restricted to only a small fraction of the design dimension (see A), there is no theoretical or computational evidence in favor of quasi-Newton methods.

Consequently, in our case, the most promising methods seem to be simple gradient-descent type methods aimed at solving convex problems with simple constraints. For these methods, the complexity per iteration is linear in n. Moreover, in favorable circumstances, the rate of convergence of gradient-type methods, although poor, is independent (or nearly so) of the design dimension. As a result, with a gradient-type method one usually reaches the first one or two digits of the optimal value in a small number of iterations, and then the method "dies," i.e., in many subsequent iterations no more progress in accuracy is obtained. Note that this "convergence pattern" is, essentially, what is needed in the PET image reconstruction problem. Indeed, this is an inverse (and, as such, an ill-posed) optimization problem; practice demonstrates that when solving it to high accuracy, in terms of the optimal value (which is possible in the 2D case), the quality of the image first improves and then tends to deteriorate, resulting eventually in a highly noisy image. Thus, in the case in question we in fact are not interested in high-accuracy solutions, which makes gradient descent techniques an appropriate choice.

4. The mirror descent scheme and minimization over a simplex.

4.1. The general mirror descent scheme. The general mirror descent (gMD) scheme is aimed at solving a convex optimization problem

$$f(x) \to \min \mid x \in X \subset \mathbf{R}^n, \tag{7}$$

where X is a convex compact set in $\mathbf{R}^n$ and f is a Lipschitz continuous convex function on X.

Note that the PET image reconstruction problem with $p_{ij} > 0$ can be easily converted to (7). Indeed, from the KKT conditions for (6) we deduce the complementarity equations

$$\left(p_j - \sum_i y_i \frac{p_{ij}}{\sum_\ell p_{i\ell}\lambda_\ell} \right) \lambda_j = 0, \;\; j = 1, \ldots, n.$$

Summing up these equations, we see that any optimal solution λ to problem (6) must satisfy the equation

$$\sum_j p_j \lambda_j = B \equiv \sum_i y_i.$$

Thus, we loose nothing by adding to problem (6) the equality constraint $\sum_j p_j \lambda_j = B$. If we further introduce the change of variables

$$x_j = \frac{p_j \lambda_j}{B},$$

we end up with the optimization program

$$(8) \quad f(x) \equiv -\sum_{i=1}^m y_i \ln \left(\sum_j r_{ij} x_j \right) \to \min \mid \; x \in \Delta_n \equiv \left\{ x \in \mathbf{R}^n_+ : \sum_i x_i = 1 \right\},$$

where

$$r_{ij} = B \frac{p_{ij}}{p_j},$$

which is equivalent to (6). The new formulation (8) is of the form (7), with the standard simplex Δ_n playing the role of X. Besides this, the resulting objective f is convex and Lipschitz continuous on $X = \Delta_n$, provided that $p_{ij} > 0$.

The *setup* for the gMD method is given by the following entities:

1. a *compact convex* set $Y \supset X$;
2. a *norm* $\|\cdot\|$ on $\mathbf{R}^n$ and its associated projector of Y onto X

$$\pi(y) \in \operatorname*{Argmin}_{x \in X} \|y - x\|,$$

along with the corresponding *separator*

$$(9) \quad \eta(y) \in \mathbf{R}^n : \quad \|\eta(y)\|_* \le 1, \quad \eta^T(y)(y - x) \ge \|y - \pi(y)\| \;\; \forall x \in X,$$

where

$$\|\xi\|_* = \max\{\xi^T x \mid \; \|x\| \le 1\}$$

is the norm on $\mathbf{R}^n$ conjugate to $\|\cdot\|$;

3. a positive real α and a continuously differentiable convex function $w : Y \to \mathbf{R}$ which is α-strongly convex on Y w.r.t. the norm $\|\cdot\|$, i.e.,

$$(w'(x) - w'(y))^T (x - y) \ge \alpha \|y - x\|^2 \quad \forall x, y \in Y \qquad (w' \equiv \nabla w).$$

It is assumed that we can compute efficiently

- the projector $\pi(y)$ and the separator $\eta(y)$, $y \in Y$;
- the Legendre transformation

$$W(\xi) = \max_{y \in Y} \left[\xi^T y - w(y)\right]$$

of $w(\cdot)$, $\xi \in \mathbf{R}^n$.

Note that α-strong convexity of w on Y implies, via the standard duality relations [RW98, Proposition 12.54], that W is continuously differentiable on the entire $\mathbf{R}^n$ with Lipschitz continuous gradient

$$\|W'(\xi) - W'(\eta)\| \le \frac{1}{\alpha}\|\xi - \eta\|_* \quad \forall \xi, \eta \in \mathbf{R}^n. \tag{10}$$

Moreover, the mapping $\xi \mapsto W'(\xi) = \operatorname{argmax}_{x \in Y}[\xi^T x - w(x)]$ is a parameterization of Y.

The gMD method for solving (7) generates sequences $\xi_t \in \mathbf{R}^n$, $\widehat{x}_t \in Y$, $x_t \in X$ as follows:

- *Initialization:* Choose (arbitrarily) $x_0 \in X$ and set $\xi_1 = w'(x_0)$;
- *Step t,t* $= 1, 2, \ldots$:

(S.1) Set

$$\widehat{x}_t = W'(\xi_t); \quad x_t = \pi(\widehat{x}_t); \quad \eta_t = \eta(\widehat{x}_t).$$

(S.2) Compute the value $f(x_t)$ and a subgradient $f'(x_t)$ of f at x_t. If $f'(x_t) = 0$, then x_t is the exact minimizer of f on X, and we terminate. If $f'(x_t) \neq 0$, we set

$$\xi_{t+1} = w'(\widehat{x}_t) - \gamma_t[f'(x_t) + \|f'(x_t)\|_* \eta_t], \tag{11}$$

where $\gamma_t > 0$ is a stepsize, and pass to step $t + 1$.

- *Approximate solution* x^t generated in the course of the first t steps of the method is the best (with the smallest value of f) of the points $x_1, \ldots, x_t$: $x^t \in \operatorname{Argmin}_{x \in \{x_1, \ldots, x_t\}} f(x)$.

The convergence properties of the MD method are summarized in the following theorem.

THEOREM 4.1. *Assume that f is convex and Lipschitz continuous on X, with Lipschitz constant, w.r.t. $\|\cdot\|$, equal to $L_{\|\cdot\|}(f)$, and that the subgradients $f'(x_t)$ used in the gMD satisfy the condition*

$$\|f'(x_t)\|_* \le L_{\|\cdot\|}(f).$$

Then for every $t \ge 1$ one has

$$f(x^t) - \min_{x \in X} f(x) \le \min_{1 \le s \le r \le t} \frac{\Gamma(w) + \frac{2}{\alpha} \sum_{\tau=s}^{r} \gamma_\tau^2 \|f'(x_\tau)\|_*^2}{\sum_{\tau=s}^{r} \gamma_\tau}, \tag{12}$$

where

$$\Gamma(w) = \max_{x,y \in Y} [w(x) - w(y) - (x - y)^T w'(y)].$$

In particular, whenever $\gamma_t \to +0$ *as* $t \to \infty$ *and* $\sum_\tau \gamma_\tau = \infty$, *one has* $f(x^t) - \min_{x\in X} f(x) \to 0$ *as* $t \to \infty$. *Moreover, with the stepsizes chosen as*

$$\gamma_\tau = \frac{C(\alpha\Gamma(w))^{1/2}}{\|f'(x_\tau)\|_*\sqrt{t}}, \tag{13}$$

one has

$$f(x^t) - \min_{x\in X} f(x) \le \widehat{C}(C) L_{\|\cdot\|}(f)\sqrt{\frac{\Gamma(w)}{\alpha}}t^{-1/2}, \ \ t = 1, 2, \ldots, \tag{14}$$

with certain universal function $\widehat{C}(\cdot)$.

The theorem, in a slightly modified setting, is proved in [Nem78]. Here it will be derived as a straightforward simplification of the proof of Theorem 5.1 below.

4.2. $\|\cdot\|_p$-MD and minimization over the standard simplex. As we have seen, the PET image reconstruction problem can be converted to the form of (8), i.e., posed as the problem of minimizing a convex function f over the standard simplex Δ_n. Therefore we focus on the gMD scheme as applied to the particular case of $X = \Delta_n$.

Let us choose somehow $p \in (1, 2]$ and consider the following setup for gMD:

$$Y = \{x \mid \|x\|_p \le 1\} \ [\supset \Delta_n]; \quad \|\cdot\| = \|\cdot\|_p; \quad w(x) = \frac{1}{2}\|x\|_p^2. \tag{15}$$

This setup defines a family $\{\mathrm{MD}_p\}_{1<p\le 2}$ of *ℓ_p-MD methods* for minimizing convex functions over the standard simplex Δ_n (in fact, MD_p can be used to minimize a convex function over a convex subset of the unit $\|\cdot\|_p$-ball). A natural question is, *Which one of these methods is best suited for minimization over* Δ_n *?* To answer this question, note first that for setup (15), a straightforward calculation yields that

$$W(\xi) = \begin{cases} \frac{1}{2}\|\xi\|_q^2, & \|\xi\|_q \le 1 \\ \|\xi\|_q - \frac{1}{2}, & \|\xi\|_q > 1 \end{cases}, \quad q = \frac{p}{p-1}. \tag{16}$$

Moreover, it is known (to be self-contained, we reproduce the proof in Appendix 1) that the parameter α of strong convexity of w w.r.t. the $\|\cdot\|_p$-norm satisfies the relation

$$\alpha \equiv \alpha_p(n) \ge O(1)(p-1), \tag{17}$$

and the quantity $\Gamma(w)$ defined in (12) is

$$\Gamma(w) = O(1)$$

(here and in what follows, $O(1)$ are appropriate positive absolute constants). Consequently, the efficiency estimate (14) becomes

$$f(x^t) - \min_{x\in X} f(x) \le \widehat{C}(C)\frac{L_{\|\cdot\|_p}(f)}{\sqrt{p-1}}t^{-1/2}, \ \ t = 1, 2, \ldots. \tag{18}$$

Recalling that for every $x \in \mathbf{R}^n$ one clearly has $\|x\|_p \le \|x\|_1 \le \|x\|_p n^{\frac{p-1}{p}}$, and therefore

$$L_{\|\cdot\|_1}(f) \le L_{\|\cdot\|_p}(f) \le L_{\|\cdot\|_1} n^{\frac{p-1}{p}},$$

we derive from (18) that

$$f(x^t) - \min_{x\in X} f(x) \le \widehat{C}(C)\frac{n^{\frac{p-1}{p}}}{\sqrt{p-1}} L_{\|\cdot\|_1}(f) t^{-1/2}, \quad t = 1, 2, \ldots. \tag{19}$$

Assuming $n > 1$ and minimizing the right-hand side over $p \in (1, 2]$, we see that a good choice of p is

$$p = p(n) = 1 + \frac{O(1)}{\ln n}. \tag{20}$$

With this choice of p, the efficiency estimate (19) becomes

$$f(x^t) - \min_{x\in X} f(x) \le \widehat{C}(C)\frac{\sqrt{\ln n} L_{\|\cdot\|_1}(f)}{\sqrt{t}}, \quad t = 1, 2, \ldots, \tag{21}$$

while the underlying stepsizes are

$$\gamma_t = \frac{C}{\|f'(x_\tau)\|_* \sqrt{\ln n}\sqrt{t}} \qquad [C > 0]. \tag{22}$$

In what follows, we refer to the MD method with the setup given by (15), (20), (22) (where $C = O(1)$) as to $\|\cdot\|_1$-*MD method* MD_1.

Discussion. In the family $\{\mathrm{MD}_p\}_{1\le p\le 2}$ of MD methods, the special case MD_2 is well known—it is a kind of the standard subgradient descent (SD) method originating from [Sho67] and [Pol67] and studied in numerous papers. (For the "latest news" on SD, see [KLP99] and references therein.) The only modification needed to get from the MD scheme not a "kind of" the SD but *exactly* the standard SD method

$$x_{t+1} = \pi_X(x_t - \gamma_t f'(x_t)), \quad \pi_X(x) = \operatorname*{argmin}_{y\in X} \|x - y\|_2 \tag{23}$$

for minimizing a convex function over a convex subset X of the unit Euclidean ball, is to set in (15) $p = 2$ and $Y = X$ rather than $p = 2$ and $Y = \{x \mid \|x\|_2 \le 1\}$. Our analysis demonstrates, however, that when minimizing over the standard simplex, the "non-Euclidean" mirror descent MD_1 is preferable to the usual SD. Indeed, the best efficiency estimate known so far for SD as applied to minimizing a convex Lipschitz continuous function f over the standard simplex Δ_n is

$$f(x^t) - \min_{x\in\Delta_n} f(x) \le O(1)\frac{L_{\|\cdot\|_2}(f)}{\sqrt{t}},$$

while the efficiency bound for MD_1 is

$$f(x^t) - \min_{x\in\Delta_n} f(x) \le O(1)\frac{\sqrt{\ln n} L_{\|\cdot\|_1}(f)}{\sqrt{t}};$$

the ratio of these efficiency estimates is

$$R = O(1)\frac{L_{\|\cdot\|_2}(f)}{\sqrt{\ln n} L_{\|\cdot\|_1}(f)}.$$

Now, the ratio $L_{\|\cdot\|_2}(f)/L_{\|\cdot\|_1}(f)$ is always ≥ 1 and can be as large as $O(1)\sqrt{n}$ (in the case where all partial derivatives of f are of order of 1, and their sum is identically zero). It follows that for the problem of minimization over the standard simplex, *as far as the efficiency estimates are concerned*, the "non-Euclidean" mirror descent MD_1 can outperform the standard SD by a factor of the order of $(n/\ln n)^{1/2}$, which, for large n, can make a huge difference.

4.3. MD_1 and complexity of large-scale convex minimization over a simplex. We next show that the efficiency estimate of MD_1 as applied to minimization of Lipschitz continuous functions over an n-dimensional simplex cannot be improved by more than an $O(\ln n)$-factor, provided that n is large. Thus, MD_1 is a "nearly optimal" method, in the sense of information-based complexity theory, for large-scale convex minimization over the standard simplex.

Consider the family $\mathcal{F} \equiv \mathcal{F}(L,n)$ of all problems

$$f(x) \to \min \mid \; x \in \Delta_n \equiv \left\{ x \in \mathbf{R}^n_+ : \sum_{i=1}^n x_i = 1 \right\}$$

associated with convex functions $f : \Delta_n \to \mathbf{R}$ which are Lipschitz continuous and whose Lipschitz constant (taken w.r.t. $\|\cdot\|_1$) does not exceed a given positive L. The *information-based complexity* $\mathrm{Compl}(\varepsilon)$ *of the family* $\mathcal{F}$ is defined as follows. Let $\mathcal{B}$ be a routine which, as applied to a problem f from the family $\mathcal{F}$, successively generates *search points* $x_t = x_t(\mathcal{B}, f) \in \mathbf{R}^n$ and *approximate solutions* $x^t = x^t(\mathcal{B}, f)$; the only restriction on the mechanism of generating the search points and the approximate solutions is that both x_t and x^t should be deterministic functions of the values $f(x_\tau)$ and the subdifferentials $\partial f(x_\tau)$ of the objective taken at the previous search points x_τ, $\tau < t$, so that x_1, x^1 are independent of f, x_2, x^2 depend only on $f(x_1), \partial f(x_1)$, and so on. We define the *complexity of* $\mathcal{F}$ *w.r.t.* $\mathcal{B}$ as the function

$$\mathrm{Compl}_{\mathcal{B}}(\varepsilon) = \inf\{T : f(x^t(\mathcal{B}, f)) - \min_{\Delta_n} f \le \varepsilon \quad \forall (t \ge T, f \in \mathcal{F})\},$$

i.e., as the smallest number of steps after which the inaccuracy of approximate solutions generated by $\mathcal{B}$ is at most ε, whatever is $f \in \mathcal{F}$. The *complexity of the family* $\mathcal{F}$ is defined as

$$\mathrm{Compl}(\varepsilon) = \min_{\mathcal{B}} \mathrm{Compl}_{\mathcal{B}}(\varepsilon),$$

where the minimum is taken over all aforementioned "solution methods" $\mathcal{B}$. Note that the efficiency bound (21) says that

$$\mathrm{Compl}_{\mathrm{MD}_1}(\varepsilon) \le O(1) \left[\frac{L^2 \ln n}{\varepsilon^2} + 1 \right], \quad \varepsilon > 0. \tag{24}$$

On the other hand, the following statement takes place (for the proof, see Appendix 2).

PROPOSITION 4.2. *The information-based complexity of the family* $\mathcal{F}(L,n)$ *is at least* $O(1)\min[\frac{L^2}{\varepsilon^2}, n]$.

Comparing (24) with the lower complexity bound given by Proposition 4.2, we see that in the case of $\varepsilon \ge Ln^{-1/2}$ the accuracy guarantees given by MD_1 as applied to optimization problems from $\mathcal{F}$ cannot be improved by more than factor $O(\ln n)$.

5. Incremental gradient version of the MD scheme—the OSMD method. The objective function in the PET image reconstruction problem is a sum of a huge number m of simple convex functions. A natural way to exploit this fact in order to reduce the computational effort per iteration is offered by the *incremental gradient* technique (see, e.g., [Ber95]), which in the medical imaging literature is known as the OS scheme (see [Hud94]).

The idea of the OS scheme is very simple: when solving problem (7) with the objective of the form

$$f(x) = \sum_{\ell=1}^{k} f_\ell(x), \tag{25}$$

one replaces at iteration t the "true" gradient $f'(x_t)$ with "partial gradient" $f'_{\ell(t)}(x_t)$, with $\ell(t)$ running, in the cyclic order, through the set $1, \dots, k$ of indices of the components $f_1, \dots, f_k$. With this approach, one reduces the computational effort required to compute f' and thus reduces the complexity of an iteration. Computational practice in many cases demonstrates that such a modification does not much affect the quality of approximate solutions generated after a given number of iterations, provided that k is not too large.

Below, we present the OS version of the general MD scheme and demonstrate that its convergence properties are similar to those of the original scheme.

The OSMD scheme for solving problem (7) with objective of the form (25) (where all components f_ℓ are convex and Lipschitz continuous on X) has the same setup $(Y, X, \|\cdot\|, w, W)$ as the original gMD scheme and is as follows:

• *Initialization:* Choose $x_0 \in X$ and set $\xi_1 = w'(x_0)$;

• *Outer iteration* t,$t = 1, 2, \dots$:

(O.1) Given ξ_t, run a k-iteration *inner loop* as follows:

- Initialization: Set $\xi_t^1 = \xi_t$;
- Inner iteration ℓ, $\ell = 1, \dots, k$:

I.1) Given ξ_t^ℓ, compute

$$\widehat{x}_t^\ell = W'(\xi_t^\ell); \quad x_t^\ell = \pi(\widehat{x}_t^\ell); \quad \eta_t^\ell = \eta(\widehat{x}_t^\ell)$$

(cf. step S.1 in the original MD scheme).

I.2) Compute the value $f_\ell(x_t^\ell)$ and a subgradient $f'_\ell(x_t^\ell)$ of f_ℓ at the point x_t^ℓ and set

$$\xi_t^{\ell+1} = \xi_t^\ell - \gamma_t[f'_\ell(x_t^\ell) + \|f'_\ell(x_t^\ell)\|_* \eta_t^\ell],$$

where $\gamma_t > 0$ is a stepsize.

(O.2) Set

$$\xi_{t+1} = w'(W'(\xi_t^{m+1}))$$

and pass to outer iteration $t + 1$.

• *Approximate solution* x^t generated in course of t steps of the method is the point $x^1_{\tau(t)}$, where

$$\tau(t) \in \operatorname*{Argmin}_{t/2 \le \tau \le t} \widetilde{f}_\tau, \quad \widetilde{f}_\tau = \sum_{\ell=1}^{k} f_\ell(x_\tau^\ell)$$

(note that $\widetilde{f}_\tau$ is a natural estimate of $f(x_\tau^1)$).

The main theoretical result of our paper summarizes the convergence properties of the OS version of the MD scheme in the following theorem.

THEOREM 5.1. *Assume that f_ℓ, $\ell = 1, \dots, m$, are convex and Lipschitz continuous on X, with Lipschitz constants w.r.t. $\|\cdot\|$ not exceeding $L_{\|\cdot\|}(f)$, and that the subgradients $f'_\ell(x_t^\ell)$ used in the MD method satisfy the condition*

$$\|f'_\ell(x_t^\ell)\|_* \le L_{\|\cdot\|}(f).$$

Assume, in addition, that the $\|\cdot\|$-projector $\pi(\cdot)$ is Lipschitz continuous on Y, with a Lipschitz constant β w.r.t. $\|\cdot\|$, i.e.,

$$\|\pi(x)-\pi(x')\| \le \beta\|x-x'\| \quad \forall x,x' \in Y.$$

Then for every $t \ge 1$ one has

$$f(x^t)-\min_{x\in X} f(x) \le \frac{\Gamma(w)+2k(k+1)\beta\alpha^{-1}L^2_{\|\cdot\|}(f)\sum\limits_{t/2\le\tau\le t}\gamma_\tau^2}{\sum\limits_{t/2\le\tau\le t}\gamma_\tau} + 4k^2\beta\alpha^{-1}L^2_{\|\cdot\|}(f)\max_{t/2\le\tau\le t}\gamma_\tau. \tag{26}$$

In particular, whenever $\gamma_t \to +0$ and $\sum_{t/2\le\tau\le t}\gamma_\tau \to \infty$ as $t\to\infty$, one has $f(x^t)-\min_{x\in X} f(x)\to 0$ as $t\to\infty$. Moreover, with the stepsizes chosen as

$$\gamma_t = \frac{(\alpha\beta^{-1}\Gamma(w))^{1/2}}{kL_t\sqrt{t}}, \tag{27}$$

where L_t are any numbers satisfying

$$0 < L_{\min} \le L_t \le L_{\max} < \infty,$$

one has

$$f(x^t)-\min_{x\in X} f(x) \le O(1)k\sqrt{\frac{\beta\Gamma(w)}{\alpha}}\left(L_{\max}+\frac{L^2_{\|\cdot\|}(f)}{L_{\min}}\right)t^{-1/2}, \quad t=1,2,\dots. \tag{28}$$

Proof. 1^0. Let x_* be a minimizer of f on X, let $W_*(\xi)=W(\xi)-\xi^T x_*$, and let

$$g_\tau^\ell = f'_\ell(x_\tau^\ell), \quad h_\tau^\ell = g_\tau^\ell + \|g_\tau^\ell\|_*\eta_\tau^\ell.$$

Observe, first, that from $\|\eta_\tau^\ell\|_* \le 1$ and $\|f'_\ell(x_\tau^\ell)\|_* \le L_{\|\cdot\|}(f)$ it follows that

$$\|h_\tau^\ell\|_* \le 2\|g_\tau^\ell\|_* \le 2L, \quad L = L_{\|\cdot\|}(f), \tag{29}$$

whence

$$\|\xi_\tau^\ell - \xi_\tau^{\ell+1}\|_* \le 2\gamma_\tau L.$$

Besides this, by (10) and by assumptions on $\pi(\cdot)$ and f_ℓ we have

$$\begin{gathered}\|W'(\xi)-W'(\eta)\| \le \tfrac{1}{\alpha}\|\xi-\eta\|_* \quad \forall \xi,\eta\in\mathbf{R}^n,\\ \|\pi(x)-\pi(y)\| \le \beta\|x-y\| \quad \forall x,y\in Y,\\ |f_\ell(x)-f_\ell(y)| \le L\|x-y\| \quad \forall x,y\in X.\end{gathered}$$

Combining these relations and taking into account the description of the method, we get

$$\begin{array}{ll}\text{(a)} & \|\widehat{x}_\tau^\ell - \widehat{x}_\tau^1\| \le 2k\alpha^{-1}\gamma_\tau L, \qquad \ell=1,\dots,k;\\ \text{(b)} & \|x_\tau^\ell - x_\tau^1\| \le 2k\beta\alpha^{-1}\gamma_\tau L, \qquad \ell=1,\dots,k;\\ \text{(c)} & |f_\ell(x_\tau^\ell)-f_\ell(x_\tau^1)| \le 2k\beta\alpha^{-1}\gamma_\tau L^2, \qquad \ell=1,\dots,k.\end{array} \tag{30}$$

2^0. Since W_* differs from W by a linear function, relation (10) holds true for W_* as well, whence

$$\begin{aligned} W_*(\xi+\eta) &= W_*(\xi)+\eta^T W_*'(\xi)+\int_0^1 [W_*'(\xi+t\eta)-W_*'(\xi)]^T\eta\,dt \\ &\le W_*(\xi)+\eta^T W_*'(\xi)+\frac{1}{2\alpha}\|\eta\|_*^2. \end{aligned} \tag{31}$$

Besides this, whenever $\xi\in\mathbf{R}^n$, we have

$$W'(\xi)=\mathrm{argmax}_{x\in Y}[\xi^T x-w(x)],$$

and since w is continuously differentiable on Y, it follows that

$$[\xi-w'(W'(\xi))]^T(W'(\xi)-y)\ge 0\quad\forall y\in Y.$$

It follows that

$$\begin{array}{rcl} W_*(\xi) &=& W(\xi)-\xi^T x_* = \xi^T W'(\xi)-w(W'(\xi))-\xi^T x_* \\ &=& [w'(W'(\xi))]^T W'(\xi)-w(W'(\xi)) \\ && +[\xi-w'(W'(\xi))]^T(W'(\xi)-x_*)-[w'(W'(\xi))]^T x_* \\ &\ge& [w'(W'(\xi))]^T W'(\xi)-w(W'(\xi))-[w'(W'(\xi))]^T x_* \\ &=& W_*(w'(W'(\xi))). \end{array} \tag{32}$$

We now have

$$\begin{array}{rcll} & W_*(\xi_\tau^{\ell+1}) = W_*(\xi_\tau^\ell-\gamma_\tau h_\tau^\ell) & & \\ \le & W_*(\xi_\tau^\ell)-\gamma_\tau[h_\tau^\ell]^T W_*'(\xi_\tau^\ell)+\frac{1}{2\alpha}\gamma_\tau^2\|h_\tau^\ell\|_*^2 & & \text{(by (31))} \\ \le & W_*(\xi_\tau^\ell)-\gamma_\tau[h_\tau^\ell]^T W_*'(\xi_\tau^\ell)+\frac{2}{\alpha}\gamma_\tau^2 L^2 & & \text{(by (29))} \\ = & W_*(\xi_\tau^\ell)-\gamma_\tau[h_\tau^\ell]^T[\widehat{x}_\tau^\ell-x_*]+\frac{2}{\alpha}\gamma_\tau^2 L^2 & & \\ = & W_*(\xi_\tau^\ell)+\frac{2}{\alpha}\gamma_\tau^2 L^2+\gamma_\tau[h_\tau^\ell]^T[x_*-\widehat{x}_\tau^\ell] & & \\ = & W_*(\xi_\tau^\ell)+\frac{2}{\alpha}\gamma_\tau^2 L^2+\gamma_\tau[f_\ell'(x_\tau^\ell)]^T[x_*-\widehat{x}_\tau^\ell]+\gamma_t\|f_\ell'(x_\tau^\ell)\|_*[\eta_\tau^\ell]^T[x_*-\widehat{x}_\tau^\ell]. & & \end{array}$$

The last term here is $\le -\|\widehat{x}_\tau^\ell-x_\tau^\ell\|$ by (9), so that

$$\begin{array}{rcl} W_*(\xi_\tau^{\ell+1}) &\le& W_*(\xi_\tau^\ell)+\frac{2}{\alpha}\gamma_\tau^2 L^2-\gamma_t\|f_\ell'(x_\tau^\ell)\|_*\|\widehat{x}_\tau^\ell-x_\tau^\ell\|+\gamma_\tau[f_\ell'(x_\tau^\ell)]^T[x_*-\widehat{x}_\tau^\ell] \\ &=& W_*(\xi_\tau^\ell)+\frac{2}{\alpha}\gamma_\tau^2 L^2-\gamma_t\|f_\ell'(x_\tau^\ell)\|_*\|\widehat{x}_\tau^\ell-x_\tau^\ell\| \\ && +\gamma_\tau[f_\ell'(x_\tau^\ell)]^T[x_*-x_\tau^\ell]+\gamma_\tau[f_\ell'(x_\tau^\ell)]^T[x_\tau^\ell-\widehat{x}_\tau^\ell]. \end{array}$$

The last term here is $\le \|f_\ell'(x_\tau^\ell)\|_*\|\widehat{x}_\tau^\ell-x_\tau^\ell\|$, whence

$$\begin{array}{rcll} W_*(\xi_\tau^{\ell+1}) &\le& W_*(\xi_\tau^\ell)+\frac{2}{\alpha}\gamma_\tau^2 L^2+\gamma_\tau[f_\ell'(x_\tau^\ell)]^T[x_*-x_\tau^\ell] & \\ &\le& W_*(\xi_\tau^\ell)+\frac{2}{\alpha}\gamma_\tau^2 L^2+\gamma_\tau[f_\ell(x_*)-f_\ell(x_\tau^\ell)] & \text{(convexity of } f_\ell) \\ &\le& W_*(\xi_\tau^\ell)+\frac{2}{\alpha}\gamma_\tau^2 L^2+\gamma_\tau[f_\ell(x_*)-f_\ell(x_\tau^1)]+\gamma_\tau[f_\ell(x_\tau^1)-f_\ell(x_\tau^\ell)]. & \end{array}$$

Since the last term is $\le 2k\beta\alpha^{-1}\gamma_\tau L^2$ by (30c), we come to

$$W_*(\xi_\tau^{\ell+1})\le W(\xi_\tau^\ell)-\gamma_\tau[f_\ell(x_\tau^1)-f_\ell(x_*)]+2(k+1)\beta\alpha^{-1}\gamma_\tau^2 L^2.$$

Adding up these inequalities for $\ell=1,\dots,k$, we conclude that

$$W_*(\xi_\tau^{k+1})\le W(\xi_\tau^1)-\gamma_\tau[f(x_\tau^1)-f(x_*)]+2k(k+1)\beta\alpha^{-1}\gamma_\tau^2 L^2.$$

Since $\xi_\tau^1 = \xi_\tau$ and $\xi_{\tau+1} = w'(W'(\xi_\tau^{k+1}))$, the latter inequality, by (32), implies that

$$W_*(\xi_{\tau+1}) \le W_*(\xi_\tau) - \gamma_\tau[f(x_\tau^1) - f(x_*)] + 2k(k+1)\beta\alpha^{-1}\gamma_\tau^2 L^2. \tag{33}$$

Summing up the resulting inequalities over τ, $t/2 \le \tau \le t$, and denoting by $\bar{t}$ the smallest value of τ in this range, we get

$$\begin{aligned}\left[\min_{\bar{t}\le\tau\le t} f(x_\tau^1) - f(x_*)\right]\sum_{\tau=\bar{t}}^{t}\gamma_\tau &\le \sum_{\tau=\bar{t}}^{t}\gamma_\tau[f(x_\tau^1) - f(x_*)] \\ &\le W_*(\xi_{\bar{t}}) - W_*(\xi_{t+1}) + 2k(k+1)\beta\alpha^{-1}L^2\sum_{\tau=\bar{t}}^{t}\gamma_\tau^2.\end{aligned} \tag{34}$$

Now, since W is the Legendre transformation of $w|_Y$ and $x_* \in X \subset Y$, we have $W_*(\xi_{t+1}) = W(\xi_{t+1}) - \xi_{t+1}^T x_* \ge -w(x_*)$, while, by construction, $\xi_{\bar{t}} = w'(y_{\bar{t}})$ for certain $y_{\bar{t}} \in Y$. It follows that $W_*(\xi_{\bar{t}}) = [w'(y_{\bar{t}})]^T y_{\bar{t}} - w(y_{\bar{t}}) - [w'(y_{\bar{t}})]^T x_*$, whence

$$W_*(\xi_{\bar{t}}) - W_*(\xi_{t+1}) \le w(x_*) - \left[w(y_{\bar{t}}) + [w'(y_{\bar{t}})]^T(x_* - y_{\bar{t}})\right] \le \Gamma(w).$$

Thus, (34) implies that

$$\min_{\bar{t}\le\tau\le t} f(x_\tau^1) - f(x_*) \le \frac{\Gamma(w) + 2k(k+1)\beta\alpha^{-1}L^2\sum_{\tau=\bar{t}}^{t}\gamma_\tau^2}{\sum_{\tau=\bar{t}}^{t}\gamma_\tau}. \tag{35}$$

At the same time, from (30c) it follows that whenever $t \ge \tau \ge t/2$, one has

$$|\widetilde{f}_\tau - f(x_\tau^1)| \le 2k^2\beta\alpha^{-1}L^2 \max_{t/2\le\tau\le t}\gamma_\tau \qquad \left[\widetilde{f}_\tau = \sum_{\ell=1}^{m} f_\ell(x_\tau^\ell)\right]$$

Taking into account the latter inequality, the inequality (30), and the rule for generating x^t, we come to (26).

The remaining statements of Theorem 5.1 are straightforward consequences of (26). □

REMARK 5.1. *The theoretical efficiency estimate of OSMD stated by Theorem* 5.1 *is not better (in fact, it is larger, by a factor $O(k\beta^{1/2})$) than the estimate stated in Theorem* 4.1 *for gMD. The advantage of the OS techniques is a matter of practical experience in several difficult application areas (e.g., training of neural nets* [Ber97] *and tomography* [Hud94]*). In this regard, the role of Theorem* 5.1 *is to make the approach theoretically legitimate.*

5.1. OS implementation of MD_1. From now on, we focus on problem (7) with objective of the form (25), and assume that X is the standard n-dimensional simplex Δ_n. Our current goal is to complete the description of the associated OS version of MD_1. The only elements still missing are the calculation of the projector

$$\pi(x) \equiv \pi_p(x) = \operatorname*{argmin}_{y\in\Delta_n}\|x - y\|_p$$

of the separator $\eta(x)$ and an explicit upper bound on the Lipschitz constant of this projector w.r.t. $\|\cdot\|_p$-norm, i.e., on the quantity

$$\beta(p) = \sup_{x,x'\in\mathbf{R}^n, x\ne x'}\frac{\|\pi_p(x) - \pi_p(x')\|_p}{\|x - x'\|_p}.$$

The required information is provided by the following result.

PROPOSITION 5.2. *Let $1 < p < \infty$. Then the following hold.*

(i) *The projector $\pi_p(x)$ is independent of p and is given componentwise by*

$$(\pi_p(x))_j = (x_j + \lambda(x))_+,\ j = 1, \dots, n \qquad (a_+ = \max[0, a]) \tag{36}$$

where $\lambda(x)$ is the unique root of the equation

$$\sum_{j=1}^{n} (x_j + \lambda)_+ = 1. \tag{37}$$

In particular, $\pi_p(x)$, for every $p > 1$, is also a $\|\cdot\|_1$-projector of $\mathbf{R}^n$ onto Δ_n:

$$\pi_p(x) \in \operatorname{Argmin}\{\|x - y\|_1 : y \in \Delta_n\}.$$

The separator $\eta_p(x)$,

$$\|\eta_p(x)\|_q \le 1,\ \eta_p^T(x)(x - y) \ge \|x - \pi_p(x)\|_p \quad \forall y \in X \qquad \left(q = \tfrac{p}{p-1}\right),$$

is readily given by $\pi_p(x)$:

$$\begin{array}{lcl} x \in X & \Rightarrow & \eta_p(x) = 0; \\ x \notin X & \Rightarrow & \eta_p(x) = [\nabla \|z\|_p]_{z = x - \pi_p(x)} = \left\{ \frac{|\delta_i|^{p-1} \operatorname{sign}(\delta_i)}{\|\delta\|_p^{p-1}} \right\}_{i=1}^{n},\ \delta = x - \pi_p(x). \end{array} \tag{38}$$

(ii) $\beta(p) \le 2$.

Proof. 0^0. Relation (38) is evident, since $\|\cdot\|_p$ is continuously differentiable outside of the origin for $p > 1$.

1^0. Let us verify first that $\pi_p(x)$ is indeed given by (36) and thus is independent of p. There is nothing to prove when $x \in \Delta_n$ (in this case the unique root of (37) is $\lambda(x) = 0$, and (36) says correctly that $\pi_p(x) = x$). Now let $x \notin \Delta_n$. It is immediately seen that $\lambda(x)$ is well defined; let y be the vector with the coordinates given by the right hand side of (36). This vector clearly belongs to Δ_n, and the vector $d = y - x$ is as follows: there exists a nonempty subset J of the index set $\{1, \dots, n\}$ such that $d_j = \lambda(x)$ for $j \in J$ and $d_j < \lambda(x)$ and $y_j = 0$ for $j \notin J$. In order to verify that y is the $\|\cdot\|_p$-projection of x onto Δ_n, it suffices to prove that if $\delta = \frac{\partial \|z\|_p}{\partial z}|_{z=d}$, then the linear form $\delta^T u$ attains its minimum over $u \in \Delta_n$ at the point y. We have

$$\delta_j = \theta |d_j|^{p-1} \operatorname{sign}(d_j),\ j = 1, \dots, n \quad (\theta > 0),$$

i.e., the same as for the vector d itself, for certain μ it holds $\delta_j = \mu$, $j \in J$ and $\delta_j < \mu$, $y_j = 0$ for $j \notin J$, so that the linear form $\delta^T u$ indeed attains its minimum over $u \in \Delta_n$ at the point y.

2^0. Now let us prove that $\beta(p) \le 2$. Observe that $\pi_p(x)$ is Lipschitz continuous (since $\pi_p(\cdot)$ is independent of p, and the $\|\cdot\|_2$-projector onto a closed convex set is Lipschitz continuous, with constant 1, w.r.t. $\|\cdot\|_2$).

$2^0.1$. Let $J(x) = \{j \mid x_j + \lambda(x) \ge 0\}$, and let $k(x)$ be the cardinality of $J(x)$. Since $\lambda(x)$ solves (37), we have $k(x) \ge 1$ and $\lambda(x) = \frac{1}{k(x)}[1 - \sum_{j \in J(x)} x_j]$. Denoting by $e(x)$ the characteristic vector of the set $J(x)$ and by $E(x)$ the matrix $\operatorname{Diag}(e(x))$, we therefore get

$$\pi_p(x) = E(x)x + \frac{1}{k(x)} e(x) - \frac{1}{k(x)} e(x) e^T(x) x. \tag{39}$$

Let $\mathcal{J}$ be the set of all nonempty subsets of the index set $\{1,\dots,n\}$, and let $X[J] = \{x \mid J(x) = J\}$ for $J \in \mathcal{J}$. From (39) it follows that for every $J \in \mathcal{J}$ we have

$$\begin{array}{rcl} & & x,y \in X[J] \Rightarrow \\ \|\pi_p(x)-\pi_p(y)\|_p & \le & \|E(x)(x-y)\|_p + \frac{1}{k(x)}\|e(x)e^T(x)(x-y)\|_p \\ & \le & \|x-y\|_p + \frac{1}{k(x)}\|e(x)\|_p\|e(x)\|_{\frac{p}{p-1}}\|x-y\|_p \\ & = & 2\|x-y\|_p. \end{array} \tag{40}$$

2^0.2. Let

$$\mathcal{X} = \{x \in \mathbf{R}^n : \exists (J \in \mathcal{J}, j \le n) : \mathrm{Card}(J)x_j = \sum_{j' \in J} x_{j'} - 1\}.$$

Note that $\mathcal{X}$ is the union of finitely many hyperplanes. We claim that if $x, y \in \mathbf{R}^n$ are such that the segment $[x,y]$ does not intersect $\mathcal{X}$, then $J(x) = J(y)$ and, consequently (see (40)),

$$\|\pi_p(x)-\pi_p(y)\|_p \le 2\|x-y\|_p. \tag{41}$$

Indeed, assume that $J(x) \ne J(y)$, or, which is the same, the sets $\{j : x_j \ge -\lambda(x)\}$ and $\{j : y_j \ge -\lambda(y)\}$ are distinct from each other. Since $\lambda(\cdot)$ clearly is continuous, it follows that on the segment $[x,y]$ there exists a point $\bar{x}$ such that one of the coordinates of the point equals to $-\lambda(\bar{x})$, i.e., to $\frac{1}{k(\bar{x})}[\sum_{j' \in J(\bar{x})} \bar{x}_{j'} - 1]$. In other words, $\bar{x} \in \mathcal{X}$, which contradicts the assumption.

2^0.3. Now let $y, y' \in \mathbf{R}^n \backslash \mathcal{X}$. Since $\mathcal{X}$ is a union of finitely many hyperplanes, the segment $[y,y']$ can be partitioned into subsequent segments $[y,y_1]$, $[y_1,y_2]$,..., $[y_s,y']$ in such a way that the interior of every segment of the partition does not intersect $\mathcal{X}$. By the result of 2^0.2, $\pi_p(\cdot)$ is Lipschitz continuous with constant 2 w.r.t. $\|\cdot\|_p$ on the interiors of the above segments. Since $\pi_p(\cdot)$, as we just mentioned, is continuous, it follows that

$$\|\pi_p(y)-\pi_p(y')\|_p \le 2\|y-y'\|_p.$$

The latter relation holds true for all pairs $y, y' \in \mathbf{R}^n \backslash \mathcal{X}$, i.e., for all pairs from a set which is dense in $\mathbf{R}^n$; since $\pi_p(\cdot)$ is continuous, this relation in fact holds for all y, y'. □

REMARK 5.2. *The upper bound* 2 *on* $\beta(p)$ *cannot be improved, unless one restricts the range of values of* p *and/or values of* n. *Indeed, the* $\|\cdot\|_p$*-distance from the origin to a vertex of* Δ_n *is* 1, *while the* $\|\cdot\|_p$*-distance between the* $\|\cdot\|_p$*-projections of these points onto* Δ_n, *i.e., the* $\|\cdot\|_p$*-distance from a vertex to the barycenter of* Δ_n, *is* $\left(\frac{n-1}{n^p} + \left(\frac{n-1}{n}\right)^p\right)^{1/p}$; *when* n *is large and* p *is close to* 1, *the latter quantity is close to* 2.

We see that to project onto Δ_n is easy: computation of $\pi(x)$ requires, basically, the same effort as ordering the coordinates of x, which can be done in time $O(n \ln n)$.

6. Implementation and testing. In this section, we present results of the MD method as applied to the PET image reconstruction problem based on several sets of simulated and real clinical data. We compare the results obtained by OSMD_1 and MD_1. In addition, we compare the results of MD to those of the usual SD method.

6.1. Implementation of the algorithms. In our experiments, we have worked with several sets of tomography data. Each data set gives rise to a particular optimization problem of the form of (8) which was solved by the MD scheme (in both the usual and the OS versions). The setup for MD was

$$Y = \{x \mid \|x\|_p \le 1\} \ [\supset \Delta_n], \quad \|\cdot\| = \|\cdot\|_1, \quad w(x) = \frac{1}{2}\|x\|_p^2, \quad p = p(n) = 1 + \frac{1}{\ln n}.$$

This setup differs from (15) – (20) by setting $\|\cdot\| = \|\cdot\|_1$ instead of $\|\cdot\| = \|\cdot\|_{p(n)}$; with the above $p(n)$, this modification does not affect the theoretical efficiency estimate of the algorithm.

The indicated setup defines the algorithm up to the stepsize policy. The latter for the "no ordered subsets" version MD of the method was chosen as (cf. (22))

$$\gamma_t = \frac{C}{\|f'(x_\tau)\|_\infty \sqrt{\ln n}\sqrt{t}}$$

with $C = 0.03$. (This value of the stepsize factor C was found to be the best one in our preliminary experiments and was never changed afterwards.)

The OS version OSMD of the method uses 24-component representation (25) of the objective, the components being partial sums of the terms in the sum (8), with $m/24$ subsequent terms in every one of the partial sums. The stepsizes here were chosen according to the rule (cf. (27))

$$\gamma_t = \frac{C}{24 L_t \sqrt{t}\sqrt{\ln n}},$$

where L_t is a current guess for the $\|\cdot\|_1$-Lipschitz constant of the objective; in our implementation, this guess, starting with the second outer iteration, was defined as $\sum_{1\le\ell\le 24} \|f'_\ell(x^\ell_{t-1})\|_\infty$. The stepsize factor C in OSMD was set to 0.3.

In our experiments we have used, as a "reference point," the standard SD method (23) (in the usual "no subsets") version with the "theoretical" stepsize policy

$$\gamma_t = \frac{C}{\|f'(x_t)\|_2 \sqrt{t}}.$$

The stepsize factor C was tuned to get the best reconstruction possible; the resulting "optimal value" turned out to be 0.006.

The starting point x_0 in all our runs was the barycenter of the simplex Δ_n.

Measuring quality of reconstructions. In medical imaging, the standard way to evaluate the quality of a reconstruction algorithm is to apply the algorithm to simulated data and to check how the resulting pictures reproduce important—for a particular application—elements of the true image. (In tomography, these elements could be, e.g., small areas with high density of the tracer mimicking tumors.) In what follows we combine this, basically qualitative, way of evaluation with a quantitative one, where the quality of the approximate solution x^t to (8) yielded after t steps of the method is measured by the quantity $\varepsilon_t = f(x^t) - \min_{\Delta_n} f$. Note that this quantity is not "observable" (since the true optimal value $f_* = \min_{\Delta_n} f$ is unknown). We can, however, easily compute a *lower bound* on f_*. Assume, e.g., that we have run a "no subset" version of the method and in the course of computations have computed the values $f(x_t)$ and subgradients $f'(x_t)$ of the objective at N search points x_t, $1 \le t \le N$. Then we can build the standard piecewise-linear minorant $f^N(\cdot)$ of our objective:

$$f^N(x) = \max_{1\le t\le N} \left[[f(x_t) - x_t^T f'(x_t)] + x^T f'(x_t)\right] \le f(x).$$

The quantity $f_*^N \equiv \min_{x\in\Delta_n} f^N(x)$ clearly is a lower bound on f_*, so that the "observable" quantities

$$\widehat{\varepsilon}_t = f(x_t) - f_*^N,\ 1 \le t \le N,$$

are upper bounds on the actual inaccuracies ε_t. In our experiments, the bound f_*^N was computed at the post-optimization phase according to the relation

$$f_*^N \equiv \min_{x\in\Delta_n} \max_{t\le N} \left[[f(x_t) - x_t^T f'(x_t)] + x^T f'(x_t)\right] = \max_{\lambda\in\Delta_N} \phi(\lambda),$$
$$\phi(\lambda) \equiv \min_{x\in\Delta_n} \sum_{t=1}^N \lambda_t \left[\underbrace{[f(x_t) - x_t^T f'(x_t)]}_{d_t} + x^T f'(x_t)\right] = \sum_{t=1}^N \lambda_t d_t + \min_{j\le n}\left[\sum_{t=1}^N \lambda_t f'(x_t)\right]_j,$$

which reduces the computation of f_*^N to maximizing a concave function $\phi(\lambda)$ of N variables. In our experiments, the total number of iterations N was just 10, and there was no difficulty in minimizing ϕ.

In the OS version of the method, the policy for bounding f_* from below was similar: here after N outer iterations we know the values and the subgradients of the components f_ℓ, $\ell = 1,\dots,k$, in decomposition (25) along the points x_t^ℓ, $t = 1,\dots,N$. This allows us to build a piecewise linear minorant

$$f^N(x) = \sum_{\ell=1}^k \max_{t=1,\dots,N} \left[[f_\ell(x_t^\ell) - [x_t^\ell]^T f_\ell'(x_t^\ell)] + x^T f_\ell'(x_t^\ell)\right]$$

of the objective and to use, as the lower bound on f_*, the quantity

$$f_*^N \equiv \min_{x\in\Delta_n} f^N(x) = \max_{\mu}\left\{\psi(\mu) : \mu = \{\mu_{t\ell}\} \ge 0, \sum_t \mu_{t\ell} = 1,\ \ell = 1,\dots,k\right\},$$
$$\psi(\mu) \equiv \min_{x\in\Delta_n} \sum_{t,\ell} \mu_{t\ell}\left[\underbrace{[f_\ell(x_t^\ell) - [x_t^\ell]^T f_\ell'(x_t^\ell)]}_{d_{t\ell}} + x^T f_\ell'(x_t^\ell)\right] = \sum_{t,\ell} \mu_{t\ell} d_{t\ell} + \min_j \left[\sum_{t,\ell} \mu_{t\ell} f_\ell'(x_t^\ell)\right]_j.$$

6.2. Results. We tested the algorithms on five sets of tomography data; the first four are simulated scans of *phantoms* (artificial bodies) obtained from the Eidolon simulator [Zai98], [Zai99] of the PRT-1 PET-scanner. The phantoms (Cylinder, Utah, Spheres, Jaszczak) are 3D cylinders with piecewise constant density of the tracer; they are commonly used in tomography to test the effectiveness of scanners and reconstruction methods (for more details, see [Thi99]). The fifth data set Brain is obtained from the GE Advance PET-scanner in an actual brain study.

All experiments were carried out on the Intel Marlinspike Windows NT Workstation (500 MHz 1Mb Cache Intel Pentium III Xeon processor, 2GB RAM). A single outer iteration of OSMD takes nearly the same time as a single iteration of MD, namely, approximately two minutes in each of the four "phantom" tests ($n = 515,871, m = 3,170,304$), and approximately 90 minutes in the `Brain` test ($n = 2,763,635, m \approx 25,000,000$). About 95% of the running time is used to compute the value and the gradient of the objective.

TABLE 1
Objective values along iterations (for OSMD, $x_t = x_t^1$).

Itr#	Cylinder $f(x_t) \times 10^{-8}$		Utah $f(x_t) \times 10^{-8}$		Spheres $f(x_t) \times 10^{-7}$		Jazszak $f(x_t) \times 10^{-7}$		Brain $f(x_t) \times 10^{-9}$	
	MD	OSMD	MD	OSMD	MD	OSMD	MD	OSMD	MD	OSMD
1	−2.382	−2.382	−2.549	−2.549	−4.295	−4.295	−5.021	−5.021	−1.463	−1.463
2	−2.648	−2.725	−2.807	−2.902	−4.767	−5.132	−5.643	−5.908	−1.725	−1.848
3	−2.708	−2.732	−2.890	−2.926	−5.079	−5.191	−5.867	−5.968	−1.867	−2.001
4	−2.732	−2.732	−2.929	−2.939	−5.189	−5.200	−5.970	−6.000	−1.951	−2.012
5	−2.723	−2.734	−2.917	−2.938	−5.168	−5.212	−5.950	−5.988	−1.987	−2.015
6	−2.738	−2.738	−2.943	−2.937	−5.230	−5.216	−6.001	−6.005	−1.978	−2.015
7	−2.727	−2.740	−2.923	−2.936	−5.181	−5.205	−5.967	−5.991	−1.997	−2.016
8	−2.740	−2.742	−2.942	−2.936	−5.227	−5.218	−6.007	−6.005	−2.008	−2.016
9	−2.731	−2.737	−2.925	−2.937	−5.189	−5.212	−5.974	−5.994	−1.999	−2.016
10	−2.741	−2.741	−2.941	−2.937	−5.225	−5.205	−6.030	−6.002	−2.009	−2.016
Lower bound	-2.754		-2.966		-5.283		-6.093		-2.050	

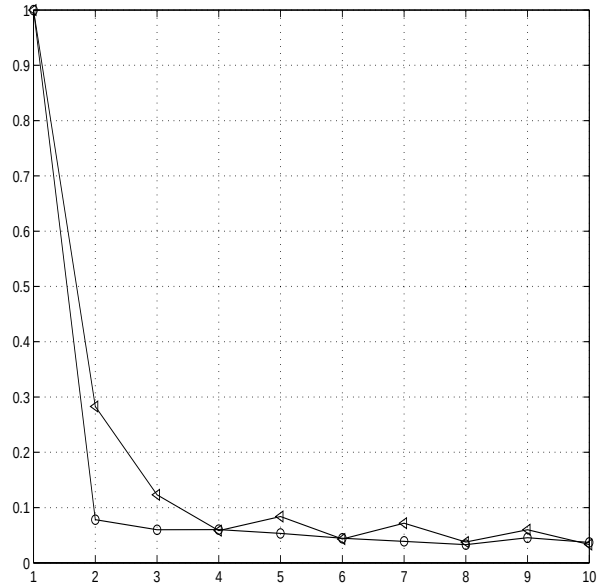


FIG. 1. *Cylinder, progress in accuracy: plot of* $\theta(t) = \frac{f(x_t)-f_*^{10}}{f(x_1)-f_*^{10}}$ $\left[\geq \frac{f(x_t)-f_*}{f(x_1)-f_*}\right]$. $\triangle$ = *MD*; $\circ$ = *OSMD*.

Our numerical results are summarized in Table 1.

Note that in OSMD there is no necessity to compute the true values of the objective along the iterates x_t^ℓ, and an attempt to compute these values would increase the execution time by factor k. For the sake of this paper we, however, did compute the values $f(x_t^1)$.

A more detailed description of the data and the results is as follows.

Cylinder ($n = 515,871, m = 3,170,304$): This phantom is a cylinder with a uniform density of the tracer. Figure 1 displays the "progress in accuracy" in the experiment.

Utah ($n = 515,871, m = 3,170,304$): This phantom (see Figure 2) is a pair of coaxial cylinders with two vertical tubes in the inner cylinder, and the density of the tracer is high between the cylinders and in one of the tubes, is low in the other tube, and is moderate within the inner cylinder outside the tubes. The phantom allows us to test the ability of an algorithm to reconstruct the borders between areas with

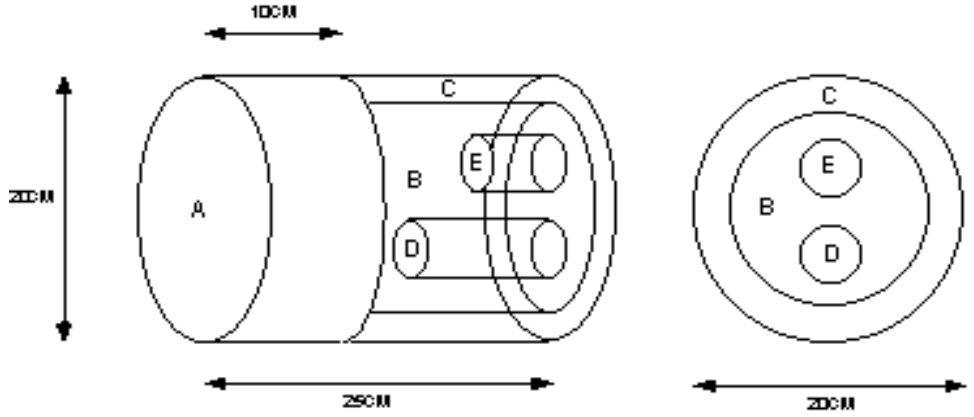


FIG. 2. *The Utah phantom.*

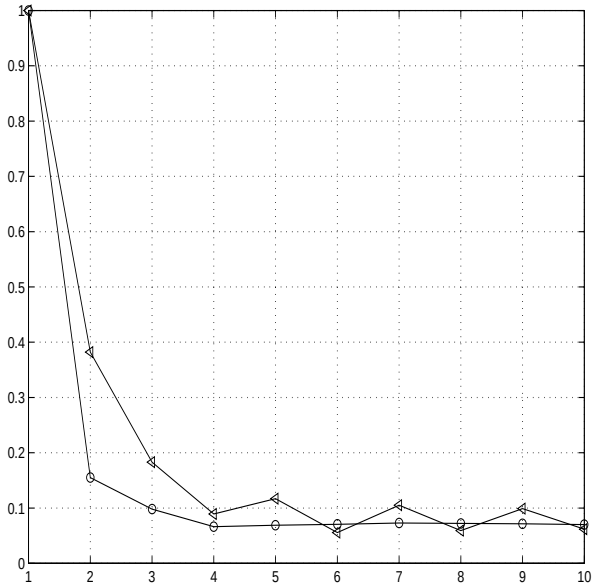


FIG. 3. *Utah, progress in accuracy: plot of* $\theta(t) = \frac{f(x_t)-f_*^{10}}{f(x_1)-f_*^{10}}$ $\left[\geq \frac{f(x_t)-f_*}{f(x_1)-f_*}\right]$. $\triangle$ = *MD*; $\circ$ = *OSMD.*

different densities of the tracer and the ratios of these densities.

Figure 3 displays the "progress in accuracy."

In clinical applications, the yield of a reconstruction algorithm is a collection of *slices*—pictures of different 2D cross-sections of the resulting 3D image. To give an idea of the quality of our reconstructions, Figure 4 represents their slices (the cross-sections of the outer cylinder by a plane orthogonal to its axis); in all our pictures, white corresponds to high and black to low density of the tracer.

Spheres ($n = 515,871, m = 3,170,304$)*:* This phantom is a cylinder containing six spheres of different radii centered at the mid-slice of the cylinder. The density of the tracer is high within the spheres and low outside of them. The mid-slice of the phantom is shown on Figure 5.

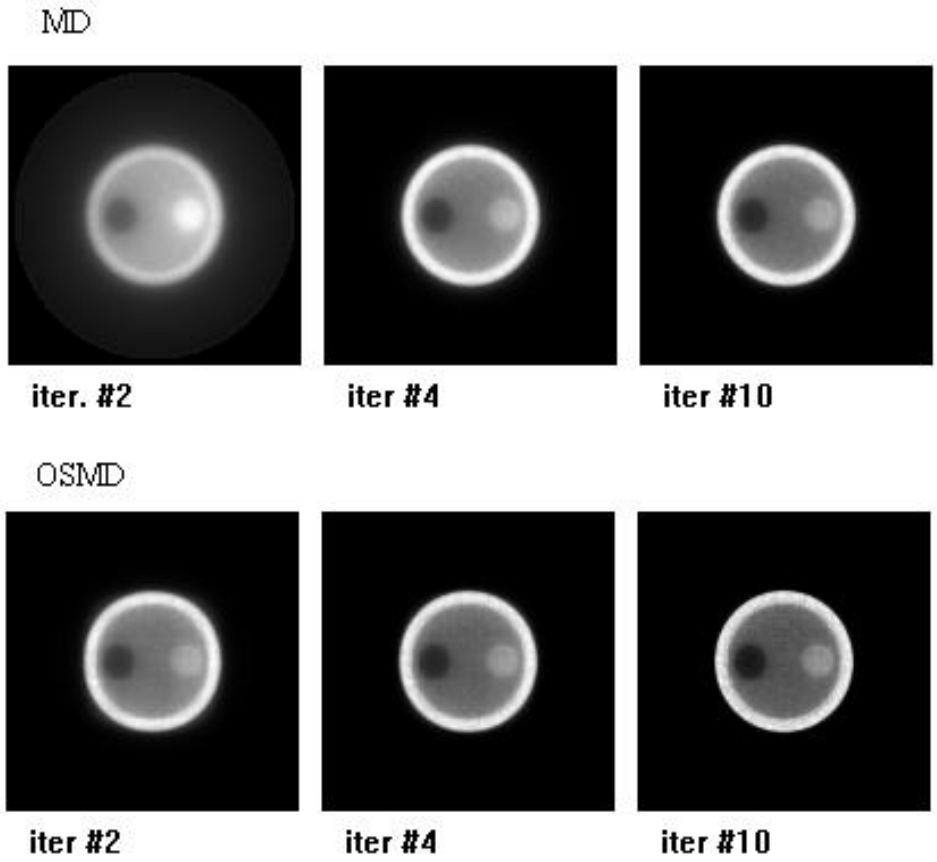


FIG. 4. *Utah, near-top slice of the reconstruction.*

The phantom is used to test tumor detection capability, mainly for torso studies. Figure 6 displays the "progress in accuracy."

The mid-slices of our 3D reconstructions are shown on Figure 7. The Spheres experiment clearly demonstrates the advantages of the $\|\cdot\|_1$-MD as compared to the usual SD. The best progress in accuracy we were able to get with SD was to reduce in 10 iterations the initial residual in the objective by factor 5.26, which is 3.5 times worse than the similar factor (18.51) for MD. What is much more dangerous from the clinical viewpoint is that the reconstructions given by SD can be heavily affected by artifacts, as can be seen from Figure 8.

Jaszczak ($n = 515,871, m = 3,170,304$): This phantom is a cylinder containing a number of vertical tubes of different cross-sections. The density of the tracer is high outside of the tubes and is zero inside them. The mid-slice of the phantom is shown on Figure 9.

The number and the sizes of tubes "recognized" by a reconstruction algorithm allow us to quantify the resolution of the algorithm.

Figure 10 displays the "progress in accuracy."

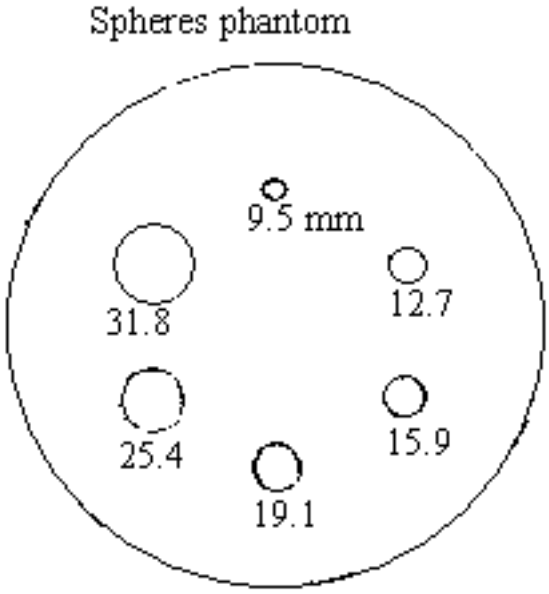


FIG. 5. *Mid-slice of the Spheres phantom.*

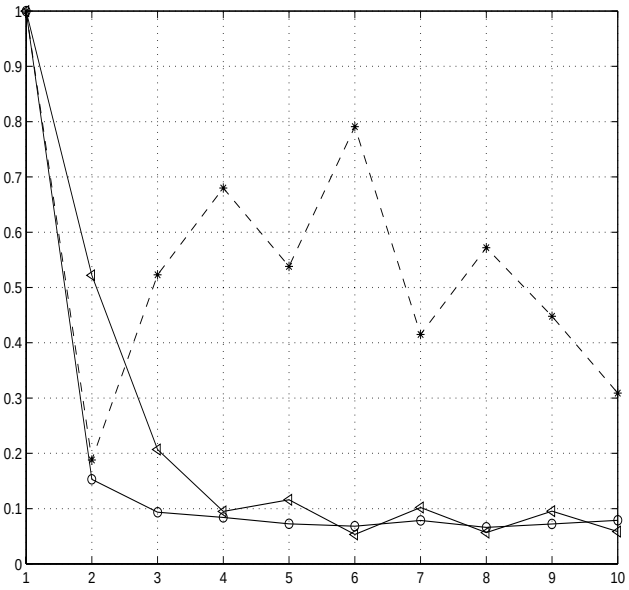


FIG. 6. *Spheres, progress in accuracy: plot of* $\theta(t) = \frac{f(x_t)-f_*^{10}}{f(x_1)-f_*^{10}}$ $\left[\geq \frac{f(x_t)-f_*}{f(x_1)-f_*}\right]$. $\Delta = MD$; $\circ = OSMD$; $* = SD$.

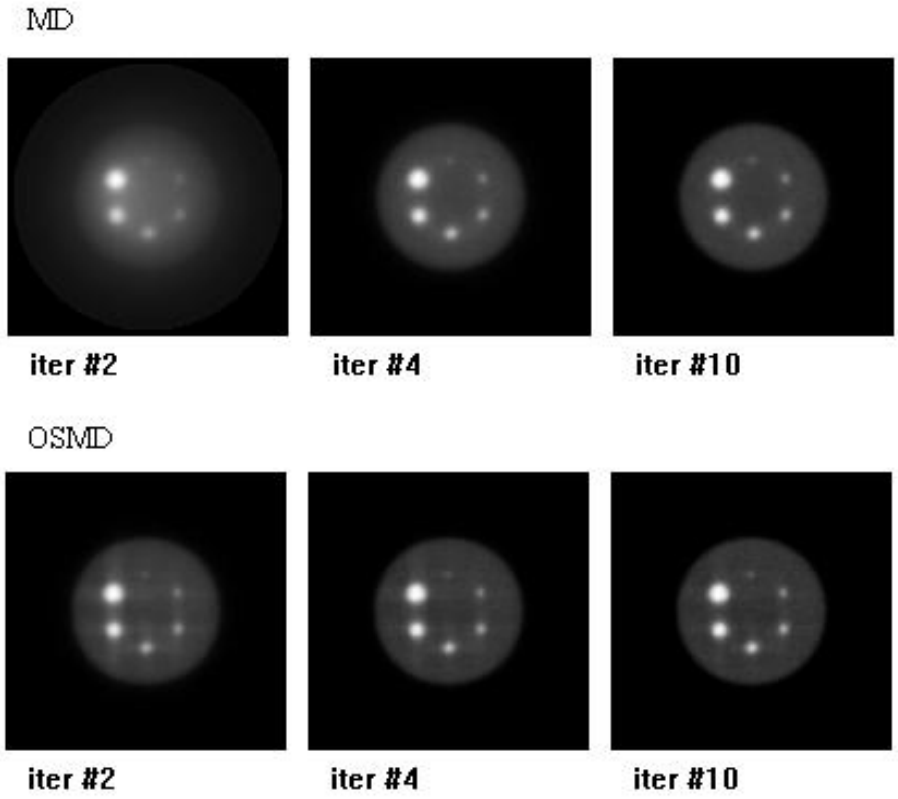


FIG. 7. *Spheres, mid-slice of the reconstructions.*

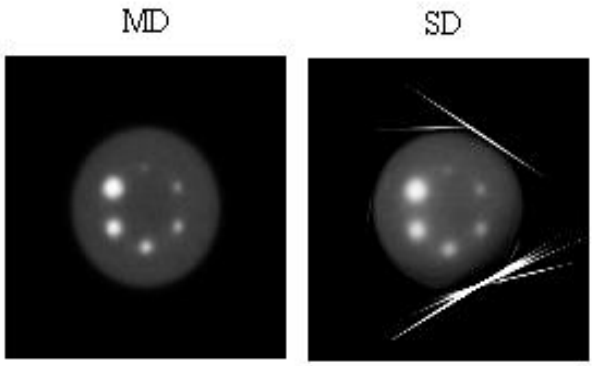


FIG. 8. *Spheres, mid-slice of the SD reconstruction after* 10 *iterations.*

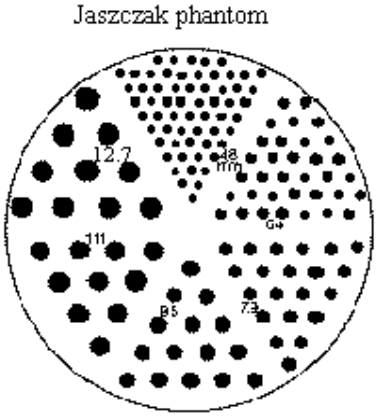


FIG. 9. *Mid-slice of the Jaszczak phantom.*

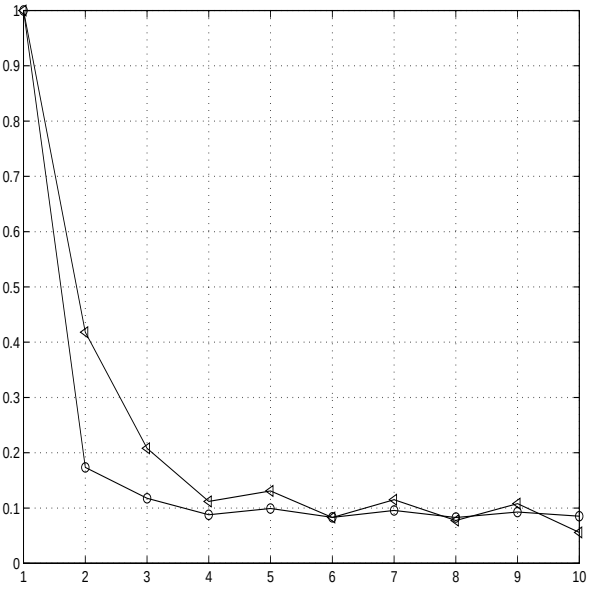


FIG. 10. *Jaszczak, progress in accuracy: plot of* $\theta(t) = \frac{f(x_t)-f_*^{10}}{f(x_1)-f_*^{10}}$ $\left[\geq \frac{f(x_t)-f_*}{f(x_1)-f_*}\right]$. $\triangle$ = *MD*; $\circ$ = *OSMD.*

The mid-slices of our 3D reconstructions are shown on Figure 11.

The Jaszczak experiment clearly demonstrates the advantages of OSMD as compared to MD. We see that the quality of the image after just 2 outer iterations of OSMD is at leas as good as the one obtained after 4 iterations of MD. Likewise, 4 iterations of OSMD result in an image comparable to the one obtained by MD in 10 iterations.

Brain ($n = 2,763,635, m \approx 25,000,000$)*:* This data is an actual clinical brain study of a patient with Alzheimer's disease.

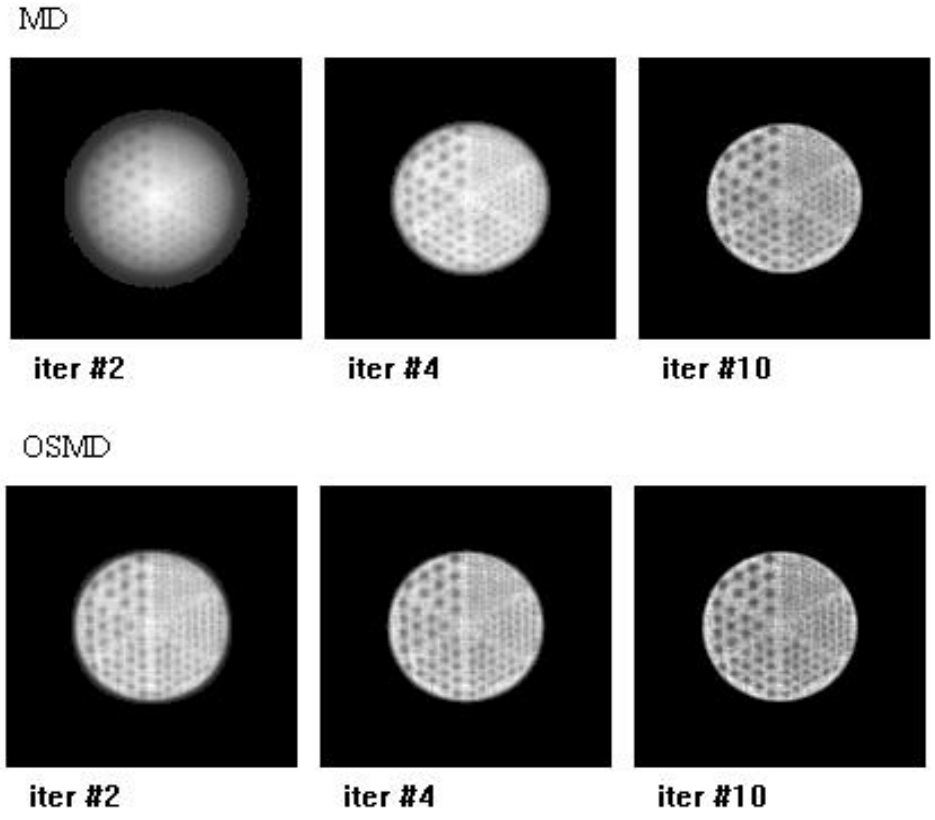


FIG. 11. *Jaszczak, mid-slice of the reconstructions.*

Figure 12 displays the "progress in accuracy."

The mid-slices of our 3D reconstructions are shown on Figure 13.

The Brain experiment again demonstrates the advantages of OSMD as compared to MD. Indeed, OSMD produced in 4 iterations an image which is as good as the one produced after 10 iterations of MD.

The quality of our reconstructions compares favorably with the one given by the commercially used algorithms (based on FBP). As compared to the "golden standard" of the new generation of 3D imaging algorithms—the so-called OSEM (ordered subset expectation maximization) algorithm, OSMD is highly competitive both in image quality and computational effort. Moreover, the OSMD algorithm possesses a solid theoretical background (guaranteed efficiency estimates), which is not the case for OSEM.

7. Conclusions. The outlined results of our research suggest the following conclusions:

1. Simple gradient-descent type optimization techniques, which seem to be the only option when solving really large-scale (hundreds, thousands, and millions of variables) convex optimization problems, can be quite successful and can yield a solution of a satisfactory quality in few iterations.

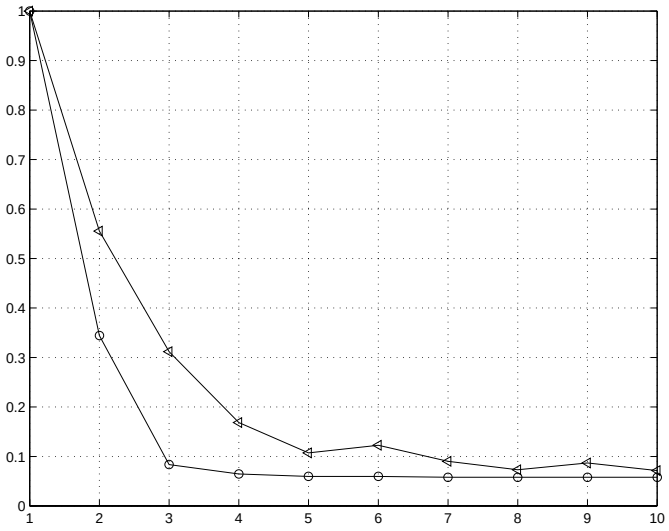


FIG. 12. *Brain, progress in accuracy: plot of* $\theta(t) = \frac{f(x_t)-f_*^{10}}{f(x_1)-f_*^{10}} \left[\geq \frac{f(x_t)-f_*}{f(x_1)-f_*}\right]$. $\triangle$ = *MD*; $\circ$ = *OSMD*.

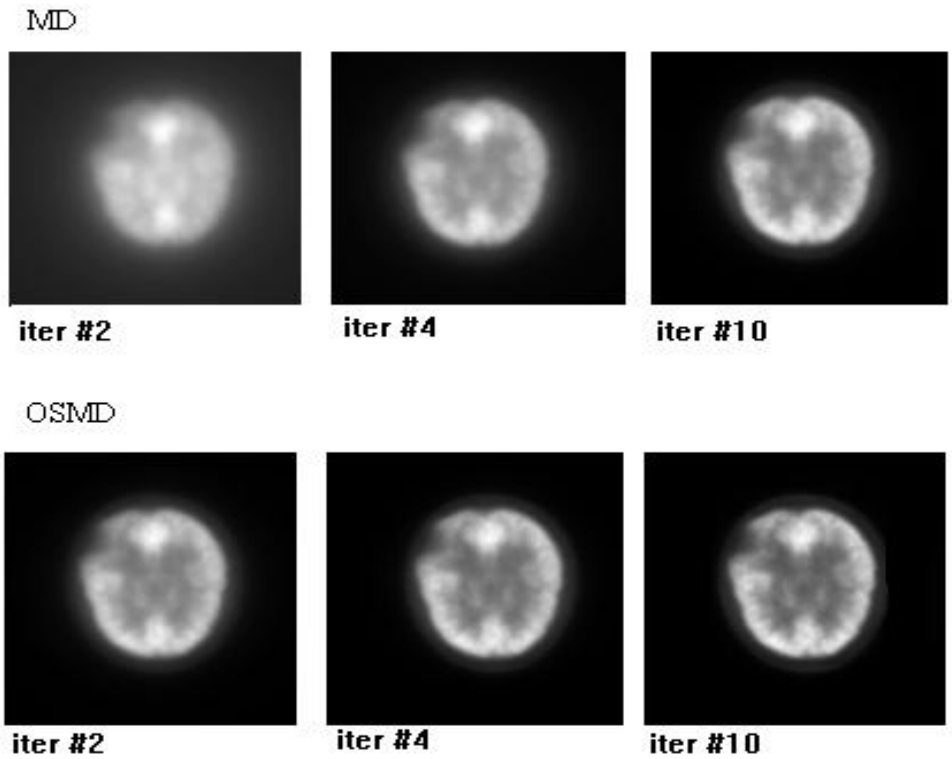


FIG. 13. *Brain, near-mid slice of the reconstructions. (The top-left missing part is the area affected by Alzheimer's disease.)*

2. When implementing gradient-type optimization techniques, one should try to adjust the method to the "geometry" of the problem. For such an adjustment, the general MD scheme can be used.
3. Implementing gradient-descent–type techniques in an "incremental gradient" fashion can accelerate significantly the solution process.

8. Appendix 1: Strong convexity of $\frac{1}{2}\|\cdot\|_p^2$. Here we reproduce the proof of the following known fact (see, e.g., [Nem78]).

LEMMA 8.1. *Let $1 < p \le 2$, and let $w(x) = \frac{1}{2}\|x\|_p^2 : \mathbf{R}^n \to \mathbf{R}$. Then the function w is α-strongly convex w.r.t. the norm $\|\cdot\|_p$, with*

$$\alpha = p - 1. \tag{42}$$

Proof. It is known [RW98, Propositions 12.54], 12.60 that the fact that a continuously differentiable convex function $v : \mathbf{R}^n \to \mathbf{R}$ is α-strongly convex on $\mathbf{R}^n$ w.r.t. a norm $\|\cdot\|$ is equivalent to the fact that the Legendre transformation

$$V(\xi) = \max_{x \in \mathbf{R}^n} [\xi^T x - v(x)]$$

of v is continuously differentiable and satisfies the relation

$$V(\xi + \eta) \le V(\xi) + \eta^T \nabla V(\xi) + \frac{1}{2\alpha}\|\eta\|_*^2 \quad \forall \xi, \eta \in \mathbf{R}^n, \tag{43}$$

where $\|\cdot\|_*$ is the norm conjugate to $\|\cdot\|$. In our case, $\|\cdot\| = \|\cdot\|_p$ and $V(\xi) = \frac{1}{2}\|\xi\|_q^2$, $q = p/(p-1) \ge 2$, so that V is twice continuously differentiable outside of the origin (and, of course, is convex); therefore, in order to verify that (43) is satisfied with $\alpha = p - 1$, it suffices to prove that

$$\eta^T \nabla^2 V(\xi) \eta \le \frac{1}{p-1}\|\eta\|_q^2 \tag{44}$$

for every $\xi \ne 0$. By homogeneity, $\nabla^2 V(t\xi) = \nabla^2 V(\xi)$, $t > 0$, so that when proving (44), we may assume that $\|\xi\|_q = 1$. We now have

$$\begin{aligned}
\eta^T \nabla V(\xi) &= \|\xi\|_q^{2-q} \sum_{i=1}^n |\xi|_i^{q-1} \operatorname{sign}(\xi_i)\eta_i, \\
\eta^T \nabla^2 V(\xi)\eta &= (2-q)\|\xi\|_q^{2-2q} \left(\sum_{i=1}^n |\xi|_i^{q-1} \operatorname{sign}(\xi_i)\eta_i \right)^2 \\
&\quad + (q-1)\|\xi\|_q^{2-q} \sum_{i=1}^n |\xi_i|^{q-2}\eta_i^2 \\
&\le (q-1) \sum_{i=1}^n |\xi_i|^{q-2}\eta_i^2 \qquad \text{(since } q \ge 2, \|\xi\|_q = 1\text{)} \\
&\le (q-1) \left(\sum_i |\xi_i|^q \right)^{\frac{q-2}{q}} \left(\sum_i |\eta|_i^q \right)^{\frac{2}{q}} \qquad \text{(Hölder's inequality)} \\
&\le (q-1)\|\eta\|_q^2,
\end{aligned}$$

so that (44) is satisfied, due to $q - 1 = \frac{1}{p-1}$. ∎

9. Appendix 2: Proof of Proposition 4.2.

Proof. 1^0. W.l.o.g., we can assume that n is a power of 2: $n = 2^k$. It is known that there exists an orthogonal basis $u^1, \dots, u^m$ in $\mathbf{R}^m$, $m = 2^{k-1}$, such that $|u^\ell_j| = 1 \ \forall\ \ell, j = 1, \dots, m$. Let $e^\ell = \begin{pmatrix} u^\ell \\ -u^\ell \end{pmatrix} \in \mathbf{R}^{2m} = \mathbf{R}^n$, $\ell = 1, \dots, m$. Note that

1^0.A. $\|e^\ell\|_2^2 = n$, $\ell = 1, \dots, m$;

1^0.B. $[e^\ell]^T e^{\ell'} = 0$, $1 \le \ell < \ell' \le m$.

1^0.C. $\sum_{t=1}^n e^\ell_t = 0$, $\ell = 1, \dots, m$.

1^0.D. For every linear combination $e[\lambda] = \sum_{\ell=1}^m \lambda_\ell e^\ell$ one has $e_i[\lambda] = -e_{m+i}[\lambda]$, $i = 1, \dots, m$, whence

$$\|e[\lambda]\|_\infty = \max_{i \le n} e_i[\lambda] = -\min_{i \le n} e_i[\lambda].$$

2^0. Let $\delta > 0$, $1 < k \le m$, and let $\mathcal{B}$ be a method for solving problems from $\mathcal{F} = \mathcal{F}(L, n)$. Let us set

$$\varepsilon(\mathcal{B}, k) = \sup_{f \in \mathcal{F}} \left[f(x^{k-1}(\mathcal{B}, f)) - \min_{\Delta_n} f \right].$$

We are about to prove that

$$\varepsilon(\mathcal{B}, k) \ge \frac{L}{\sqrt{k}}. \tag{45}$$

Note that this inequality immediately implies the desired lower bound on the information-based complexity of $\mathcal{F}$.

From the viewpoint of the behavior of $\mathcal{B}$ at the first $k-1$ steps (which is the only issue we are interested in when proving (45)), we change nothing when assuming that $\mathcal{B}$, as applied to a problem from $\mathcal{F}$, performs exactly k steps; the search points generated by the method at the first $k-1$ steps are as given by the search rules specifying the method, and the last search point x_k is the kth approximate solution generated by $\mathcal{B}$ as applied to the problem. Thus, from now on we assume that the point $x^{k-1}(\mathcal{B}, f)$ in (45) is the kst search point generated by $\mathcal{B}$ as applied to f.

3^0. To prove (45), we intend to construct a "difficult" for $\mathcal{B}$ objective f as the pointwise maximum of k linear functions with orthogonal descent directions chosen from the set $\{\pm\ell^1, \dots, \pm\ell^m\}$. These linear functions will be successively constructed according to the adversary principle, i.e., when $\mathcal{B}$ requires evaluation at search point x_i, the ith linear function is defined such that little progress is achieved while consistency with previous information is maintained. The construction is as follows. Let x_1 be the first search point of the method (this point is problem-independent), let

$$\ell_1 \in \operatorname*{Argmax}_{1 \le \ell \le k} |x_1^T e^\ell|, \quad \sigma_1 = \operatorname{sign}(x_1^T e^{\ell_1}) \quad \left[\operatorname{sign}(s) = \begin{cases} 1, & s \ge 0 \\ -1, & s < 0 \end{cases}\right],$$
$$f^1(x) = L\sigma_1 x^T e^{\ell_1} - \delta.$$

Suppose we have already defined $x_1, \dots, x_p$, $\ell_1, \dots, \ell_p$, $f^1(\cdot), \dots, f^p(\cdot)$, $\sigma_1, \dots, \sigma_p \in \{-1; 1\}$ in such a way that

(a_p) $1 \le \ell_i \le k$ and the indices $\ell_1, \dots, \ell_p$ are distinct from each other;

(b_p) $f^i(x) = \max_{j=1,\dots,i}[L\sigma_j x^T e^{\ell_j} - j\delta]$, $i = 1, \dots, p$;

(c_p) $x_1, \dots, x_i$ is the initial i-element segment of the *trajectory* (the sequence of search points) of $\mathcal{B}$ as applied to $f^i(\cdot)$;

(d_p) $\sigma_i x_i^T e^{\ell_i} = \max\{|x_i^T e^\ell| \mid \ell \in \{1,\dots,k\}\backslash\{\ell_1,\dots,\ell_{i-1}\}\}$, $i = 1,\dots,p$.
Note that with our initialization conditions (a_1)–(d_1) do hold.

In the case of $p < k$, let us extend the collection we have built to a similar collection of $(p+1)$-element tuples; to this end we define x_{p+1} as the $(p+1)$th search point of $\mathcal{B}$ as applied to $f^p(\cdot)$, ℓ_{p+1} as the index from the set $I^p = \{1,\dots,k\}\backslash\{\ell_1,\dots,\ell_p\}$ which maximizes the quantities $x_{p+1}^T e^\ell$ over $\ell \in I^p$, and σ_{p+1} as $\mathrm{sign}(x_{p+1}^T e^{\ell_{p+1}})$, and finally set

$$f^{p+1}(x) = \max\{f^p(x), L\sigma_{p+1} x^T e^{\ell_{p+1}} - (p+1)\delta\}.$$

It is easily seen that when $1 \le i \le j \le p+1$, one has $f^j(x) = f^i(x)$ in a neighborhood of x_i; with this observation, (a_{p+1})–(d_{p+1}) immediately follow from (a_p)–(d_p) and our construction.

After k steps of the aforementioned construction, we get a function

$$f(x) \equiv f^k(x) = \max_{1\le i\le k}[L\sigma_i x^T e^{\ell_i} - i\delta]$$

such that the trajectory of $\mathcal{B}$ on f is $x_1, .., x_k$, so that x_k is the result of $\mathcal{B}$ as applied to f. Observe that $f \in \mathcal{F}(L,n)$, due to $\|e^\ell\|_\infty = 1$. In view of (d_p), we have

$$f(x_k) \ge -k\delta. \tag{46}$$

On the other hand, let us bound from above the minimum value of f over Δ_n. We have

$$f(x) = \max_{i=1,\dots,k}[L\sigma_i x^T e^{\ell_i} - i\delta] \le g(x) \equiv \max_{i=1,\dots,k} L\sigma_i x^T e^{\ell_i}$$

and therefore

$$\begin{aligned}
\min_{x\in\Delta_n} f(x) &\le \min_{x\in\Delta_n} g(x) = L \min_{x\in\Delta_n} \max_{i\le k} x^T[\sigma_i e^{\ell_i}] \\
&= L \min_{x\in\Delta_n} \max_{\lambda\in\Delta_k} x^T \underbrace{\left[\sum_{i=1}^k \lambda_i \sigma_i e^{\ell_i}\right]}_{\widetilde{e}[\lambda]} \\
&= L \max_{\lambda\in\Delta_k} \min_{x\in\Delta_n} x^T\widetilde{e}[\lambda] = L \max_{\lambda\in\Delta_k} \min_{i=1,\dots,n} \widetilde{e}_i[\lambda] \\
&= L \max_{\lambda\in\Delta_k} [-\|\widetilde{e}_i[\lambda]\|_\infty] && \text{(see } 1^0\text{.D)} \\
&= -L \min_{\lambda\in\Delta_k} \|\widetilde{e}[\lambda]\|_\infty \le -Ln^{-1/2} \min_{\lambda\in\Delta_k} \|\widetilde{e}[\lambda]\|_2 \\
&= -Ln^{-1/2} \min_{\lambda\in\Delta_k} \sqrt{\sum_{i=1}^k \lambda_i^2\sigma_i^2\|e^{\ell_i}\|_2^2} && \text{(see } 1^0\text{.B)} \\
&= -Ln^{-1/2} \min_{\lambda\in\Delta_k} \sqrt{\sum_{i=1}^k \lambda_i^2 n} && \text{(see } 1^0\text{.A)} \\
&\le -Lk^{-1/2}.
\end{aligned}$$

We see that $\min_{x\in\Delta_n} f(x) \le -Lk^{-1/2}$, which combines with (46) to yield that

$$f(x_k) - \min_{x\in\Delta_n} f \ge Lk^{-1/2} - k\delta.$$

Since $f \in \mathcal{F}$, $x_k = x^{k-1}(\mathcal{B}, f)$, and $\delta > 0$ is arbitrary, (45) follows. □

Acknowledgments. We gratefully acknowledge the help of members of the PARAPET consortium, especially Matthew Jacobson and Dr. Ron Levkovitz. We are greatly indebted to anonymous referees for their suggestions aimed at improving the structure of the paper.

REFERENCES

[Ber97] D.P. Bertsekas, *A new class of incremental gradient methods for least squares problems*, SIAM J. Optim., 7 (1997), pp. 913–926.

[Ber95] D.P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, MA, 1995.

[Ber96] D.P. Bertsekas, *Incremental least squares methods and the extended Kalman filter*, SIAM J. Optim., 6 (1996), pp. 807–822.

[Hud94] H.M. Hudson and R.S. Larkin, *Accelerated image reconstruction using OS of projection data*, IEEE Trans. Medical Imaging, 13 (1994), pp. 601–609.

[Kam98] C. Kamphuis and F.J. Beekman, *Accelerated iterative transmission CT reconstruction using an OS convex algorithm*, IEEE Trans. Medical Imaging, 17 (1998), pp. 1101–1105.

[KLP99] K.C. Kiwiel, T. Larson, and P.O. Lindberg, *The efficiency of ballstep subgradient level methods for convex optimization*, Math. Oper. Res., 24 (1999), pp. 237–254.

[Lan84] K. Lange and R. Carson, *EM reconstruction algorithms for emission and transmission tomography*, J. Comp. Assist. Tomogr., 8 (1984), pp. 306–316.

[Luo91] Z.Q. Luo, *On the convergence of the LMS algorithm with adaptive learning rate for linear feedforward networks*, Neural Computation, 3 (1991), pp. 226–245.

[Luo94] Z.Q. Luo and P.Tseng, *Analysis of an approximate gradient projection method with applications to the backpropagation algorithm*, Optim. Methods Softw., 4 (1994), pp. 85–101.

[Man95] S.H. Manglos, G.M. Gagne, A. Krol, F.D. Thomas, and R. Narayanaswamy, *Transmission maximum-likelihood reconstruction with OS for cone bean CT*, Phys. Med. Biol., 40 (1995), pp. 1225–1241.

[Nem78] A. Nemirovski and D. Yudin, *Problem Complexity and Method Efficiency in Optimization*, Nauka Publishers, Moscow, 1978 (in Russian); John Wiley, New York, 1983 (in English).

[Pol67] B.T. Polyak, *A general method for solving extremal problems*, Soviet Math. Doklady, 174 (1967), pp. 33–36.

[RW98] R.T. Rockafellar and R.J-B. Wets, *Variational Analysis*, Springer, New York, 1998.

[Roc76] A. Rockmore and A. Makobvski, *A maximum likelihood approach to emission image reconstruction from projections*, IEEE Trans. Nucl. Sci., 23 (1976), pp. 1428–1432.

[She74] L.A. Shepp and B.F. Logan, *The Fourier reconstruction of a head section*, IEEE Trans. Nucl. Sci., 32 (1974), pp. 21–43.

[She82] L.A. Shepp and Y. Vardi, *Maximum likelihood reconstruction for emission tomography*, IEEE Trans. Medical Imaging, MI-1 (1982), pp. 113–122.

[Sho67] N.Z. Shor, *Generalized gradient descent with application to block programming*, Kibernetika, 3, (1967) (in Russian).

[Thi99] K. Thielemans, *A Data Library for the PARAPET Project*, PARAPET ESPRIT consortium deliveravle 1.2, 1999.

[Tse98] P. Tseng, *An incremental gradient(-projection) method with momentum term and adaptive stepsize rule*, SIAM J. Optim., 8 (1998), pp. 506–531.

[Var85] Y. Vardi, L.A. Shepp, and L. Kaufman, *A statistical model for positron emission tomography*, J. Amer. Statist. Assoc., 80 (1985), pp. 8–37.

[Zai98] H. Zaidi, C. Labbe, and C. Morel, *Implementation of a Monte Carlo simulation environment for fully 3D PET on a high performance parallel platform*, Parallel Comput., 24 (1998), pp. 1523–1536.

[Zai99] H. Zaidi, A.K. Hermann Scheurer, and C. Morel, *An object-oriented Monte Carlo simulator for 3D cylindrical PET tomographs*, Computer Methods and Programs in Biomedicine, 58 (1999), pp. 133–145.

SIAM J. OPTIM.
Vol. 12, No. 1, pp. 109–138

INCREMENTAL SUBGRADIENT METHODS FOR NONDIFFERENTIABLE OPTIMIZATION*

ANGELIA NEDIĆ† AND DIMITRI P. BERTSEKAS†

Abstract. We consider a class of subgradient methods for minimizing a convex function that consists of the sum of a large number of component functions. This type of minimization arises in a dual context from Lagrangian relaxation of the coupling constraints of large scale separable problems. The idea is to perform the subgradient iteration incrementally, by sequentially taking steps along the subgradients of the component functions, with intermediate adjustment of the variables after processing each component function. This incremental approach has been very successful in solving large differentiable least squares problems, such as those arising in the training of neural networks, and it has resulted in a much better practical rate of convergence than the steepest descent method.

In this paper, we establish the convergence properties of a number of variants of incremental subgradient methods, including some that are stochastic. Based on the analysis and computational experiments, the methods appear very promising and effective for important classes of large problems. A particularly interesting discovery is that by randomizing the order of selection of component functions for iteration, the convergence rate is substantially improved.

Key words. nondifferentiable optimization, convex programming, incremental subgradient methods, stochastic subgradient methods

AMS subject classification. 90C25

PII. S1052623499362111

1. Introduction.

Throughout this paper, we focus on the problem

$$\begin{aligned} &\text{minimize} \quad f(x) = \sum_{i=1}^{m} f_i(x) \\ &\text{subject to} \quad x \in X, \end{aligned} \tag{1.1}$$

where $f_i : \Re^n \to \Re$ are convex functions, and X is a nonempty, closed, and convex subset of $\Re^n$. We are primarily interested in the case where f is nondifferentiable. A special case of particular interest is when f is the dual function of a primal separable combinatorial problem of the form

$$\begin{aligned} &\text{maximize} \quad \sum_{i=1}^{m} c_i' y_i \\ &\text{subject to} \quad y_i \in Y_i, \quad i = 1, \ldots, m, \qquad \sum_{i=1}^{m} A_i y_i \geq b, \end{aligned}$$

where prime denotes transposition, c_i are given vectors in $\Re^p$, Y_i is a given finite subset of $\Re^p$, A_i are given $n \times p$ matrices, and b is a given vector in $\Re^n$. Then, by viewing x as a Lagrange multiplier vector for the coupling constraint $\sum_{i=1}^{m} A_i y_i \geq b$, we obtain a dual problem of the form (1.1), where

$$f_i(x) = \max_{y_i \in Y_i} (c_i + A_i' x)' y_i - \beta_i' x, \qquad i = 1, \ldots, m, \tag{1.2}$$

*Received by the editors September 15, 1999; accepted for publication (in revised form) January 19, 2001; published electronically July 2, 2001. This research was supported by the NSF under grant ACI-9873339.

http://www.siam.org/journals/siopt/12-1/36211.html

†Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 (angeg@andja.mit.edu, dimitrib@mit.edu).

β_i are vectors in $\Re^n$ such that $\beta_1 + \cdots + \beta_m = b$, and X is the positive orthant $\{x \in \Re^n \mid x \geq 0\}$. It is well known that solving dual problems of the type above, possibly in a branch-and-bound context, is one of the most important and challenging algorithmic areas of optimization.

A principal method for solving problem (1.1) is the subgradient method

$$x_{k+1} = \mathcal{P}_X \left[x_k - \alpha_k \sum_{i=1}^{m} d_{i,k} \right], \tag{1.3}$$

where $d_{i,k}$ is a subgradient of f_i at x_k, α_k is a positive stepsize, and $\mathcal{P}_X$ denotes projection on the set X. There is an extensive theory for this method (see, e.g., the textbooks by Dem'yanov and Vasil'ev [DeV85], Shor [Sho85], Minoux [Min86], Polyak [Pol87], Hiriart-Urruty and Lemaréchal [HiL93], and Bertsekas [Ber99]). In many important applications, the set X is simple enough so that the projection can be easily implemented. In particular, for the special case of the dual problem (1.1), (1.2), the set X is the positive orthant and projecting on X is not expensive.

The incremental subgradient method is similar to the standard subgradient method (1.3). The main difference is that at each iteration, x is changed incrementally, through a sequence of m steps. Each step is a subgradient iteration for a single component function f_i, and there is one step per component function. Thus, an iteration can be viewed as a cycle of m subiterations. If x_k is the vector obtained after k cycles, the vector x_{k+1} obtained after one more cycle is

$$x_{k+1} = \psi_{m,k}, \tag{1.4}$$

where $\psi_{m,k}$ is obtained after the m steps

$$\psi_{i,k} = \mathcal{P}_X \left[\psi_{i-1,k} - \alpha_k g_{i,k} \right], \qquad g_{i,k} \in \partial f_i(\psi_{i-1,k}), \qquad i = 1, \ldots, m, \tag{1.5}$$

starting with

$$\psi_{0,k} = x_k, \tag{1.6}$$

where $\partial f_i(\psi_{i-1,k})$ denotes the subdifferential (set of all subgradients) of f_i at the point $\psi_{i-1,k}$. The updates described by (1.5) are referred to as the *subiterations* of the kth cycle.

Incremental gradient methods for *differentiable* unconstrained problems have a long tradition, most notably in the training of neural networks, where they are known as *backpropagation methods*. They are related to the Widrow–Hoff algorithm [WiH60] and to stochastic gradient/stochastic approximation methods, and they are supported by several recent convergence analyses (Luo [Luo91], Gaivoronski [Gai94], Grippo [Gri94], Luo and Tseng [LuT94], Mangasarian and Solodov [MaS94], Bertsekas and Tsitsiklis [BeT96], Bertsekas [Ber97], Tseng [Tse98], Bertsekas and Tsitsiklis [BeT00]). It has been experimentally observed that incremental gradient methods often converge much faster than the steepest descent method when far from the eventual limit. However, near convergence, they typically converge slowly because they require a diminishing stepsize (e.g., $\alpha_k = O(1/k)$) for convergence. If α_k is instead taken to be a small enough constant, "convergence" to a limit cycle occurs, as first shown by Luo [Luo91]. In the special case where all the stationary points of f are also stationary points of all the component functions f_i, the limit cycle typically reduces to a single point and convergence is obtained; this is the subject of the paper by Solodov [Sol98].

In general, however, the limit cycle consists of m points, each corresponding to one of the subiterations of (1.5), and these m points are usually distinct.

Incremental subgradient methods exhibit behavior similar to that of incremental gradient methods and are similarly motivated by rate of convergence considerations. They were studied first by Kibardin [Kib80] and more recently by Solodov and Zavriev [SoZ98], Nedić and Bertsekas [NeB99], [NeB00], and Ben-Tal, Margalit, and Nemirovski [BMN00]. An asynchronous parallel version of the incremental subgradient method was proposed by Nedić, Bertsekas, and Borkar [NBB00]. Incremental subgradient methods that are somewhat different from the ones in this paper have been proposed by Kaskavelis and Caramanis [KaC98] and Zhao, Luh, and Wang [ZLW99], while a parallel implementation of related methods was proposed by Kiwiel and Lindberg [KiL00]. These methods share with ours the characteristic of computing a subgradient of only one component f_i per iteration, but differ from ours in that the direction used in an iteration is the sum of the (approximate) subgradients of all the components f_i.

In this paper, we study the convergence properties of the incremental subgradient method for three types of stepsize rules: a *constant stepsize rule*, a *diminishing stepsize rule* (where $\alpha_k \to 0$), and a *dynamic stepsize rule* (where α_k is based on exact or approximate knowledge of the optimal cost function value). Earlier convergence analyses of incremental subgradient methods have focused only on the diminishing stepsize rule. Some understanding into the convergence process is gained by viewing the incremental subgradient method as an approximate subgradient method (or a subgradient method with errors). In particular, we have for all $z \in \Re^n$

$$\begin{aligned}
\left(\sum_{i=1}^m g_{i,k}\right)'(z - x_k) &= \sum_{i=1}^m g'_{i,k}(z - \psi_{i-1,k}) + \sum_{i=1}^m g'_{i,k}(\psi_{i-1,k} - x_k) \\
&\le \sum_{i=1}^m \big(f_i(z) - f_i(\psi_{i-1,k})\big) + \sum_{i=1}^m ||g_{i,k}|| \cdot ||\psi_{i-1,k} - x_k|| \\
&= f(z) - f(x_k) + \sum_{i=2}^m \big(f_i(x_k) - f_i(\psi_{i-1,k})\big) \\
&\quad + \sum_{i=2}^m ||g_{i,k}|| \cdot ||\psi_{i-1,k} - x_k|| \\
&\le f(z) - f(x_k) + \sum_{i=2}^m \big(||\tilde{g}_{i,k}|| + ||g_{i,k}||\big)||\psi_{i-1,k} - x_k|| \\
&\le f(z) - f(x_k) + \sum_{i=2}^m \big(||\tilde{g}_{i,k}|| + ||g_{i,k}||\big) \left(\alpha_k \sum_{j=1}^{i-1} \|\tilde{g}_{i,k}\|\right) \\
&\le f(z) - f(x_k) + \epsilon_k,
\end{aligned}$$

where $\tilde{g}_{i,k} \in \partial f_i(x_k)$, $g_{i,k} \in \partial f_i(\psi_{i-1,k})$, and

$$\epsilon_k = 2\alpha_k \sum_{i=2}^m C_i \left(\sum_{j=1}^{i-1} C_j\right), \qquad C_i = \sup_{k\ge 0}\big\{||g|| \mid g \in \partial f_i(x_k) \cup \partial f_i(\psi_{i-1,k})\big\}.$$

Thus if the subgradients $\tilde{g}_{i,k}$, $g_{i,k}$ are bounded so that the C_i are finite, ϵ_k is bounded and diminishes to zero if $\alpha_k \to 0$. It follows that if a diminishing stepsize rule ($\alpha_k \to 0$)

is used and some additional conditions hold, such as $\sum_{k=0}^{\infty} \alpha_k = \infty$, some of the convergence properties of the incremental method can be derived from known results on ϵ-subgradient methods (see, e.g., Dem'yanov and Vasil'ev [DeV85], Polyak [Pol87, p. 144], Correa and Lemaréchal [CoL93], Hiriart-Urruty and Lemaréchal [HiL93], and Bertsekas [Ber99]). However, the connection with ϵ-subgradient methods is not helpful for the convergence analysis under the other stepsize rules that we consider (constant and dynamic), because for these rules α_k need not tend to 0, and the same is true for ϵ_k. As a consequence, there are no convergence results for ϵ-subgradient methods under these rules, which can be applied to our analysis.

We also propose a randomized version of the incremental subgradient method (1.4)–(1.6), where the component function f_i in (1.5) is chosen randomly among the components $f_1, \ldots, f_m$, according to a uniform distribution. This method may be viewed as a stochastic subgradient method for the problem

$$\min_{x \in X} E_\omega \{ f_\omega(x) \},$$

where ω is a random variable that is uniformly distributed over the index set $\{1, \ldots, m\}$. Thus some of the insights and analysis from the stochastic subgradient methods can be brought to bear (see e.g., Ermoliev [Erm69], [Erm76], [Erm83], [Erm88], Shor [Sho85, p. 46], and Bertsekas and Tsitsiklis [BeT96]). Nonetheless, the idea of using randomization in the context of deterministic nondifferentiable optimization is original and much of our analysis, particularly the part that relates to the constant and the dynamic stepsize rules in section 3, is also original. An important conclusion, based on Propositions 2.1 and 3.1, is that randomization has a significant favorable effect on the method's performance; see also the discussion in section 3 and Nedić and Bertsekas [NeB99], [NeB00] which provide convergence rate estimates.

The paper is organized as follows. In the next section, we analyze the convergence of the incremental subgradient method under the three types of stepsize rules mentioned above. In section 3, we establish the convergence properties of randomized versions of the method. Finally, in section 4, we present some computational results. In particular, we compare the performance of the ordinary subgradient method with that of the incremental subgradient method, and we compare different order rules for processing the component functions f_i within a cycle. The computational results indicate a substantial performance advantage for the randomized processing order over the fixed order. We trace the reason for this to a substantially better error estimate for the randomized order (compare Propositions 2.1 and 3.1).

2. Convergence analysis of the incremental subgradient method. Throughout this paper, we use the notation

$$f^* = \inf_{x \in X} f(x), \qquad X^* = \{x \in X \mid f(x) = f^*\}, \qquad dist(x, X^*) = \inf_{x^* \in X^*} \|x - x^*\|,$$

where $\|\cdot\|$ denotes the standard Euclidean norm. Our convergence results in this section use the following assumption.

Assumption 2.1 (subgradient boundedness). There exist scalars $C_1, \ldots, C_m$ such that

$$||g|| \leq C_i \qquad \forall\ g \in \partial f_i(x_k) \cup \partial f_i(\psi_{i-1,k}), \qquad i = 1, \ldots, m, \quad k = 0, 1, \ldots.$$

We note that Assumption 2.1 is satisfied if each f_i is polyhedral (i.e., f_i is the pointwise maximum of a finite number of affine functions). In particular, Assumption 2.1 holds for the dual problem (1.1), (1.2), where for each i and all x the set of subgradients $\partial f_i(x)$ is the convex hull of a finite number of points. More generally, since each component f_i is real-valued and convex over the entire space $\Re^n$, the subdifferential $\partial f_i(x)$ is nonempty and compact for all x and i. If the set X is compact or the sequences $\{\psi_{i,k}\}$ are bounded, then Assumption 2.1 is satisfied since the set $\cup_{x\in B}\partial f_i(x)$ is bounded for any bounded set B (see, e.g., Bertsekas [Ber99, Prop. B.24]).

The following lemma gives an estimate that will be used repeatedly in the subsequent convergence analysis.

LEMMA 2.1. *Let Assumption* 2.1 *hold and let* $\{x_k\}$ *be the sequence generated by the incremental subgradient method* (1.4)–(1.6). *Then for all* $y \in X$ *and* $k \geq 0$, *we have*

$$||x_{k+1} - y||^2 \leq ||x_k - y||^2 - 2\alpha_k\big(f(x_k) - f(y)\big) + \alpha_k^2 C^2, \tag{2.1}$$

where $C = \sum_{i=1}^m C_i$ *and* C_i *is as in Assumption* 2.1.

Proof. Using the nonexpansion property of the projection, the subgradient boundedness (cf. Assumption 2.1), and the subgradient inequality for each component function f_i, we obtain for all $y \in X$

$$\begin{aligned}
||\psi_{i,k} - y||^2 &= ||\mathcal{P}_X[\psi_{i-1,k} - \alpha_k g_{i,k}] - y||^2 \\
&\leq ||\psi_{i-1,k} - \alpha_k g_{i,k} - y||^2 \\
&\leq ||\psi_{i-1,k} - y||^2 - 2\alpha_k g_{i,k}'(\psi_{i-1,k} - y) + \alpha_k^2 C_i^2 \\
&\leq ||\psi_{i-1,k} - y||^2 - 2\alpha_k\big(f_i(\psi_{i-1,k}) - f_i(y)\big) + \alpha_k^2 C_i^2 \qquad \forall\ i,k.
\end{aligned}$$

By adding the above inequalities over $i = 1, \ldots, m$, we have for all $y \in X$ and k

$$\begin{aligned}
||x_{k+1} - y||^2 &\leq ||x_k - y||^2 - 2\alpha_k \sum_{i=1}^m \big(f_i(\psi_{i-1,k}) - f_i(y)\big) + \alpha_k^2 \sum_{i=1}^m C_i^2 \\
&= ||x_k - y||^2 - 2\alpha_k \left(f(x_k) - f(y) + \sum_{i=1}^m \big(f_i(\psi_{i-1,k}) - f_i(x_k)\big) \right) \\
&\quad + \alpha_k^2 \sum_{i=1}^m C_i^2.
\end{aligned}$$

By strengthening the above inequality, we have for all $y \in X$ and k

$$\begin{aligned}
||x_{k+1} - y||^2 &\leq ||x_k - y||^2 - 2\alpha_k\big(f(x_k) - f(y)\big) \\
&\quad + 2\alpha_k \sum_{i=1}^m C_i ||\psi_{i-1,k} - x_k|| + \alpha_k^2 \sum_{i=1}^m C_i^2 \\
&\leq ||x_k - y||^2 - 2\alpha_k\big(f(x_k) - f(y)\big) \\
&\quad + \alpha_k^2 \left(2\sum_{i=2}^m C_i \left(\sum_{j=1}^{i-1} C_j \right) + \sum_{i=1}^m C_i^2 \right) \\
&= ||x_k - y||^2 - 2\alpha_k\big(f(x_k) - f(y)\big) + \alpha_k^2 \left(\sum_{i=1}^m C_i \right)^2 \\
&= ||x_k - y||^2 - 2\alpha_k\big(f(x_k) - f(y)\big) + \alpha_k^2 C^2,
\end{aligned}$$

where in the first inequality we use the relation

$$f_i(x_k) - f_i(\psi_{i-1,k}) \leq ||\tilde{g}_{i,k}|| \cdot ||\psi_{i-1,k} - x_k|| \leq C_i||\psi_{i-1,k} - x_k||$$

with $\tilde{g}_{i,k} \in \partial f_i(x_k)$, and in the second inequality we use the relation

$$||\psi_{i,k} - x_k|| \leq \alpha_k \sum_{j=1}^{i} C_j, \qquad i = 1, \ldots, m, \quad k \geq 0,$$

which follows from (1.4)–(1.6) and Assumption 2.1. □

Among other things, Lemma 2.1 guarantees that given the current iterate x_k and some other point $y \in X$ with lower cost than x_k, the next iterate x_{k+1} will be closer to y than x_k, provided the stepsize α_k is sufficiently small (less than $2\big(f(x_k)-f(y)\big)/C^2$). This fact is used repeatedly, with a variety of choices for y, in what follows.

2.0.1. Constant stepsize rule. We first consider the case of a constant stepsize rule.

PROPOSITION 2.1. *Let Assumption* 2.1 *hold. Then, for the sequence* $\{x_k\}$ *generated by the incremental method* (1.4)–(1.6) *with the stepsize* α_k *fixed to some positive constant* α*, we have the following:*

(a) *If* $f^* = -\infty$*, then*

$$\liminf_{k\to\infty} f(x_k) = -\infty.$$

(b) *If* $f^* > -\infty$*, then*

$$\liminf_{k\to\infty} f(x_k) \leq f^* + \frac{\alpha C^2}{2},$$

where $C = \sum_{i=1}^{m} C_i$.

Proof. We prove (a) and (b) simultaneously. If the result does not hold, there must exist an $\epsilon > 0$ such that

$$\liminf_{k\to\infty} f(x_k) > f^* + \frac{\alpha C^2}{2} + 2\epsilon.$$

Let $\hat{y} \in X$ be such that

$$\liminf_{k\to\infty} f(x_k) \geq f(\hat{y}) + \frac{\alpha C^2}{2} + 2\epsilon,$$

and let k_0 be large enough so that for all $k \geq k_0$ we have

$$f(x_k) \geq \liminf_{k\to\infty} f(x_k) - \epsilon.$$

By adding the preceding two relations, we obtain for all $k \geq k_0$

$$f(x_k) - f(\hat{y}) \geq \frac{\alpha C^2}{2} + \epsilon.$$

Using Lemma 2.1 for the case where $y = \hat{y}$ together with the above relation, we obtain for all $k \geq k_0$,

$$||x_{k+1} - \hat{y}||^2 \leq ||x_k - \hat{y}||^2 - 2\alpha\epsilon.$$

Thus we have

$$||x_{k+1}-\hat{y}||^2 \leq ||x_k-\hat{y}||^2-2\alpha\epsilon \leq ||x_{k-1}-\hat{y}||^2-4\alpha\epsilon \leq \cdots \leq ||x_{k_0}-\hat{y}||^2-2(k+1-k_0)\alpha\epsilon,$$

which cannot hold for k sufficiently large, a contradiction. □

2.0.2. Diminishing stepsize rule. The next result is the analog of a classical convergence result for the ordinary subgradient method of Ermoliev [Erm66] (see also Polyak [Pol67]).

PROPOSITION 2.2. *Let Assumption* 2.1 *hold and assume that the stepsize* α_k *is such that*

$$\alpha_k > 0, \qquad \lim_{k\to\infty} \alpha_k = 0, \qquad \sum_{k=0}^{\infty} \alpha_k = \infty.$$

Then, for the sequence $\{x_k\}$ *generated by the incremental method* (1.4)–(1.6)*, we have*

$$\liminf_{k\to\infty} f(x_k) = f^*.$$

Proof. The proof uses Lemma 2.1 and Proposition 1.2 of Correa and Lemaréchal [CoL93]. □

If we assume in addition that X^* is nonempty and bounded, Proposition 2.2 can be strengthened as in the next proposition. This proposition is similar to a result of Solodov and Zavriev [SoZ98], which was proved by different methods under the stronger assumption that X is a compact set.

PROPOSITION 2.3. *Let Assumption* 2.1 *hold, and let* X^* *be nonempty and bounded. Also, assume that the stepsize* α_k *is such that*

$$\alpha_k > 0, \qquad \lim_{k\to\infty} \alpha_k = 0, \qquad \sum_{k=0}^{\infty} \alpha_k = \infty.$$

Then, for the sequence $\{x_k\}$ *generated by the incremental subgradient method* (1.4)–(1.6)*, we have*

$$\lim_{k\to\infty} dist(x_k, X^*) = 0, \qquad \lim_{k\to\infty} f(x_k) = f^*.$$

Proof. The idea is to show that once x_k enters a certain level set, it cannot get too far away from that set. Fix a $\gamma > 0$, and let k_0 be such that $\gamma \geq \alpha_k C^2$ for all $k \geq k_0$. We distinguish two cases:

Case 1. $f(x_k) > f^* + \gamma$. From Lemma 2.1 we obtain for all $x^* \in X^*$ and all k

$$||x_{k+1} - x^*||^2 \leq ||x_k - x^*||^2 - 2\alpha_k\big(f(x_k) - f^*\big) + \alpha_k^2 C^2. \tag{2.2}$$

Hence

$$\begin{aligned} ||x_{k+1} - x^*||^2 &< ||x_k - x^*||^2 - 2\gamma\alpha_k + \alpha_k^2 C^2 \\ &= ||x_k - x^*||^2 - \alpha_k(2\gamma - \alpha_k C^2) \\ &\leq ||x_k - x^*||^2 - \alpha_k\gamma, \end{aligned}$$

so that

$$dist(x_{k+1}, X^*) \leq dist(x_k, X^*) - \alpha_k\gamma. \tag{2.3}$$

Case 2. $f(x_k) \leq f^* + \gamma$. This case must occur for infinitely many k, in view of (2.3) and the fact $\sum_{k=0}^{\infty} \alpha_k = \infty$. Since x_k belongs to the level set

$$L_\gamma = \big\{y \in X \mid f(y) \leq f^* + \gamma\big\},$$

which is bounded (in view of the boundedness of X^*), we have

$$dist(x_k, X^*) \leq d(\gamma) < \infty, \tag{2.4}$$

where we denote

$$d(\gamma) = \max_{y \in L_\gamma} dist(y, X^*).$$

From the iteration (1.4)–(1.6), we have $||x_{k+1} - x_k|| \leq \alpha_k C$, so for all $x^* \in X^*$

$$||x_{k+1} - x^*|| \leq ||x_k - x^*|| + ||x_{k+1} - x_k|| \leq ||x_k - x^*|| + \alpha_k C.$$

By taking the minimum over $x^* \in X^*$ and by using (2.4), we obtain

$$dist(x_{k+1}, X^*) \leq d(\gamma) + \alpha_k C. \tag{2.5}$$

Combining (2.3), which holds when $f(x_k) > f^* + \gamma$ (Case 1 above), with (2.5), which holds for the infinitely many k for which $f(x_k) \leq f^* + \gamma$ (Case 2 above), we see that

$$dist(x_k, X^*) \leq d(\gamma) + \alpha_k C \qquad \forall\ k \geq k_0.$$

Therefore, since $\alpha_k \to 0$,

$$\limsup_{k \to \infty} dist(x_k, X^*) \leq d(\gamma) \qquad \forall\ \gamma > 0.$$

In view of the continuity of f and the compactness of its level sets, we have $\lim_{\gamma \to 0} d(\gamma) = 0$, so that $\lim_{k \to \infty} dist(x_k, X^*) = 0$. This relation also implies that $\lim_{k \to \infty} f(x_k) = f^*$. □

The assumption that X^* is nonempty and bounded holds, for example, if all $\inf_{x \in X} f_i(x)$ are finite and at least one of the components f_i has bounded level sets (see Rockafellar [Roc 70, Theorem 9.3]. Proposition 2.3 does not guarantee convergence of the entire sequence $\{x_k\}$. With slightly different assumptions that include an additional mild restriction on the stepsize sequence, this convergence is guaranteed, as indicated in the following proposition.

PROPOSITION 2.4. *Let Assumption* 2.1 *hold and let the optimal set X^* be nonempty. Also assume that the stepsize α_k is such that*

$$\alpha_k > 0, \qquad \sum_{k=0}^{\infty} \alpha_k = \infty, \qquad \sum_{k=0}^{\infty} \alpha_k^2 < \infty.$$

Then the sequence $\{x_k\}$ generated by the incremental subgradient method (1.4)–(1.6) *converges to some optimal solution.*

Proof. Use Lemma 2.1 with $y \in X^*$ and Proposition 1.3 of Correa and Lemaréchal [CoL93]. □

In Propositions 2.2–2.4, we use the same stepsize α_k in all subiterations of a cycle. As shown by Kibardin in [Kib80] and by Nedić, Bertsekas, and Borkar in [NBB00] (for a more general incremental method), the convergence can be preserved if we vary the stepsize α_k within each cycle, provided that the variations of α_k in the cycles are suitably small.

2.0.3. Dynamic stepsize rule for known f^*. The preceding results apply to the constant and the diminishing stepsize choices. An interesting alternative for the ordinary subgradient method is the dynamic stepsize rule

$$\alpha_k = \gamma_k \frac{f(x_k) - f^*}{||g_k||^2},$$

with $g_k \in \partial f(x_k)$, $0 < \underline{\gamma} \leq \gamma_k \leq \overline{\gamma} < 2$, introduced by Polyak in [Pol69] (see also discussions in Shor [Sho85], Brännlund [Brä93], and Bertsekas [Ber99]). For the incremental method, to avoid the calculation of g_k we propose a variant of this stepsize where $||g_k||$ is replaced by an upper bound C:

$$\alpha_k = \gamma_k \frac{f(x_k) - f^*}{C^2}, \qquad 0 < \underline{\gamma} \leq \gamma_k \leq \overline{\gamma} < 2, \tag{2.6}$$

where

$$C = \sum_{i=1}^{m} C_i \tag{2.7}$$

and

$$C_i \geq \sup_{k \geq 0} \big\{ ||g|| \mid g \in \partial f_i(x_k) \cup \partial f_i(\psi_{i-1,k}) \big\}, \qquad i = 1, \ldots, m. \tag{2.8}$$

For this choice of stepsize we must be able to calculate suitable upper bounds C_i, which can be done, for example, when the components f_i are polyhedral.

We first consider the case where f^* is known. We later modify the stepsize, so that f^* can be replaced by a dynamically updated estimate.

PROPOSITION 2.5. *Let Assumption* 2.1 *hold and let the optimal set X^* be nonempty. Then the sequence $\{x_k\}$ generated by the incremental subgradient method* (1.4)–(1.6) *with the dynamic stepsize rule* (2.6)–(2.8) *converges to some optimal solution.*

Proof. From Lemma 2.1 with $y = x^* \in X^*$, we have

$$||x_{k+1} - x^*||^2 \leq ||x_k - x^*||^2 - 2\alpha_k \big(f(x_k) - f^*\big) + \alpha_k^2 C^2 \qquad \forall\, x^* \in X^*, \quad k \geq 0,$$

and by using the definition of α_k (cf. (2.6)), we obtain

$$||x_{k+1} - x^*||^2 \leq ||x_k - x^*||^2 - \underline{\gamma}(2 - \overline{\gamma}) \frac{\big(f(x_k) - f^*\big)^2}{C^2} \qquad \forall\, x^* \in X^*, \quad k \geq 0.$$

Therefore $\{x_k\}$ is bounded. Furthermore, $f(x_k) \to f^*$, since otherwise we would have $||x_{k+1} - x^*|| \leq ||x_k - x^*|| - \epsilon$ for some suitably small $\epsilon > 0$ and infinitely many k. Hence for any limit point $\overline{x}$ of $\{x_k\}$, we have $\overline{x} \in X^*$, and since the sequence $\{||x_k - x^*||\}$ is decreasing, it converges to $||\overline{x} - x^*||$ for every $x^* \in X^*$. If there are two distinct limit points $\tilde{x}$ and $\overline{x}$ of $\{x_k\}$, we must have $\tilde{x} \in X^*$, $\overline{x} \in X^*$, and $||\tilde{x} - x^*|| = ||\overline{x} - x^*||$ for all $x^* \in X^*$, which is possible only if $\tilde{x} = \overline{x}$. □

2.0.4. Dynamic stepsize rule for unknown f^*. In most practical problems the value f^* is not known. In this case we may modify the dynamic stepsize (2.6) by replacing f^* with an estimate. This leads to the stepsize rule

$$\alpha_k = \gamma_k \frac{f(x_k) - f_k^{\text{lev}}}{C^2}, \qquad 0 < \underline{\gamma} \leq \gamma_k \leq \overline{\gamma} < 2, \qquad \forall\, k \geq 0, \tag{2.9}$$

where C is defined by (2.7), (2.8), and $f_k^{\rm lev}$ is an estimate of f^*.

We discuss two procedures for updating $f_k^{\rm lev}$. In both procedures $f_k^{\rm lev}$ is equal to the best function value $\min_{0\le j\le k} f(x_j)$ achieved up to the kth iteration minus a positive amount δ_k which is adjusted based on the algorithm's progress. The first adjustment procedure (new even when specialized to the ordinary subgradient method) is simple but is guaranteed to yield only a δ-optimal objective function value with δ positive and arbitrarily small (unless $f^*=-\infty$ in which case the procedure yields the optimal function value). The second adjustment procedure for $f_k^{\rm lev}$ is more complex but is guaranteed to yield the optimal value f^* in the limit. This procedure is based on the ideas and algorithms of Brännlund [Brä93] and Goffin and Kiwiel [GoK99].

In the first adjustment procedure, $f_k^{\rm lev}$ is given by

$$f_k^{\rm lev}=\min_{0\le j\le k} f(x_j)-\delta_k, \tag{2.10}$$

and δ_k is updated according to

$$\delta_{k+1}=\begin{cases}\rho\delta_k & \text{if } f(x_{k+1})\le f_k^{\rm lev},\\ \max\{\beta\delta_k,\delta\} & \text{if } f(x_{k+1})> f_k^{\rm lev},\end{cases} \tag{2.11}$$

where δ_0, δ, β, and ρ are fixed positive constants with $\beta<1$ and $\rho\ge 1$. Thus in this procedure we essentially "aspire" to reach a target level that is smaller by δ_k over the best value achieved thus far. Whenever the target level is achieved, we increase δ_k or we keep it at the same value depending on the choice of ρ. If the target level is not attained at a given iteration, δ_k is reduced up to a threshold δ. This threshold guarantees that the stepsize α_k of (2.9) is bounded away from zero, since from (2.10) we have $f(x_k)-f_k^{\rm lev}\ge\delta$ and hence

$$\alpha_k\ge\underline{\gamma}\,\frac{\delta}{C^2}.$$

As a result, the method's behavior resembles the one with a constant stepsize (cf. Proposition 2.1), as indicated by the following proposition.

PROPOSITION 2.6. *Let Assumption* 2.1 *hold. Then, for the sequence* $\{x_k\}$ *generated by the incremental method* (1.4)–(1.6) *and the dynamic stepsize rule* (2.9) *with the adjustment procedure* (2.10)–(2.11)*, we have*

(a) *If* $f^*=-\infty$*, then*

$$\inf_{k\ge 0} f(x_k)=f^*.$$

(b) *If* $f^*>-\infty$*, then*

$$\inf_{k\ge 0} f(x_k)\le f^*+\delta.$$

Proof. To arrive at a contradiction, assume that

$$\inf_{k\ge 0} f(x_k)> f^*+\delta. \tag{2.12}$$

Each time the target level is attained (i.e., $f(x_k)\le f_{k-1}^{\rm lev}$), the current best function value $\min_{0\le j\le k} f(x_j)$ decreases by at least δ (cf. (2.10) and (2.11)), so in view of (2.12), the target value can be attained only a finite number of times. From (2.11) it

follows that after finitely many iterations, δ_k is decreased to the threshold value and remains at that value for all subsequent iterations; i.e., there is an index $\overline{k}$ such that

$$\delta_k = \delta, \qquad \forall\ k \geq \overline{k}. \tag{2.13}$$

In view of (2.12), there exists $\overline{y} \in X$ such that $\inf_{k\geq 0} f(x_k) - \delta \geq f(\overline{y})$. From (2.10) and (2.13), we have

$$f_k^{\rm lev} = \min_{0\leq j\leq k} f(x_j) - \delta \geq \inf_{k\geq 0} f(x_k) - \delta \geq f(\overline{y}) \qquad \forall\ k \geq \overline{k},$$

so that

$$\alpha_k\big(f(x_k) - f(\overline{y})\big) \geq \alpha_k\big(f(x_k) - f_k^{\rm lev}\big) = \gamma_k \left(\frac{f(x_k) - f_k^{\rm lev}}{C}\right)^2 \qquad \forall\ k \geq \overline{k}.$$

By using Lemma 2.1 with $y = \overline{y}$, we have

$$||x_{k+1} - \overline{y}||^2 \leq ||x_k - \overline{y}||^2 - 2\alpha_k\big(f(x_k) - f(\overline{y})\big) + \alpha_k^2 C^2 \qquad \forall\ k \geq 0.$$

By combining the preceding two relations and the definition of α_k (cf. (2.9)), we obtain

$$\begin{aligned} ||x_{k+1} - \overline{y}||^2 &\leq ||x_k - \overline{y}||^2 - 2\gamma_k \left(\frac{f(x_k) - f_k^{\rm lev}}{C}\right)^2 + \gamma_k^2 \left(\frac{f(x_k) - f_k^{\rm lev}}{C}\right)^2 \\ &= ||x_k - \overline{y}||^2 - \gamma_k(2 - \gamma_k) \left(\frac{f(x_k) - f_k^{\rm lev}}{C}\right)^2 \\ &\leq ||x_k - \overline{y}||^2 - \underline{\gamma}(2 - \overline{\gamma})\frac{\delta^2}{C^2} \qquad \forall\ k \geq \overline{k}, \end{aligned}$$

where the last inequality follows from the facts $\gamma_k \in [\underline{\gamma}, \overline{\gamma}]$ and $f(x_k) - f_k^{\rm lev} \geq \delta$ for all k. By summing the above inequalities over k, we have

$$||x_k - \overline{y}||^2 \leq ||x_{\overline{k}} - \overline{y}||^2 - (k - \overline{k})\underline{\gamma}(2 - \overline{\gamma})\frac{\delta^2}{C^2} \qquad \forall\ k \geq \overline{k},$$

which cannot hold for large k—a contradiction. □

When $m = 1$, the incremental subgradient method (1.4)–(1.6) becomes the ordinary subgradient method

$$x_{k+1} = \mathcal{P}_X[x_k - \alpha_k g_k] \qquad \forall\ k \geq 0.$$

The dynamic stepsize rule (2.9) using the adjustment procedure of (2.10)–(2.11) (with $C = ||g_k||$), and the convergence result of Proposition 2.6 are new to our knowledge for this method.

We now consider the second procedure for adjusting $f_k^{\rm lev}$, which guarantees that $f_k^{\rm lev} \to f^*$, and convergence of the associated method to the optimum. In this procedure we reduce δ_k whenever the method "travels" for a long distance without reaching the corresponding target level.

Path-Based Incremental Target Level Algorithm.

Step 0 (*Initialization*): Select x_0, $\delta_0 > 0$, and $B > 0$. Set $\sigma_0 = 0$, $f_{-1}^{\rm rec} = \infty$. Set $k = 0$, $l = 0$, and $k(l) = 0$ [$k(l)$ will denote the iteration number when the lth update of $f_k^{\rm lev}$ occurs].

Step 1 (*Function evaluation*): Calculate $f(x_k)$. If $f(x_k) < f_{k-1}^{\rm rec}$, then set $f_k^{\rm rec} = f(x_k)$. Otherwise set $f_k^{\rm rec} = f_{k-1}^{\rm rec}$ [so that $f_k^{\rm rec}$ keeps the record of the smallest value attained by the iterates that are generated so far, i.e., $f_k^{\rm rec} = \min_{0\le j\le k} f(x_j)$].

Step 2 (*Sufficient descent*): If $f(x_k) \le f_{k(l)}^{\rm rec} - \frac{\delta_l}{2}$, then set $k(l+1) = k$, $\sigma_k = 0$, $\delta_{l+1} = \delta_l$, increase l by 1, and go to Step 4.

Step 3 (*Oscillation detection*): If $\sigma_k > B$, then set $k(l+1) = k$, $\sigma_k = 0$, $\delta_{l+1} = \frac{\delta_l}{2}$, and increase l by 1.

Step 4 (*Iterate update*): Set $f_k^{\rm lev} = f_{k(l)}^{\rm rec} - \delta_l$. Select $\gamma_k \in [\underline{\gamma}, \overline{\gamma}]$ and calculate x_{k+1} via (1.4)–(1.6) with the stepsize (2.9).

Step 5 (*Path length update*): Set $\sigma_{k+1} = \sigma_k + \alpha_k C$. Increase k by 1 and go to Step 1.

The algorithm uses the same target level $f_k^{\rm lev} = f_{k(l)}^{\rm rec} - \delta_l$ for $k = k(l), k(l)+1, \ldots, k(l+1)-1$. The target level is updated only if sufficient descent or oscillation is detected (Step 2 or Step 3, respectively). It can be shown that the value σ_k is an upper bound on the length of the path traveled by iterates $x_{k(l)}, \ldots, x_k$ for $k < k(l+1)$. Whenever σ_k exceeds the prescribed upper bound B on the path length, the parameter δ_l is decreased, which increases the target level $f_k^{\rm lev}$.

We will show that $\inf_{k\ge 0} f(x_k) = f^*$ even if f^* is not finite. First, we give a preliminary result showing that the target values $f_k^{\rm lev}$ are updated infinitely often (i.e., $l \to \infty$), and that $\inf_{k\ge 0} f(x_k) = -\infty$ if δ_l is nondiminishing.

LEMMA 2.2. *Let Assumption* 2.1 *hold. Then for the path-based incremental target level algorithm we have* $l \to \infty$, *and either* $\inf_{k\ge 0} f(x_k) = -\infty$ *or* $\lim_{l\to\infty} \delta_l = 0$.

Proof. Assume that l takes only a finite number of values, say $l = 0, 1, \ldots, \bar{l}$. In this case we have $\sigma_k + \alpha_k C = \sigma_{k+1} \le B$ for all $k \ge k(\bar{l})$, so that $\lim_{k\to\infty} \alpha_k = 0$. But this is impossible, since for all $k \ge k(\bar{l})$ we have

$$\alpha_k = \gamma_k \frac{f(x_k) - f_k^{\rm lev}}{C^2} \ge \underline{\gamma}\, \frac{\delta_{\bar{l}}}{C^2} > 0.$$

Hence $l \to \infty$.

Let $\delta = \lim_{l\to\infty} \delta_l$. If $\delta > 0$, then from Steps 2 and 3 it follows that for all l large enough, we have $\delta_l = \delta$ and

$$f_{k(l+1)}^{\rm rec} - f_{k(l)}^{\rm rec} \le -\frac{\delta}{2},$$

implying that $\inf_{k\ge 0} f(x_k) = -\infty$. □

We have the following convergence result. In the special case of the ordinary subgradient method, this result was proved by Goffin and Kiwiel [GoK99] using a different (and much longer) proof.

PROPOSITION 2.7. *Let Assumption* 2.1 *hold. Then, for the sequence* $\{x_k\}$ *generated by the path-based incremental target level algorithm, we have*

$$\inf_{k\ge 0} f(x_k) = f^*.$$

Proof. If $\lim_{l\to\infty} \delta_l > 0$, then, according to Lemma 2.2, we have $\inf_{k\ge 0} f(x_k) = -\infty$ and we are done, so assume that $\lim_{l\to\infty} \delta_l = 0$. Let L be given by

$$L = \left\{ l \in \{1, 2, \ldots\} \;\middle|\; \delta_l = \frac{\delta_{l-1}}{2} \right\}.$$

Then, from Steps 3 and 5, we obtain

$$\sigma_k = \sigma_{k-1} + \alpha_{k-1}C = \sum_{j=k(l)}^{k-1} C\alpha_j,$$

so that $k(l+1) = k$ and $l+1 \in L$ whenever $\sum_{j=k(l)}^{k-1} \alpha_j C > B$ at Step 3. Hence

$$\sum_{j=k(l-1)}^{k(l)-1} \alpha_j > \frac{B}{C} \qquad \forall\, l \in L,$$

and, since the cardinality of L is infinite, we have

$$\sum_{j=0}^{\infty} \alpha_j \geq \sum_{l\in L} \sum_{j=k(l-1)}^{k(l)-1} \alpha_j > \sum_{l\in L} \frac{B}{C} = \infty. \tag{2.14}$$

Now, in order to arrive at a contradiction, assume that $\inf_{k\geq 0} f(x_k) > f^*$, so that for some $\hat{y} \in X$ and some $\epsilon > 0$

$$\inf_{k\geq 0} f(x_k) - \epsilon \geq f(\hat{y}). \tag{2.15}$$

Since $\delta_l \to 0$, there is a large enough $\hat{l}$ such that $\delta_l \leq \epsilon$ for all $l \geq \hat{l}$, so that for all $k \geq k(\hat{l})$

$$f_k^{\text{lev}} = f_{k(l)}^{\text{rec}} - \delta_l \geq \inf_{k\geq 0} f(x_k) - \epsilon \geq f(\hat{y}).$$

Using this relation, Lemma 2.1 for $y = \hat{y}$, and the definition of α_k, we obtain

$$\begin{aligned}
||x_{k+1} - \hat{y}||^2 &\leq ||x_k - \hat{y}||^2 - 2\alpha_k\big(f(x_k) - f(\hat{y})\big) + \alpha_k^2 C^2 \\
&\leq ||x_k - \hat{y}||^2 - 2\alpha_k\big(f(x_k) - f_k^{\text{lev}}\big) + \alpha_k^2 C^2 \\
&= ||x_k - \hat{y}||^2 - \gamma_k(2-\gamma_k)\frac{\big(f(x_k) - f_k^{\text{lev}}\big)^2}{C^2} \\
&\leq ||x_k - \hat{y}||^2 - \underline{\gamma}(2-\overline{\gamma})\frac{\big(f(x_k) - f_k^{\text{lev}}\big)^2}{C^2} \qquad \forall\, k \geq k(l).
\end{aligned}$$

By summing these inequalities over $k \geq k(\hat{l})$, we have

$$\frac{\underline{\gamma}(2-\overline{\gamma})}{C^2} \sum_{k=k(\hat{l})}^{\infty} \big(f(x_k) - f_k^{\text{lev}}\big)^2 \leq ||x_{k(\hat{l})} - \hat{y}||^2,$$

and consequently $\sum_{k=k(\hat{l})}^{\infty} \alpha_k^2 < \infty$ (see the definition of α_k in (2.9)). Since $\alpha_k \to 0$ and $\sum_{k=0}^{\infty} \alpha_k = \infty$ (cf. (2.14)), according to Proposition 2.2, we must have

$$\liminf_{k\to\infty} f(x_k) = f^*.$$

Hence $\inf_{k\geq 0} f(x_k) = f^*$, which contradicts (2.15). □

In an attempt to improve the efficiency of the path-based incremental target level algorithm, one may introduce parameters $\beta, \tau \in (0,1)$ and $\rho \geq 1$ (whose values will be fixed at Step 0), and modify Steps 2 and 3 as follows:

Step 2′ If $f(x_k) \leq f^{\rm rec}_{k(l)} - \tau\delta_l$, then set $k(l+1) = k$, $\sigma_k = 0$, $\delta_{l+1} = \rho\delta_l$, increase l by 1, and go to Step 4.

Step 3′ If $\sigma_k > B$, then set $k(l+1) = k$, $\sigma_k = 0$, $\delta_{l+1} = \beta\delta_l$, and increase l by 1.

It can be seen that the result of Proposition 2.7 still holds for this modified algorithm. If we choose $\rho > 1$ at Step 3′, then in the proofs of Lemma 2.2 and Proposition 2.7 we have to replace $\lim_{l\to\infty}\delta_l$ with $\limsup_{l\to\infty}\delta_l$.

Let us remark that there is no need to keep the path bound B fixed. Instead, as the method progresses, we can decrease B in such a way that $\sum_{l\in L} B_l = \infty$ holds, which ensures that the convergence result of Proposition 2.7 is preserved (cf. (2.14)).

It can be verified that all the results presented in this section are valid for the incremental method that does not use projections within the cycles but rather employs projections at the end of cycles:

$$\psi_{i,k} = \psi_{i-1,k} - \alpha_k g_{i,k}, \qquad g_{i,k} \in \partial f_i(\psi_{i-1,k}), \quad i = 1, \ldots, m,$$

where $\psi_{0,k} = x_k$ and the iterate x_{k+1} is given by

$$x_{k+1} = \mathcal{P}_X[\psi_{m,k}].$$

This method and its modifications, including additive-type errors on subgradients, synchronous parallelization, and a momentum term is given by Solodov and Zavriev [SoZ98] and is analyzed for the case of a compact set X and a diminishing stepsize rule.

3. An incremental subgradient method with randomization. It can be verified that the preceding convergence analysis goes through assuming any order for processing the component functions f_i, as long as each component is taken into account exactly once within a cycle. In particular, at the beginning of each cycle k, we could reorder the components f_i by either shifting or reshuffling and then proceed with the calculations until the end of the cycle. However, the order used can significantly affect the rate of convergence of the method. Unfortunately, determining the most favorable order may be very difficult in practice. A popular technique for incremental gradient methods (for differentiable components f_i) is to reshuffle randomly the order of the functions f_i at the beginning of each cycle. A variation of this method is to pick randomly a function f_i at each iteration rather than to pick each f_i exactly once in every cycle according to a randomized order. This variation can be viewed as a gradient method with random errors, as shown in Bertsekas and Tsitsiklis [BeT96, p. 143] (see also [BeT00]). Similarly, the corresponding incremental subgradient method at each step picks randomly a function f_i to be processed next. For the case of a diminishing stepsize, the convergence of the method follows from known stochastic subgradient convergence results (e.g., Ermoliev [Erm69], [Erm88], Polyak [Pol87, p. 159])—see the subsequent Proposition 3.2. In this section, we also analyze the method for the constant and dynamic stepsize rules. This analysis is new and has no counterpart in the available stochastic subgradient literature.

The formal description of the randomized method is as follows:

$$x_{k+1} = \mathcal{P}_X\big[x_k - \alpha_k g(\omega_k, x_k)\big], \tag{3.1}$$

where ω_k is a random variable taking equiprobable values from the set $\{1, \ldots, m\}$ and $g(\omega_k, x_k)$ is a subgradient of the component f_{ω_k} at x_k. This simply means that if the random variable ω_k takes a value j, then the vector $g(\omega_k, x_k)$ is a subgradient of f_j at x_k.

Throughout this section we assume the following regarding the randomized method (3.1).

Assumption 3.1.

(a) The sequence $\{\omega_k\}$ is a sequence of independent random variables, each uniformly distributed over the set $\{1,\dots,m\}$. Furthermore, the sequence $\{\omega_k\}$ is independent of the sequence $\{x_k\}$.

(b) The set of subgradients $\{g(\omega_k,x_k) \mid k=0,1,\dots\}$ is bounded, i.e., there exists a positive constant C_0 such that with probability 1

$$||g(\omega_k,x_k)|| \le C_0 \qquad \forall\ k \ge 0.$$

Note that if the set X is compact or the components f_i are polyhedral, then Assumption 3.1(b) is satisfied. The proofs of several propositions in this section rely on the supermartingale convergence theorem as stated, for example, in Bertsekas and Tsitsiklis [BeT96, p. 148].

THEOREM 3.1 (supermartingale convergence theorem). *Let Y_k, Z_k, and W_k, $k=0,1,2,\dots$, be three sequences of random variables and let $\mathcal{F}_k$, $k=0,1,2,\dots$, be sets of random variables such that $\mathcal{F}_k \subset \mathcal{F}_{k+1}$ for all k. Suppose that*

(a) *the random variables Y_k, Z_k, and W_k are nonnegative, and are functions of the random variables in $\mathcal{F}_k$;*

(b) *for each k, we have $E\{Y_{k+1} \mid \mathcal{F}_k\} \le Y_k - Z_k + W_k$;*

(c) *there holds $\sum_{k=0}^{\infty} W_k < \infty$.*

Then we have $\sum_{k=0}^{\infty} Z_k < \infty$, and the sequence Y_k converges to a nonnegative random variable Y, with probability 1.

3.0.5. Constant stepsize rule.

PROPOSITION 3.1. *Let Assumption* 3.1 *hold. Then, for the sequence $\{x_k\}$ generated by the randomized incremental method* (3.1)*, with the stepsize α_k fixed to some positive constant α, we have the following:*

(a) *If $f^* = -\infty$, then with probability* 1

$$\inf_{k\ge 0} f(x_k) = f^*.$$

(b) *If $f^* > -\infty$, then with probability* 1

$$\inf_{k\ge 0} f(x_k) \le f^* + \frac{\alpha m C_0^2}{2}.$$

Proof. By adapting Lemma 2.1 to the case where f is replaced by f_{ω_k}, we have

$$||x_{k+1}-y||^2 \le ||x_k-y||^2 - 2\alpha\big(f_{\omega_k}(x_k) - f_{\omega_k}(y)\big) + \alpha^2 C_0^2 \qquad \forall\ y\in X,\quad k\ge 0.$$

By taking the conditional expectation with respect to $\mathcal{F}_k = \{x_0,\dots,x_k\}$, the method's history up to x_k, we obtain for all $y\in X$ and k

$$\begin{aligned} E\big\{||x_{k+1}-y||^2 \mid \mathcal{F}_k\big\} &\le ||x_k-y||^2 - 2\alpha E\big\{f_{\omega_k}(x_k) - f_{\omega_k}(y) \mid \mathcal{F}_k\big\} + \alpha^2 C_0^2 \\ &= ||x_k-y||^2 - 2\alpha \sum_{i=1}^{m} \frac{1}{m}\big(f_i(x_k) - f_i(y)\big) + \alpha^2 C_0^2 \qquad (3.2)\\ &= ||x_k-y||^2 - \frac{2\alpha}{m}\big(f(x_k) - f(y)\big) + \alpha^2 C_0^2, \end{aligned}$$

where the first equality follows since ω_k takes the values $1,\ldots,m$ with equal probability $1/m$.

Now, fix a nonnegative integer N, consider the level set L_N defined by

$$L_N = \begin{cases} \left\{x \in X \mid f(x) < -N+1+\frac{\alpha m C_0^2}{2}\right\} & \text{if } f^* = -\infty, \\ \left\{x \in X \mid f(x) < f^* + \frac{2}{N} + \frac{\alpha m C_0^2}{2}\right\} & \text{if } f^* > -\infty, \end{cases}$$

and let $y_N \in X$ be such that

$$f(y_N) = \begin{cases} -N & \text{if } f^* = -\infty, \\ f^* + \frac{1}{N} & \text{if } f^* > -\infty. \end{cases}$$

Note that $y_N \in L_N$ by construction. Define a new process $\{\hat{x}_k\}$ as follows

$$\hat{x}_{k+1} = \begin{cases} \mathcal{P}_X\big[\hat{x}_k - \alpha g(\omega_k, \hat{x}_k)\big] & \text{if } \hat{x}_k \notin L_N, \\ y_N & \text{otherwise,} \end{cases}$$

where $\hat{x}_0 = x_0$. Thus the process $\{\hat{x}_k\}$ is identical to $\{x_k\}$, except that once x_k enters the level set L_N, the process terminates with $\hat{x}_k = y_N$ (since $y_N \in L_N$). Using (3.2) with $y = y_N$, we have

$$E\big\{||\hat{x}_{k+1} - y_N||^2 \mid \mathcal{F}_k\big\} \le ||\hat{x}_k - y_N||^2 - \frac{2\alpha}{m}\big(f(\hat{x}_k) - f(y_N)\big) + \alpha^2 C_0^2,$$

or equivalently

$$E\big\{||\hat{x}_{k+1} - y_N||^2 \mid \mathcal{F}_k\big\} \le ||\hat{x}_k - y_N||^2 - z_k, \tag{3.3}$$

where

$$z_k = \begin{cases} \frac{2\alpha}{m}\big(f(\hat{x}_k) - f(y_N)\big) - \alpha^2 C_0^2 & \text{if } \hat{x}_k \notin L_N, \\ 0 & \text{if } \hat{x}_k = y_N. \end{cases}$$

(a) Let $f^* = -\infty$. Then if $\hat{x}_k \notin L_N$, we have

$$z_k = \frac{2\alpha}{m}\big(f(\hat{x}_k) - f(y_N)\big) - \alpha^2 C_0^2 \ge \frac{2\alpha}{m}\left(-N+1+\frac{\alpha m C_0^2}{2} + N\right) - \alpha^2 C_0^2 = \frac{2\alpha}{m}.$$

Since $z_k = 0$ for $\hat{x}_k \in L_N$, we have $z_k \ge 0$ for all k, and by (3.3) and the supermartingale convergence theorem, $\sum_{k=0}^{\infty} z_k < \infty$, implying that $\hat{x}_k \in L_N$ for sufficiently large k, with probability 1. Therefore, in the original process we have

$$\inf_{k\ge 0} f(x_k) \le -N + 1 + \frac{\alpha m C_0^2}{2}$$

with probability 1. Letting $N \to \infty$, we obtain $\inf_{k\ge 0} f(x_k) = -\infty$ with probability 1.

(b) Let $f^* > -\infty$. Then if $\hat{x}_k \notin L_N$, we have

$$z_k = \frac{2\alpha}{m}\big(f(\hat{x}_k) - f(y_N)\big) - \alpha^2 C_0^2 \ge \frac{2\alpha}{m}\left(f^* + \frac{2}{N} + \frac{\alpha m C_0^2}{2} - f^* - \frac{1}{N}\right) - \alpha^2 C_0^2 = \frac{2\alpha}{mN}.$$

Hence, $z_k \geq 0$ for all k, and by the supermartingale convergence theorem, we have $\sum_{k=0}^{\infty} z_k < \infty$ implying that $\hat{x}_k \in L_N$ for sufficiently large k, so that in the original process

$$\inf_{k\geq 0} f(x_k) \leq f^* + \frac{2}{N} + \frac{\alpha m C_0^2}{2}$$

with probability 1. Letting $N \to \infty$, we obtain $\inf_{k\geq 0} f(x_k) \leq f^* + \alpha m C_0^2/2$. □

From Proposition 3.1(b), it can be seen that when $f^* > -\infty$, the randomized method (3.1) with a fixed stepsize has a better error bound (by a factor m, since $C^2 \approx m^2 C_0^2$) than the one of the nonrandomized method (1.4)–(1.6) with the same stepsize (cf. Proposition 2.1). This indicates that when randomization is used, the stepsize α_k should generally be chosen larger than in the nonrandomized methods of section 2. This can also be observed from our experimental results. Being able to use a larger stepsize suggests a potential rate of convergence advantage in favor of the randomized methods, which is consistent with our experimental results. A more precise result is shown in Nedić and Bertsekas [NeB00]: given any $\epsilon > 0$, by using $m\big(dist(x_0, X^*)\big)^2/\alpha\epsilon$ iterations of the nonrandomized method we are guaranteed a cost function value that is within a tolerance $(\alpha m^2 C_0^2 + \epsilon)/2$ from the optimum f^*, while by using the same *expected* number of iterations of the randomized method we are guaranteed a cost function value that is within the potentially much smaller tolerance $(\alpha m C_0^2 + \epsilon)/2$ from f^*.

3.0.6. Diminishing stepsize rule. As mentioned earlier, the randomized method (3.1) with a diminishing stepsize can be viewed as a special case of a stochastic subgradient method. Consequently, we just state the main convergence result and refer to the literature for its proof.

PROPOSITION 3.2. *Let Assumption* 3.1 *hold and let the optimal set* X^* *be nonempty. Also assume that the stepsize* α_k *in* (3.1) *is such that*

$$\alpha_k > 0, \qquad \sum_{k=0}^{\infty} \alpha_k = \infty, \qquad \sum_{k=0}^{\infty} \alpha_k^2 < \infty.$$

Then the sequence $\{x_k\}$ *generated by the randomized method* (3.1) *converges to some optimal solution with probability* 1.

Proof. See Theorem 1 of Ermoliev [Erm69] (also [Erm76, p. 97], [Erm83]). □

3.0.7. Dynamic stepsize rule for known f^*. One possible version of the dynamic stepsize rule for the method (3.1) has the form

$$\alpha_k = \gamma_k \frac{f(x_k) - f^*}{m C_0^2}, \qquad 0 < \underline{\gamma} \leq \gamma_k \leq \overline{\gamma} < 2,$$

where $\{\gamma_k\}$ is a deterministic sequence, and requires knowledge of the cost function value $f(x_k)$ at the current iterate x_k. However, it would be inefficient to compute $f(x_k)$ at each iteration since that iteration involves a single component f_i, while the computation of $f(x_k)$ requires all the components. We thus modify the dynamic stepsize rule so that the value of f and the parameter γ_k that are used in the stepsize formula are updated every M iterations, where M is any fixed positive integer, rather than at each iteration. In particular, assuming f^* is known, we use the stepsize

$$\alpha_k = \gamma_p \frac{f(x_{Mp}) - f^*}{m M C_0^2},$$

$$0 < \underline{\gamma} \leq \gamma_p \leq \overline{\gamma} < 2, \quad k = Mp, \ldots, M(p+1) - 1, \quad p = 0, 1, \ldots, \tag{3.4}$$

where $\{\gamma_p\}$ is a deterministic sequence. We can choose M greater than m if m is relatively small, or we can select M smaller than m if m is very large.

PROPOSITION 3.3. *Let Assumption* 3.1 *hold and let* X^* *be nonempty. Then the sequence* $\{x_k\}$ *generated by the randomized method* (3.1) *with the stepsize* (3.4) *converges to some optimal solution with probability* 1.

Proof. By adapting Lemma 2.1 to the case where $y = x^* \in X^*$ and f is replaced by f_{ω_k}, we have

$$||x_{k+1} - x^*||^2 \leq ||x_k - x^*||^2 - 2\alpha_k\big(f_{\omega_k}(x_k) - f_{\omega_k}(x^*)\big) + \alpha_k^2 C_0^2 \qquad \forall\, x^* \in X^*, \quad k \geq 0.$$

By summing this inequality over $k = Mp, \ldots, M(p+1)-1$ (i.e., over the M iterations of a cycle), we obtain for all $x^* \in X^*$ and all p

$$||x_{M(p+1)} - x^*||^2 \leq ||x_{Mp} - x^*||^2 - 2\alpha_{Mp} \sum_{k=Mp}^{M(p+1)-1} \big(f_{\omega_k}(x_k) - f_{\omega_k}(x^*)\big) + M\alpha_{Mp}^2 C_0^2,$$

since $\alpha_k = \alpha_{Mp}$ for $k = Mp, \ldots, M(p+1)-1$. By taking the conditional expectation with respect to $\mathcal{G}_p = \{x_0, \ldots, x_{M(p+1)-1}\}$, we have for all $x^* \in X^*$ and p

$$\begin{aligned} E\big\{||x_{M(p+1)} - x^*||^2 \mid \mathcal{G}_p\big\} &\leq ||x_{Mp} - x^*||^2 \\ &\quad - 2\alpha_{Mp} \sum_{k=Mp}^{M(p+1)-1} E\big\{f_{\omega_k}(x_k) - f_{\omega_k}(x^*) \mid x_k\big\} \\ &\quad + M^2\alpha_{Mp}^2 C_0^2 \leq ||x_{Mp} - x^*||^2 \\ &\quad - \frac{2\alpha_{Mp}}{m} \sum_{k=Mp}^{M(p+1)-1} \big(f(x_k) - f^*\big) + M^2\alpha_{Mp}^2 C_0^2. \end{aligned} \tag{3.5}$$

We now relate $f(x_k)$ and $f(x_{Mp})$ for $k = Mp, \ldots, M(p+1)-1$. We have

$$\begin{aligned} f(x_k) - f^* &= \big(f(x_k) - f(x_{Mp})\big) + \big(f(x_{Mp}) - f^*\big) \\ &\geq \tilde{g}_{Mp}'(x_k - x_{Mp}) + f(x_{Mp}) - f^* \\ &\geq f(x_{Mp}) - f^* - mC_0||x_k - x_{Mp}||, \end{aligned} \tag{3.6}$$

where $\tilde{g}_{Mp}$ is a subgradient of f at x_{Mp} and in the last inequality we use the fact

$$||\tilde{g}_{Mp}|| = \left\|\sum_{i=1}^{m} \tilde{g}_{i,Mp}\right\| \leq mC_0$$

(cf. Assumption 3.1(b)) with $\tilde{g}_{i,Mp}$ being a subgradient of f_i at x_{Mp}. Furthermore,

we have for all p and $k = Mp, \dots, M(p+1)-1$

$$\begin{aligned} ||x_k - x_{Mp}|| &\le \|x_k - x_{k-1}\| + \|x_{k-1} - x_{Mp}\| \\ &\le \alpha_{k-1}\|g(\omega_{k-1}, x_{k-1})\| + \|x_{k-1} - x_{Mp}\| \\ &\le \cdots \\ &\le \alpha_{Mp} \sum_{l=Mp}^{k-1} ||g(\omega_l, x_l)|| \\ &\le \big(k - Mp\big)\alpha_{Mp} C_0, \end{aligned} \tag{3.7}$$

which when substituted in (3.6) yields

$$f(x_k) - f^* \ge f(x_{Mp}) - f^* - \big(k - Mp\big) m \alpha_{Mp} C_0^2.$$

From the preceding relation and (3.5) we have

$$\begin{aligned} E\big\{||x_{M(p+1)} - x^*||^2 \mid \mathcal{G}_{p+1}\big\} \le ||x_{Mp} - x^*||^2 - \frac{2M\alpha_{Mp}}{m}\big(f(x_{Mp}) - f^*\big) \\ +2\alpha_{Mp}^2 C_0^2 \sum_{k=Mp}^{M(p+1)-1} \big(k - Mp\big) + M\alpha_{Mp}^2 C_0^2. \end{aligned} \tag{3.8}$$

Since

$$2\alpha_{Mp}^2 C_0^2 \sum_{k=Mp}^{M(p+1)-1} \big(k - Mp\big) + M\alpha_{Mp}^2 C_0^2 = 2\alpha_{Mp}^2 C_0^2 \sum_{l=1}^{M-1} l + M\alpha_{Mp}^2 C_0^2 = M^2 \alpha_{Mp}^2 C_0^2,$$

it follows that for all $x^* \in X^*$ and p

$$E\big\{||x_{M(p+1)} - x^*||^2 \mid \mathcal{G}_p\big\} \le ||x_{Mp} - x^*||^2 - \frac{2M\alpha_{Mp}}{m}\big(f(x_{Mp}) - f^*\big) + M^2\alpha_{Mp}^2 C_0^2.$$

This relation and the definition of α_k (cf. (3.4)) yield

$$E\big\{||x_{M(p+1)} - x^*||^2 \mid \mathcal{G}_p\big\} \le ||x_{Mp} - x^*||^2 - \gamma_p\big(2 - \gamma_p\big)\left(\frac{f(x_{Mp}) - f^*}{mC_0}\right)^2.$$

By the supermartingale convergence theorem, we have

$$\sum_{k=0}^{\infty} \gamma_p\big(2 - \gamma_p\big)\left(\frac{f(x_{Mp}) - f^*}{mC_0}\right)^2 < \infty$$

and for each $x^* \in X^*$ the sequence $\{||x_{Mp} - x^*||\}$ is convergent, with probability 1. Because $\gamma_p \in [\underline{\gamma}, \overline{\gamma}] \subset (0, 2)$, it follows that with probability 1

$$\lim_{p\to\infty}\big(f(x_{Mp}) - f^*\big) = 0.$$

Let $\{v_i\}$ be a countable subset of the relative interior $\mathrm{ri}(X^*)$ that is dense in X^*. Such a set exists since $\mathrm{ri}(X^*)$ is a relatively open subset of the affine hull of X^*; an

example of such a set is the intersection of X^* with the set of the form $x^* + \sum_{i=1}^{l} r_i \xi_i$, where $x^* \in X^*, r_1, \ldots, r_l$ are rational numbers, and $\xi_1, \ldots, \xi_l$ are basis vectors for the affine hull of X^*. For each i, let Ω_{v_i} be a set of sample paths such that the sequence $\{||x_{Mp} - v_i||\}$ converges. Then the intersection

$$\Omega = \cap_{i=1}^{\infty} \Omega_{v_i}$$

has probability 1, since its complement $\overline{\Omega}$ is equal to $\cup_{i=1}^{\infty} \overline{\Omega}_{v_i}$ and

$$P\Big(\cup_{i=1}^{\infty} \overline{\Omega}_{v_i} \Big) \leq \sum_{i=1}^{\infty} P(\overline{\Omega}_{v_i}) = 0.$$

For each sample path in Ω, the sequence $\{||x_{Mp} - v_i||\}$ converges for all i, so that $\{x_{Mp}\}$ is bounded. Since $f(x_{Mp}) \to f^*$ and f is continuous, all limit points of $\{x_{Mp}\}$ belong to X^*. Because $\{v_i\}$ is a dense subset of X^* and the sequences $\{||x_{Mp} - v_i||\}$ converge, $\{x_{Mp}\}$ must have a unique limit point and hence converges to some $\overline{x} \in X^*$. □

3.0.8. Dynamic stepsize rule for unknown f^*. In the case where f^* is not known, we modify the dynamic stepsize (3.4) by replacing f^* with a target level estimate f_p^{lev}. Thus the stepsize is

$$\alpha_k = \gamma_p \frac{f(x_{Mp}) - f_p^{\text{lev}}}{mMC_0^2}, \tag{3.9}$$

$$0 < \underline{\gamma} \leq \gamma_p \leq \overline{\gamma} < 2, \quad k = Mp, \ldots, M(p+1) - 1, \quad p = 0, 1, \ldots.$$

To update the target values f_p^{lev}, we may use the adjustment procedures described in section 2.

In the first adjustment procedure, f_p^{lev} is given by

$$f_p^{\text{lev}} = \min_{0 \leq j \leq p} f\big(x_{Mj}\big) - \delta_p, \tag{3.10}$$

and δ_p is updated according to

$$\delta_{p+1} = \begin{cases} \delta_p & \text{if } f\big(x_{M(p+1)}\big) \leq f_p^{\text{lev}}, \\ \max\big\{\beta\delta_p, \delta\big\} & \text{if } f\big(x_{M(p+1)}\big) > f_p^{\text{lev}}, \end{cases} \tag{3.11}$$

where δ and β are fixed positive constants with $\beta < 1$. Thus all the parameters of the stepsize are updated every M iterations. Note that here the parameter ρ of (2.11) has been set to 1. Our proof relies on this (relatively mild) restriction. Since the stepsize is bounded away from zero, the method behaves similarly to the one with a constant stepsize (cf. Proposition 3.1). More precisely, we have the following result.

PROPOSITION 3.4. *Let Assumption* 3.1 *hold. Then, for the sequence* $\{x_k\}$ *generated by the randomized method* (3.1) *and the stepsize rule* (3.9) *with the adjustment procedure* (3.10)–(3.11)*, we have the following:*

(a) *If* $f^* = -\infty$*, then with probability* 1

$$\inf_{k \geq 0} f(x_k) = f^*.$$

(b) *If* $f^* > -\infty$*, then with probability* 1

$$\inf_{k \geq 0} f(x_k) \leq f^* + \delta.$$

Proof. (a) Define the events

$$H_1 = \left\{ \lim_{p\to\infty} \delta_p > \delta \right\}, \qquad H_2 = \left\{ \lim_{p\to\infty} \delta_p = \delta \right\}.$$

Given that H_1 occurred there is an integer R such that $\delta_R > \delta$ and

$$\delta_p = \delta_R \qquad \forall\ p \geq R.$$

We let R be the smallest integer with the above property and we note that R is a discrete random variable taking nonnegative integer values. In view of (3.11), we have for all $p \geq R$

$$f\big(x_{M(p+1)}\big) \leq f_p^{\rm lev}.$$

Then from the definition of $f_p^{\rm lev}$ (cf. (3.10)), the relation $\min_{0\leq j\leq p} f\big(x_{Mj}\big) \leq f\big(x_{Mp}\big)$, and the fact $\delta_p = \delta_R$ for all $p \geq R$, we obtain

$$f\big(x_{M(p+1)}\big) \leq f\big(x_{Mp}\big) - \delta_R \qquad \forall\ p \geq R.$$

Summation of the above inequalities yields

$$f\big(x_{Mp}\big) \leq f(x_{MR}) - (p-R)\delta_R \qquad \forall\ p \geq R.$$

Therefore, given that H_1 occurred, we have $\inf_{p\geq 0} f(x_{Mp}) \geq \inf_{p\geq 0} f(x_{Mp}) = -\infty$ with probability 1, i.e.,

$$P\left\{ \inf_{p\geq 0} f(x_{Mp}) = -\infty \;\Big|\; H_1 \right\} = 1. \tag{3.12}$$

Now assume that H_2 occurred. The event H_2 occurs if and only if, after finitely many iterations, δ_p is decreased to the threshold value δ and remains at that value for all subsequent iterations. Thus H_2 occurs if and only if there is an index S such that

$$\delta_p = \delta \qquad \forall\ p \geq S. \tag{3.13}$$

Let S be the smallest integer with the above property, and note that we have $H_2 = \cup_{s\geq 0} B_s$, where $B_s = \big\{S = s\big\}$ for all integers $s \geq 0$.

Similar to the proof of Proposition 3.3 (cf. (3.8)), we have for all $y \in X$ and p

$$\begin{aligned} E\big\{||x_{M(p+1)} - y||^2 \mid \mathcal{G}_p, B_s\big\} &= E\big\{||x_{M(p+1)} - y||^2 \mid \mathcal{G}_p\big\} \\ &\leq ||x_{Mp} - y||^2 - 2\gamma_p \frac{f(x_{Mp}) - f_p^{\rm lev}}{m^2 C_0^2}\big(f(x_{Mp}) - f(y)\big) \\ &\quad + \gamma_p^2 \frac{\big(f(x_{Mp}) - f_p^{\rm lev}\big)^2}{m^2 C_0^2}, \end{aligned} \tag{3.14}$$

where $\mathcal{G}_p = \{x_0, \ldots, x_{Mp-1}\}$. Now, fix an N and let $y_N \in X$ be such that

$$f(y_N) = -N - \delta,$$

where N is a nonnegative integer. Consider a new process $\{\hat{x}_k\}$ defined by

$$\hat{x}_{k+1} = \begin{cases} \mathcal{P}_X\big[\hat{x}_k - \alpha_k g(\omega_k, \hat{x}_k)\big] & \text{if } f(\hat{x}_{Mp}) \geq -N, \\ y_N & \text{otherwise} \end{cases}$$

for $k = Mp, \ldots, M(p+1)-1,\ p = 0, 1, \ldots$, and $\hat{x}_0 = x_0$. The process $\{\hat{x}_k\}$ is identical to $\{x_k\}$ up to the point when x_{Mp} enters the level set

$$L_N = \{x \in X \mid f(x) < -N\},$$

in which case the process $\{\hat{x}_k\}$ terminates at the point y_N. Therefore, given B_s, the process $\{\hat{x}_{Mp}\}$ satisfies (3.14) for all $p \geq s$ and $y = y_N$, i.e., we have

$$E\big\{||\hat{x}_{M(p+1)} - y_N||^2 \mid \mathcal{G}_p\big\} \leq ||\hat{x}_{Mp} - y_N||^2 - 2\gamma_p \frac{f(\hat{x}_{Mp}) - f_p^{\text{lev}}}{m^2 C_0^2}\big(f(\hat{x}_{Mp}) - f(y_N)\big) + \gamma_p^2 \frac{\big(f(\hat{x}_{Mp}) - f_p^{\text{lev}}\big)^2}{m^2 C_0^2},$$

or equivalently

$$E\big\{||\hat{x}_{M(p+1)} - y_N||^2 \mid \mathcal{G}_p\big\} \leq ||\hat{x}_{Mp} - y_N||^2 - z_p,$$

where

$$z_p = \begin{cases} 2\gamma_p \dfrac{f(\hat{x}_{Mp}) - f_p^{\text{lev}}}{m^2 C_0^2}\big(f(\hat{x}_{Mp}) - f(y_N)\big) - \gamma_p^2 \dfrac{\big(f(\hat{x}_{Mp}) - f_p^{\text{lev}}\big)^2}{m^2 C_0^2} & \text{if } \hat{x}_{Mp} \notin L_N, \\ 0 & \text{if } \hat{x}_{Mp} = y_N. \end{cases}$$

By using the definition of f_p^{lev} (cf. (3.10)) and the fact $\delta_p = \delta$ for all $p \geq s$ (cf. (3.13)), we have for $p \geq s$ and $\hat{x}_{Mp} \notin L_N$

$$f(y_N) \leq \min_{0 \leq j \leq p} f(\hat{x}_{Mj}) - \delta = f_p^{\text{lev}},$$

which, when substituted in the preceding relation, yields for $p \geq s$ and $\hat{x}_{Mp} \notin L_N$

$$z_p \geq \gamma_p\big(2 - \gamma_p\big)\frac{\big(f(\hat{x}_{Mp}) - f_p^{\text{lev}}\big)^2}{m^2 C_0^2} \geq \underline{\gamma}(2 - \overline{\gamma})\frac{\delta^2}{m^2 C_0^2}.$$

The last inequality above follows from the facts $\gamma_p \in [\underline{\gamma}, \overline{\gamma}]$ and $f(\hat{x}_{Mp}) - f_p^{\text{lev}} \geq \delta$ for all p (cf. (3.10)–(3.11)). Hence $z_p \geq 0$ for all k, and by the supermartingale convergence theorem, we obtain $\sum_{p=s}^{\infty} z_p < \infty$ with probability 1. Thus, given B_s we have $\hat{x}_{Mp} \in L_N$ for sufficiently large p, with probability 1, implying that in the original process

$$P\left\{\inf_{p \geq 0} f(x_{Mp}) \leq -N \;\middle|\; B_s\right\} = 1.$$

By letting $N \to \infty$ in the preceding relation, we obtain

$$P\left\{\inf_{p \geq 0} f(x_{Mp}) = -\infty \;\middle|\; B_s\right\} = 1.$$

Since $H_2 = \cup_{s\geq 0} B_s$, it follows that

$$P\left\{\inf_{p\geq 0} f(x_{Mp}) = -\infty \;\middle|\; H_2\right\} = \sum_{s=0}^{\infty} P\left\{\inf_{p\geq 0} f(x_{Mp}) = -\infty \;\middle|\; B_s\right\} P(B_s)$$
$$= \sum_{s=0}^{\infty} P(B_s) = 1.$$

Combining (3.12) with the preceding relation, we have with probability 1

$$\inf_{p\geq 0} f(x_{Mp}) = -\infty,$$

so that $\inf_{k\geq 0} f(x_k) = -\infty$ with probability 1.

(b) Using the proof of part (a), we see that if $f^* > -\infty$, then H_2 occurs with probability 1. Thus, as in part (a), we have $H_2 = \cup_{s\geq 0} B_s$, where $B_s = \{S = s\}$ for all integer $s \geq 0$ and S is as in (3.13).

Fix an N and let $y_N \in X$ be such that

$$f(y_N) = f^* + \frac{1}{N},$$

where N is a positive integer. Consider the process $\{\hat{x}_k\}$ defined by

$$\hat{x}_{k+1} = \begin{cases} \mathcal{P}_X\left[\hat{x}_k - \alpha_k g(\omega_k, \hat{x}_k)\right] & \text{if } f(\hat{x}_{Mp}) \geq f^* + \delta + \frac{1}{N}, \\ y_N & \text{otherwise} \end{cases}$$

for $k = Mp, \ldots, M(p+1)-1$, $p = 0, 1, \ldots$, and $\hat{x}_0 = x_0$. The process $\{\hat{x}_k\}$ is the same as the process $\{x_k\}$ up to the point where x_{Mp} enters the level set

$$L_N = \left\{x \in X \;\middle|\; f(x) < f^* + \delta + \frac{1}{N}\right\},$$

in which case the process $\{\hat{x}_k\}$ terminates at the point y_N. The rest follows similarly to the proof of part (a). □

The target level f_p^{lev} can also be updated according to the second adjustment procedure discussed in section 2. In this case, it can be shown that the result of Proposition 2.7 holds with probability 1. We omit the lengthy details.

4. Experimental results. In this section we report some of the numerical results with a certain type of test problem: the dual of a generalized assignment problem (see Martello and Toth [MaT90, p. 189], and Bertsekas [Ber98, p. 362]. The problem is to assign m jobs to n machines. If job i is performed at machine j, it costs a_{ij} and requires p_{ij} time units. Given the total available time t_j at machine j, we want to find the minimum cost assignment of the jobs to the machines. Formally the problem is

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij} y_{ij} \\ \text{subject to} \quad & \sum_{j=1}^{n} y_{ij} = 1, \quad i = 1, \ldots, m, \\ & \sum_{i=1}^{m} p_{ij} y_{ij} \leq t_j, \quad j = 1, \ldots, n, \\ & y_{ij} = 0 \text{ or } 1, \quad \text{for all } i, j, \end{aligned}$$

where y_{ij} is the assignment variable, which is equal to 1 if the ith job is assigned to the jth machine and is equal to 0 otherwise. In our experiments we chose n equal to 4 and m equal to the four values 500, 800, 4000, and 7000.

By relaxing the time constraints for the machines, we obtain the dual problem

$$\begin{aligned} &\text{maximize} \quad f(x) = \sum_{i=1}^{m} f_i(x) \\ &\text{subject to} \quad x \geq 0, \end{aligned} \tag{4.1}$$

where

$$f_i(x) = \min_{\sum_{j=1}^{n} y_{ij}=1,\ y_{ij}=0 \text{ or } y_{ij}=1} \sum_{j=1}^{n} (a_{ij} + x_j p_{ij}) y_{ij} - \frac{1}{m} \sum_{j=1}^{n} t_j x_j, \qquad i = 1, \ldots, m.$$

Since $a_{ij} + x_j p_{ij} \geq 0$ for all i, j, we can easily evaluate $f_i(x)$ for each $x \geq 0$:

$$f_i(x) = a_{ij^*} + x_{j^*} p_{ij^*} - \frac{1}{m} \sum_{j=1}^{n} t_j x_j,$$

where j^* is such that

$$a_{ij^*} + x_{j^*} p_{ij^*} = \min_{1 \leq j \leq n} \{a_{ij} + x_j p_{ij}\}.$$

In the same time, at no additional cost, we obtain a subgradient g of f_i at x:

$$g = (g_1, \ldots, g_n)', \qquad g_j = \begin{cases} -\frac{t_j}{m} & \text{if } j \neq j^*, \\ p_{ij^*} - \frac{t_{j^*}}{m} & \text{if } j = j^*. \end{cases}$$

The experiments are divided in two groups, each with a different goal. The first group was designed to compare the performance of the ordinary subgradient method (1.3) and the incremental subgradient method (1.4)–(1.6) for solving the test problem (4.1) when using different stepsize choices while keeping fixed the order of processing of the components f_i. The second group of experiments was designed to evaluate the incremental method when using different rules for the order of processing the components f_i, while keeping fixed the stepsize choice.

In the first group of experiments the data for the problems (i.e., the matrices $\{a_{ij}\}, \{p_{ij}\}$) were generated randomly according to a uniform distribution over different intervals. The values t_j were calculated according to the formula

$$t_j = \frac{\bar{t}}{n} \sum_{i=1}^{m} p_{ij}, \qquad j = 1, \ldots, n, \tag{4.2}$$

with $\bar{t}$ taking one of the three values 0.5, 0.7, or 0.9. We used two stepsize rules:

(1) A diminishing stepsize that has the form

$$\alpha_{kN} = \cdots = \alpha_{(k+1)N-1} = \frac{D}{k+1} \qquad \forall\ k \geq 0,$$

where D is some positive constant, and N is some positive integer that represents the number of cycles during which the stepsize is kept at the same value. To guard

TABLE 1
$n = 4$, $m = 800$, $f^* \approx 1578.47$, $\tilde{f} = 1578$.

Ordinary subgradient method		
Initial point x_0	Diminishing $D/N/S/iter$	Target level $r/\xi/\delta_0/iter$
(0,0,0,0)	$0.08/2/7/ > 500$	$0.03/0.97/12 \times 10^5/ > 500$
(0,0,0,0)	$0.1/2/7/ > 500$	$0.5/0.98/2 \times 10^4/ > 500$
(0,0,0,0)	$0.07/3/10/ > 500$	$0.5/0.95/3 \times 10^4/ > 500$
(0,0,0,0)	$0.01/10/7/ > 500$	$0.3/0.95/5 \times 10^4/ > 400$
(0,0,0,0)	$0.09/1/7/ > 500$	$0.1/0.9/10^6/ > 200$
(0,0,0,0)	$0.03/5/500/ > 500$	$0.2/0.93/5 \times 10^4/ > 300$
(0,0,0,0)	$0.08/4/7/ > 500$	$0.8/0.97/12 \times 10^3/ > 500$
(0,0,0,0)	$0.09/5/10/ > 500$	$0.03/0.95/10^6/ > 500$
(1.2,1.1,2,1.04)	$0.005/2/5/ > 500$	$0.4/0.975/2 \times 10^4/ > 200$
(1.2,1.1,2,1.04)	$0.009/1/5/ > 500$	$0.5/0.97/4 \times 10^3/ > 50$
$(0.4, 0.2, 1.4, 0.1)$	$0.009/2/5/ > 500$	$0.4/0.8/2700/ > 500$
$(0.4, 0.2, 1.4, 0.1)$	$0.005/5/500/ > 500$	$0.5/0.9/1300/ > 500$

against an unduly large value of c we implemented an adaptive feature, whereby if within some (heuristically chosen) number S of consecutive iterations the current best cost function value is not improved, then the new iterate x_{k+1} is set equal to the point at which the current best value is attained.

(2) The stepsize rule given by (2.9) and the path-based procedure. This is essentially the target level method, in which the path bound is not fixed but rather the current value for B is multiplied by a certain factor $\xi \in (0, 1)$ whenever an oscillation is detected (see the remark following Proposition 2.7). The initial value for the path bound was $B = r||x_0 - x_1||$ for some (heuristically chosen) positive constant r.

We report in the following tables the number of iterations required for various methods and parameter choices to achieve a given threshold cost $\tilde{f}$. The notation used in the tables is as follows:

$> k \times 100$ for $k = 1, 2, 3, 4$ means that the value $\tilde{f}$ has been achieved or exceeded after $k \times 100$ iterations, but in less than $(k + 1) \times 100$ iterations.

> 500 means that the value $\tilde{f}$ has not been achieved within 500 iterations.

$D/N/S/iter$ gives the values of the parameters D, N, and S for the diminishing stepsize rule, while $iter$ is the number of iterations (or cycles) needed to achieve or exceed $\tilde{f}$.

$r/\xi/\delta_0/iter$ describes the values of the parameters and number of iterations for the target level stepsize rule.

Tables 1 and 2 show the results of applying the ordinary and incremental subgradient methods to problem (4.1) with $n = 4$, $m = 800$, and $\bar{t} = 0.5$ in (4.2). The optimal value of the problem is $f^* \approx 1578.47$. The threshold value is $\tilde{f} = 1578$. The tables show when the value $\tilde{f}$ was attained or exceeded.

Tables 3 and 4 show the results of applying the ordinary and incremental subgradient methods to problem (4.1) with $n = 4$, $m = 4000$, and $\bar{t} = 0.7$ in (4.2). The optimal value of the problem is $f^* \approx 6832.3$ and the threshold value is $\tilde{f} = 6831.5$. The tables show the number of iterations needed to attain or exceed the value $\tilde{f} = 6831.5$.

Tables 1 and 2 demonstrate that the incremental subgradient method performs substantially better than the ordinary subgradient method. As m increases, the performance of the incremental method improves as indicated in Tables 3 and 4. The results obtained for other problems that we tested are qualitatively similar and con-

TABLE 2
$n = 4,\ m = 800,\ f^* \approx 1578.47,\ \tilde{f} = 1578.$

Incremental subgradient method		
Initial point x_0	Diminishing $D/N/S/iter$	Target level $r/\xi/\delta_0/iter$
(0,0,0,0)	0.05/3/500/99	$3/0.7/5 \times 10^6/97$
(0,0,0,0)	0.09/2/500/ > 100	$2/0.6/55 \times 10^5/ > 100$
(0,0,0,0)	0.1/1/500/99	$0.7/0.8/55 \times 10^5/ > 100$
(0,0,0,0)	0.1/1/10/99	$0.4/0.95/10^7/80$
(0,0,0,0)	0.05/5/7/ > 100	$0.3/0.93/10^7/ > 100$
(0,0,0,0)	0.07/3/10/ > 100	$0.5/0.9/10^7/ > 200$
(0,0,0,0)	0.01/7/7/ > 500	$0.3/0.93/15 \times 10^6/30$
(0,0,0,0)	0.009/5/7/ > 500	$2/0.8/5 \times 10^6/ > 100$
(1.2,1.1,2,1.04)	0.05/1/500/40	$0.4/0.97/12 \times 10^6/ > 100$
(1.2,1.1,2,1.04)	0.04/3/500/35	$0.3/0.975/10^7/27$
(0.4,0.2,1.4,0.1)	0.07/1/500/48	$0.4/0.975/12 \times 10^6/100$
(0.4,0.2,1.4,0.1)	0.048/1/500/39	$0.5/0.94/12 \times 10^6/ > 100$

TABLE 3
$n = 4,\ m = 4000,\ f^* \approx 6832.3,\ \tilde{f} = 6831.5.$

Ordinary subgradient method		
Initial point x_0	Diminishing $D/N/S/iter$	Target level $r/\xi/\delta_0/iter$
(0,0,0,0)	0.01/2/7/ > 500	1/0.9/5000/58
(0,0,0,0)	0.001/5/7/ > 300	2/0.99/5500/ > 100
(0,0,0,0)	0.0008/5/10/ > 300	1.3/0.98/4800/54
(0,0,0,0)	0.0005/5/7/ > 200	1.5/0.98/2000/88
(0,0,0,0)	0.0001/5/10/99	0.5/0.8/4000/99
(0,0,0,0)	0.0001/2/500/ > 100	0.4/0.9/4000/89
(0,0,0,0)	0.0001/5/10/ > 200	0.5/0.9/3000/88
(0,0,0,0)	0.00009/5/500/100	0.5/0.95/2000/98
(0.5,0.9,1.3,0.4)	0.0005/3/500/ > 100	0.5/0.98/2000/95
(0.5,0.9,1.3,0.4)	0.0002/7/7/ > 100	0.4/0.97/3000/98
(0.26,0.1,0.18,0.05)	0.0002/5/7/100	0.3/0.98/3000/90
(0.26,0.1,0.18,0.05)	0.00005/7/7/30	0.095/0.985/10/50

TABLE 4
$n = 4,\ m = 4000,\ f^* \approx 6832.3,\ \tilde{f} = 6831.5.$

Incremental subgradient method		
Initial point x_0	Diminishing $D/N/S/iter$	Target level $r/\xi/\delta_0/iter$
(0,0,0,0)	0.005/2/500/46	$5/0.99/10^6/7$
(0,0,0,0)	0.007/1/500/37	$8/0.97/11 \times 10^5/5$
(0,0,0,0)	0.001/2/500/95	$2/0.99/7 \times 10^5/ > 100$
(0,0,0,0)	0.0008/1/500/30	$0.8/0.4/9 \times 10^5/6$
(0,0,0,0)	0.0002/2/500/21	$0.7/0.4/10^6/7$
(0,0,0,0)	0.0005/2/500/40	$0.1/0.9/10^6/15$
(0,0,0,0)	0.0002/2/7/21	$0.08/0.9/15 \times 10^5/18$
(0,0,0,0)	0.0003/1/500/21	$0.25/0.9/2 \times 10^6/20$
(0.5,0.9,1.3,0.4)	0.001/1/500/40	$0.07/0.9/10^6/7$
(0.5,0.9,1.3,0.4)	0.0004/1/500/30	$0.04/0.9/10^6/26$
(0.26,0.1,0.18,0.05)	0.00045/1/500/20	$0.04/0.9/15 \times 10^5/10$
(0.26,0.1,0.18,0.05)	0.00043/1/7/20	$0.045/0.91/1.55 \times 10^6/10$

TABLE 5
$n = 4$, $m = 800$, $f^* \approx 1672.44$, $\tilde{f} = 1672$.

Incremental subgradient method/Diminishing stepsize			
Initial point x_0	Sorted order $D/N/iter$	Sorted/Shifted order $D/N/K/iter$	Random order $D/N/iter$
(0,0,0,0)	0.005/1/ > 500	0.007/1/9/ > 500	0.0095/4/5
(0,0,0,0)	0.0045/1/ > 500	0.0056/1/13/ > 500	0.08/1/21
(0,0,0,0)	0.003/2/ > 500	0.003/2/7/ > 500	0.085/1/7
(0,0,0,0)	0.002/3/ > 500	0.002/2/29/ > 500	0.091/1/17
(0,0,0,0)	0.001/5/ > 500	0.001/6/31/ > 500	0.066/1/18
(0,0,0,0)	0.006/1/ > 500	0.0053/1/3/ > 500	0.03/2/18
(0,0,0,0)	0.007/1/ > 500	0.00525/1/11/ > 500	0.07/1/18
(0,0,0,0)	0.0009/7/ > 500	0.005/1/17/ > 500	0.054/1/17
(0.2,0.4,0.8,3.6)	0.001/1/ > 500	0.001/1/17/ > 500	0.01/1/13
(0.2,0.4,0.8,3.6)	0.0008/3/ > 500	0.0008/3/7/ > 500	0.03/1/8
(0,0.05,0.5,2)	0.0033/1/ > 400	0.0037/1/7/ > 400	0.033/1/7
(0,0.05,0.5,2)	0.001/4/ > 500	0.0024/2/13/ > 500	0.017/1/8

sistently show substantially and often dramatically faster convergence for the incremental method.

We suspected that the random generation of the problem data induced a behavior of the (nonrandomized) incremental method that is similar to the one of the randomized version. Consequently, for the second group of experiments, the coefficients $\{a_{ij}\}$ and $\{p_{ij}\}$ were generated as before and then were sorted in nonincreasing order, in order to create a sequential dependence among the data. In all runs we used the diminishing stepsize choice (as described earlier) with $S = 500$, while the order of components f_i was changed according to three rules:

(1) *Sorted*. After the data have been randomly generated and sorted, the components are processed in the fixed order $1, 2, \ldots, m$.

(2) *Sorted/Shifted*. After the data have been randomly generated and sorted, they are cyclically shifted by some number K. The components are processed in the fixed order $1, 2, \ldots, m$.

(3) *Random*. The index of the component to be processed is chosen randomly, with each component equally likely to be selected.

To compare fairly the randomized methods with the other methods, we count as an "iteration" the processing of m consecutively and randomly chosen components f_i. In this way, an "iteration" of the randomized method is equally time-consuming as a cycle or "iteration" of any of the nonrandomized methods.

Table 5 shows the results of applying the incremental subgradient method with order rules (1)–(3) for solving the problem (4.1) with $n = 4$, $m = 800$, and $\bar{t} = 0.9$ in (4.2). The optimal value is $f^* \approx 1672.44$ and the threshold value is $\tilde{f} = 1672$. The table shows the number of iterations needed to attain or exceed $\tilde{f}$.

Table 6 shows the results of applying the incremental subgradient method with order rules (1)–(3) for solving the problem (4.1) with $n = 4$, $m = 7000$, and $\bar{t} = 0.5$ in (4.2). The optimal value is $f^* \approx 14601.38$ and the threshold value is $\tilde{f} = 14600$. The tables show when the value $\tilde{f}$ was attained or exceeded.

Tables 5 and 6 show how an unfavorable fixed order can have a dramatic effect on the performance of the incremental subgradient method. Note that shifting the components at the beginning of every cycle did not improve the convergence rate of the method. However, the randomization of the processing order resulted in fast

TABLE 6
$n = 4$, $m = 7000$, $f^* \approx 14601.38$, $\tilde{f} = 14600$.

Incremental subgradient method/Diminishing stepsize			
Initial point x_0	Sorted order $D/N/iter$	Sorted/Shifted order $D/N/K/iter$	Random order $D/N/iter$
(0,0,0,0)	0.0007/1/ > 500	0.0007/1/3/ > 500	0.047/1/18
(0,0,0,0)	0.0006/1/ > 500	0.0006/1/59/ > 500	0.009/1/10
(0,0,0,0)	0.00052/1/ > 500	0.00052/1/47/ > 500	0.008/1/2
(0,0,0,0)	0.0008/1/ > 500	0.0005/1/37/ > 500	0.023/1/34
(0,0,0,0)	0.0004/2/ > 500	0.0004/2/61/ > 500	0.0028/1/10
(0,0,0,0)	0.0003/2/ > 500	0.0003/2/53/ > 500	0.06/1/22
(0,0,0,0)	0.00025/3/ > 500	0.00025/3/11/ > 500	0.05/1/18
(0,0,0,0)	0.0009/1/ > 500	0.00018/3/79/ > 500	0.007/1/10
(0,0.1,0.5,2.3)	0.0005/1/ > 500	0.0005/1/79/ > 500	0.004/1/10
(0,0.1,0.5,2.3)	0.0003/1/ > 500	0.0003/1/51/ > 500	0.0007/1/18
(0,0.2,0.6,3.4)	0.0002/1/ > 500	0.0002/1/51/ > 500	0.001/1/10
(0,0.2,0.6,3.4)	0.0004/1/ > 500	0.00007/2/93/ > 500	0.0006/1/10

convergence. The results for the other problems that we tested are qualitatively similar and also demonstrated the superiority of the randomized method.

5. Conclusions. We have proposed several variants of incremental subgradient methods, we have analyzed their convergence properties, and we have evaluated them experimentally. The methods that employ the constant and the dynamic stepsize rules are analyzed here for the first time. The subgradient methods of section 3 are the first incremental methods that use randomization in the context of deterministic nondifferentiable optimization, and their computational performance is particularly interesting. A similar randomization in the context of deterministic differentiable optimization, proposed by Bertsekas and Tsitsiklis [BeT96, p. 143], seems to have a qualitatively different computational performance, as suggested by examples (see Bertsekas [Ber99, p. 113 and p. 616].

Several of the ideas of this paper merit further investigation, some of which will be presented in future publications. In particular, we will discuss in a separate paper variants of the incremental subgradient method involving a momentum term, alternative stepsize rules, the use of ϵ-subgradients, and some other features.

REFERENCES

[Ber97] D. P. BERTSEKAS, *A new class of incremental gradient methods for least squares problems*, SIAM J. Optim., 7 (1997), pp. 913–926.

[Ber98] D. P. BERTSEKAS, *Network Optimization: Continuous and Discrete Models*, Athena Scientific, Belmont, MA, 1998.

[Ber99] D. P. BERTSEKAS, *Nonlinear Programming*, 2nd ed., Athena Scientific, Belmont, MA, 1999.

[BeT96] D. P. BERTSEKAS AND J. N. TSITSIKLIS, *Neuro-Dynamic Programming*, Athena Scientific, Belmont, MA, 1996.

[BeT00] D. P. BERTSEKAS AND J. N. TSITSIKLIS, *Gradient convergence in gradient methods*, SIAM J. Optim., 10 (2000), pp. 627–642.

[BMN00] A. BEN-TAL, T. MARGALIT, AND A. NEMIROVSKI, *The ordered subsets mirror descent optimization method and its use for the positron emission tomography reconstruction*, in Proceedings of the March 2000 Haifa Workshop on Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications, D. Butnariu, Y. Censor, and S. Reich, eds., Stud. Comput. Math., Elsevier, Amsterdam, to appear.

[Brä93] U. Brännlund, *On Relaxation Methods for Nonsmooth Convex Optimization*, Doctoral Thesis, Royal Institute of Technology, Stockholm, Sweden, 1993.

[CoL93] R. Correa and C. Lemaréchal, *Convergence of some algorithms for convex minimization*, math. program., 62 (1993), pp. 261–275.

[DeV85] V. F. Dem'yanov and L. V. Vasil'ev, *Nondifferentiable Optimization*, Optimization Software, New York, 1985.

[Erm66] Yu. M. Ermoliev, *Methods for solving nonlinear extremal problems*, Kibernet., 4 (1966), pp. 1–17.

[Erm69] Yu. M. Ermoliev, *On the stochastic quasi-gradient method and stochastic quasi-Feyer sequences*, Kibernet., 2 (1969), pp. 73–83.

[Erm76] Yu. M. Ermoliev, *Stochastic Programming Methods*, Nauka, Moscow, 1976.

[Erm83] Yu. M. Ermoliev, *Stochastic quasigradient methods and their application to system optimization*, Stochastics, 9 (1983), pp. 1–36.

[Erm88] Yu. M. Ermoliev, *Stochastic quasigradient methods*, in Numerical Techniques for Stochastic Optimization, Yu. M. Ermoliev and R. J-B. Wets, eds., Springer-Verlag, Berlin, 1988, pp. 141–185.

[Gai94] A. A. Gaivoronski, *Convergence analysis of parallel backpropagation algorithm for neural networks*, Optim. Methods Soft., 4 (1994), pp. 117–134.

[GoK99] J. L. Goffin and K. Kiwiel, *Convergence of a simple subgradient level method*, Math. Program., 85 (1999), pp. 207–211.

[Gri94] L. Grippo, *A class of unconstrained minimization methods for neural network training*, Optim. Methods Soft., 4 (1994), pp. 135–150.

[HiL93] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms*, vols. I and II, Springer-Verlag, Berlin, New York, 1993.

[KaC98] C. A. Kaskavelis and M. C. Caramanis, *Efficient Lagrangian relaxation algorithms for industry size job-shop scheduling problems*, IIE Transactions on Scheduling and Logistics, 30 (1998), pp. 1085–1097.

[Kib80] V. M. Kibardin, *Decomposition into functions in the minimization problem*, Automat. Remote Control, 40 (1980), pp. 1311–1323.

[KiL00] K. C. Kiwiel and P. O. Lindberg, *Parallel subgradient methods for convex optimization*, in Proceedings of the March 2000 Haifa Workshop on Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications, D. Butnariu, Y. Censor, and S. Reich, eds., Stud. Comput. Math., Elsevier, Amsterdam, to appear.

[Luo91] Z. Q. Luo, *On the Convergence of the LMS algorithm with adaptive learning rate for linear feedforward networks*, Neural Computation, 3 (1991), pp. 226–245.

[LuT94] Z. Q. Luo and P. Tseng, *Analysis of an approximate gradient projection method with applications to the backpropagation algorithm*, Optim. Methods Softw., 4 (1994), pp. 85–101.

[MaS94] O. L. Mangasarian and M. V. Solodov, *Serial and parallel backpropagation convergence via nonmonotone perturbed minimization*, Optim. Methods Softw., 4 (1994), pp. 103–116.

[MaT90] S. Martello and P. Toth, *Knapsack Problems*, J. Wiley, New York, 1990.

[Min86] M. Minoux, *Mathematical Programming: Theory and Algorithms*, J. Wiley, New York, 1986.

[NBB00] A. Nedić, D. P. Bertsekas, and V. S. Borkar, *Distributed asynchronous incremental subgradient methods*, in Proceedings of the March 2000 Haifa Workshop on Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications, D. Butnariu, Y. Censor, and S. Reich, eds., Studies Comput. Math., Elsevier, Amsterdam, to appear.

[NeB99] A. Nedić and D. P. Bertsekas, *Incremental Subgradient Methods for Nondifferentiable Optimization*, Lab. for Info. and Decision Systems report LIDS-P-2460, Massachusetts Institute of Technology, Cambridge, MA, 1999.

[NeB00] A. Nedić and D. P. Bertsekas, *Convergence rate of incremental subgradient algorithms*, in Stochastic Optimization: Algorithms and Applications, S. Uryasev and P. M. Pardalos, eds., to appear.

[Pol67] B. T. Polyak, *A general method of solving extremum problems*, Soviet Math. Doklady, 8 (1967), pp. 593–597.

[Pol69] B. T. Polyak, *Minimization of unsmooth functionals*, Z. Vychisl. Mat. i Mat. Fiz., 9 (1969), pp. 509–521.

[Pol87] B. T. Polyak, *Introduction to Optimization*, Optimization Software, Inc., New York, 1987.

[Roc70] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[Sho85] N. Z. Shor, *Minimization Methods for Nondifferentiable Functions*, Springer-Verlag, Berlin, 1985.

[Sol98] M. V. Solodov, *Incremental gradient algorithms with stepsizes bounded away from zero*, Comput. Opt. Appl., 11 (1998), pp. 28–35.

[SoZ98] M. V. Solodov and S. K. Zavriev, *Error stability properties of generalized gradient-type algorithms*, J. Optim. Theory Appl., 98 (1998), pp. 663–680.

[Tse98] P. Tseng, *An incremental gradient(-projection) method with momentum term and adaptive stepsize rule*, SIAM J. Optim., 8 (1998), pp. 506–531.

[WiH60] B. Widrow and M. E. Hoff, *Adaptive switching circuits*, in Institute of Radio Engineers, Western Electronic Show and Convention, convention record, part 4, 1960, pp. 96–104.

[ZLW99] X. Zhao, P. B. Luh, and J. Wang, *Surrogate gradient algorithm for Lagrangian relaxation*, J. Opt. Theory Appl., 100 (1999), pp. 699–712.

SIAM J. OPTIM.
Vol. 12, No. 1, pp. 139–169

ON SECOND-ORDER SUBDIFFERENTIALS AND THEIR APPLICATIONS*

BORIS S. MORDUKHOVICH† AND JIŘÍ V. OUTRATA‡

Abstract. We study second-order subdifferentials of nonsmooth functions that are particularly important for applications to sensitivity analysis in optimization and related problems. First we develop various calculus rules for these subdifferentials in rather general settings. Then we obtain exact formulas for computing the second-order subdifferentials for a class of separable piecewise smooth functions. Functions of this class arise, in particular, in equilibrium models related to some practical problems of continuum mechanics. Finally we provide applications of the obtained results to Lipschitzian stability of parametric variational and hemivariational inequalities and efficiently express the derived conditions in terms of the initial data for selected problems of continuum mechanics.

Key words. variational analysis, Lipschitzian stability in optimization, second-order subdifferentials, calculus rules, piecewise smooth functions, variational and hemivariational inequalities, mechanical equilibrium

AMS subject classifications. 49J52, 49K40, 58C20

PII. S1052623400377153

1. Introduction.

The paper is devoted to the theory and applications of second-order subdifferentials in variational analysis. This rapidly growing area has drawn much attention during recent years, motivated by applications to problems in optimization, control, sensitivity, etc. There are several generalized second-order differential constructions for nonsmooth functions useful in variational analysis; see excellent expositions and references in the recent books by Rockafellar and Wets [23] and Bonnans and Shapiro [2]. The variety of such objects is not surprising since even in the classical analysis there are at least two approaches to second-order differentiation; one of them is based on second-order expansions of a function and the other defines a second-order derivative as a first-order derivative of a gradient mapping.

The primary object of this paper is the second-order subdifferential of extended-real-valued functions introduced in Mordukhovich [11] as the coderivative of the first-order subdifferential mapping; see section 2 for more details. This approach follows the latter of the two classical developments in view of the fact that the coderivative can be treated as a derivative-like concept for set-valued mappings (multifunctions). The main motivation for introducing such a second-order subdifferential came from applications to sensitivity analysis for optimization-related problems. In particular, this construction was applied in [13] and [14] to the study of robust Lipschitzian stability of solution maps to parametric variational inequalities in Robinson's framework of generalized equations (GEs) [21]. Let us mention more recent applications of this and associated constructions to complete characterizations of strong regularity for varia-

*Received by the editors August 25, 2000; accepted for publication (in revised form) February 5, 2001; published electronically July 2, 2001.
http://www.siam.org/journals/siopt/12-1/37715.html

†Department of Mathematics, Wayne State University, Detroit, MI 48202 (boris@math.wayne.edu). The research of this author was partly supported by the National Science Foundation under grants DMS-9704751 and DMS-0072179 and also by the Distinguished Faculty Fellowship at Wayne State University.

‡Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, 18208 Prague, Czech Republic (outrata@utia.cas.cz). The research of this author was partly supported by grant 1075005/00 of the Czech Academy of Sciences and by the NATO Science Fellowship Program 16 (2000).

tional inequalities over convex polyhedra in [4], to second-order characterizations of stable optimal solutions to nonsmooth optimization problems in [20] and [6], and to necessary optimality conditions obtained in [17], [18], [19], [24], [25], [26], and [27] for various problems of hierarchical optimization unified under the name of mathematical programs with equilibrium constraints [7].

For further developments and implementations of these results and for their extensions to other classes of optimization-related problems, one needs to have calculus rules for second-order subdifferentials and to be able to compute them efficiently for important classes of nonsmooth functions. Both of these issues are addressed in this paper. Moreover, we are doing this in parallel not only for the basic second-order subdifferential discussed above but also for its "semiconvex" counterpart defined by applying the coderivative to the convexified first-order subdifferential mapping; see section 2. Finally we present applications of the results obtained to stability analysis for parametric GEs with their specific implementations in equilibrium models for some practical problems of continuum mechanics.

The rest of the paper is organized as follows. Section 2 contains basic definitions and required preliminary material widely used in what follows. Section 3 is devoted to calculus rules for both second-order subdifferentials under consideration. We obtain several sum and chain rules for the second-order subdifferentials of rather general nonsmooth functions. Based on the corresponding calculus results for coderivatives and first-order subdifferentials, we restrict ourselves to classes of functions for which the first-order subdifferential rules hold as equalities. This seems to be natural for second-order analysis and allows us to cover a variety of nonsmooth functions important for applications.

In section 4 we efficiently compute the second-order subdifferentials for a class of separable piecewise C^2 functions. Functions of this class frequently appear in the study of various equilibrium problems. They are particularly important for the modeling of some mechanical equilibria considered in this paper. The concluding section 5 presents applications of the main results to Lipschitzian stability of solutions maps to parametric GEs and efficiently expresses the derived conditions in terms of the initial data for selected problems of continuum mechanics.

Our notation is basically standard. Let us mention that Diag (a) denotes a diagonal matrix with vector a at its diagonal; A^* stands for the adjoint (transpose) matrix to A; $\mathbb{B}$ is the closed unit ball of the space in question; x_i is the ith component of vector $x \in \mathbb{R}^n$; $x \bullet y$ is the Hadamard product of $x,\ y \in \mathbb{R}^n$, i. e., $(x \bullet y)_i = x_i y_i$; *cl*, *co*, and *cone* signify the closure, the convex hull, and the conic hull of a set, respectively;

$$\begin{aligned}\operatorname{Lim\,sup}_{x\to\bar{x}} F(x) := \{y \in \mathbb{R}^m |\ \exists\ \text{sequences}\ \ x_k \to \bar{x},\ y_k \to y \\ \text{with}\ \ y_k \in F(x_k),\ k = 1, 2, \ldots\}\end{aligned}$$

connotes the Painlevé–Kuratowski upper (outer) limit for a multifunction $F : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$ as $x \to \bar{x}$.

2. Basic definitions and preliminaries. Let us start with the definitions of our basic *first-order* generalized differential constructions for sets, set-valued mappings, and extended-real-valued functions that appeared in [8] and [9]. We refer the reader to [10], [12], and [23] for equivalent representations and comprehensive theories of these objects.

Given a nonempty set $\Omega \subset \mathbb{R}^n$, we consider the Euclidean projector

$$\Pi(x;\Omega) := \{\omega \in cl\,\Omega |\ \ \|x - \omega\| = \operatorname{dist}(\mathrm{x};\Omega)\}$$

of $x \in \mathbb{R}^n$ on $cl\,\Omega$ and define the *normal cone* to Ω at $\bar{x} \in \Omega$ by

$$N(\bar{x};\Omega) := \operatorname*{Lim\,sup}_{x \to \bar{x}} [\operatorname{cone}(x - \Pi(x;\Omega))]. \tag{2.1}$$

Given a multifunction $F : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$, we define its *coderivative* $D^*F(\bar{x},\bar{y}) : \mathbb{R}^m \rightrightarrows \mathbb{R}^n$ at $(\bar{x},\bar{y}) \in gph\,F$ by

$$D^*F(\bar{x},\bar{y})(y^*) := \{x^* |\ (x^*,-y^*) \in N((\bar{x},\bar{y}); gph\,F)\}, \tag{2.2}$$

where $\bar{y}$ is omitted if F is single-valued at $\bar{x}$.

Given an extended-real-valued function $\varphi : \mathbb{R}^n \to \overline{\mathbb{R}} := [-\infty,\infty]$ finite at $\bar{x}$, we consider the associated epigraphical multifunction

$$E_\varphi(x) := \{\mu \in \mathbb{R}^n |\ \mu \geq \varphi(x)\}$$

and define the *basic subdifferential* and the *singular subdifferential* of φ at $\bar{x}$ by, respectively,

$$\partial\varphi(\bar{x}) := D^*E_\varphi(\bar{x},\varphi(\bar{x}))(1) \ \text{ and } \partial^\infty\varphi(\bar{x}) := D^*E_\varphi(\bar{x},\varphi(\bar{x}))(0). \tag{2.3}$$

If φ is lower semicontinuous (l.s.c.) around $\bar{x}$, then the basic subdifferential $\partial\varphi(\bar{x})$ admits the representation

$$\partial\varphi(\bar{x}) = \operatorname*{Lim\,sup}_{x \overset{\varphi}{\to} \bar{x}} \widehat{\partial}\varphi(x), \tag{2.4}$$

where $x \overset{\varphi}{\to} \bar{x}$ means that $x \to \bar{x}$ with $\varphi(x) \to \varphi(\bar{x})$ and

$$\widehat{\partial}\varphi(x) := \{x^* \in \mathbb{R}^n |\ \liminf_{u \to x} \frac{\varphi(u) - \varphi(x) - \langle x^*, u - x\rangle}{\|u - x\|} \geq 0\}.$$

Note the relationship

$$D^*f(\bar{x})(y^*) = \partial\langle y^*, f\rangle(\bar{x}) \ \ \forall y^* \in \mathbb{R}^m \tag{2.5}$$

between the coderivative (2.2) of a single-valued and locally Lipschitzian mapping $f : \mathbb{R}^n \to \mathbb{R}^m$ and the basic subdifferential (2.3) of the *scalarization* $\langle y^*, f\rangle(x) := \langle y^*, f(x)\rangle$. It follows from (2.5) that $D^*f(\bar{x})(y^*) = \{(\nabla f(\bar{x}))^* y^*\}$ if f is *strictly differentiable* at $\bar{x}$, where $\nabla f(\bar{x})$ stands for the Jacobian matrix.

We also consider the (Clarke) *convexified subdifferential* of φ at $\bar{x}$ that can be defined as

$$\overline{\partial}\varphi(\bar{x}) := \{x^* \in \mathbb{R}^n |\ (x^*,-1) \in clco\,N((\bar{x},\varphi(\bar{x})); epi\,\varphi)\}$$

(cf. [3] and [23]) and admits the equivalent representation

$$\overline{\partial}\varphi(\bar{x}) = clco\,[\partial\varphi(\bar{x}) + \partial^\infty\varphi(\bar{x})]. \tag{2.6}$$

If φ is Lipschitz continuous around $\bar{x}$, then $\partial^\infty\varphi(\bar{x}) = \{0\}$ and $\partial\varphi(\bar{x})$ is bounded, which implies $\overline{\partial}\varphi(\bar{x}) = co\,\partial\varphi(\bar{x})$ due to (2.6). Note the symmetry property

$$\overline{\partial}(-\varphi)(\bar{x}) = -\overline{\partial}\varphi(\bar{x}) \ \text{ if } \varphi \text{ is locally Lipschitz.} \tag{2.7}$$

It follows from (2.6) and (2.4) that

$$\widehat{\partial}\varphi(\bar{x}) \subset \partial\varphi(\bar{x}) \subset \overline{\partial}\varphi(\bar{x}).$$

The function φ is called *subdifferentially regular* at $\bar{x}$ if $\partial\varphi(\bar{x}) = \widehat{\partial}\varphi(\bar{x})$; see [10] and [12]. This is always implied by the *Clarke regularity* of φ at $\bar{x}$ in the sense of [3] and [23], which is equivalent to $\overline{\partial}\varphi(\bar{x}) = \widehat{\partial}\varphi(\bar{x})$ and agrees with the subdifferential regularity for locally Lipschitzian functions.

Now let us define the main objects of our study in this paper: *second-order subdifferentials* of extended-real-valued functions. We adopt the scheme of [11], where a second-order subdifferential was defined as the coderivative (2.2) of the basic first-order subdifferential mapping $\partial\varphi(\cdot)$. Along with this construction, here we consider another second-order subdifferential that is defined via the nonconvex coderivative (2.2) of the convexified subdifferential $\overline{\partial}\varphi(\cdot)$. The precise definitions follow.

DEFINITION 2.1. *Let $\varphi : \mathbb{R}^n \to \overline{\mathbb{R}}$ and let $\bar{x}$ be a point where φ is finite.*

(i) *Given $\bar{y} \in \partial\varphi(\bar{x})$, we define the* basic second-order subdifferential $\partial^2\varphi(\bar{x},\bar{y}) : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ *of φ at $\bar{x}$ relative to $\bar{y}$ by*

$$\partial^2\varphi(\bar{x},\bar{y})(y^*) := (D^*\partial\varphi)(\bar{x},\bar{y})(y^*) = \{x^* |\ (x^*,-y^*) \in N((\bar{x},\bar{y}); gph\,\partial\varphi).\} \tag{2.8}$$

(ii) *Given $\bar{y} \in \overline{\partial}\varphi(\bar{x})$, we define the* semiconvex second-order subdifferential $\overline{\partial}^2\varphi(\bar{x},\bar{y}) : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ *of φ at $\bar{x}$ relative to $\bar{y}$ by*

$$\overline{\partial}^2\varphi(\bar{x},\bar{y})(y^*) := (D^*\overline{\partial}\varphi)(\bar{x},\bar{y})(y^*) = \{x^* |\ (x^*,-y^*) \in N((\bar{x},\bar{y}); gph\,\overline{\partial}\varphi)\}. \tag{2.9}$$

If $\varphi \in C^1$ near $\bar{x}$ and $\nabla\varphi$ is strictly differentiable at $\bar{x}$ (in particular, $\varphi \in C^2$), then $\bar{y} = \nabla\varphi(\bar{x})$ and both second-order subdifferentials (2.8) and (2.9) reduce to the singleton

$$\partial^2\varphi(\bar{x})(y^*) = \overline{\partial}^2\varphi(\bar{x})(y^*) = \{(\nabla^2\varphi(\bar{x}))^*y^*\},$$

where $\nabla^2\varphi(\bar{x})$ stands for the classical Hessian matrix. Note that the sets (2.8) and (2.9) coincide when φ is Clarke regular *around* $\bar{x}$, in particular, when φ is either locally smooth or convex. In general these sets may be different and neither one is included in the other, in contrast to the first-order subdifferentials $\partial\varphi(\bar{x})$ and $\overline{\partial}\varphi(\bar{x})$.

In what follows we use also the corresponding first-order and second-order *superdifferential* sets for φ defined by

$$\partial^+\varphi(\bar{x}) := -\partial(-\varphi)(\bar{x}) \text{ and } \partial^{+2}\varphi(\bar{x},\bar{y})(y^*) := (D^*\partial^+\varphi)(\bar{x},\bar{y})(y^*), \tag{2.10}$$

where $\bar{y} \in \partial^+\varphi(\bar{x})$ and $y^* \in \mathbb{R}^n$; cf. [12].

One of the goals of this paper is to derive calculus rules for both second-order subdifferentials (2.8) and (2.9). To establish such a calculus in the next section, we use the sum and chain rules for the coderivative (2.2) of multifunctions that are stated below for the reader's convenience. Note that the original statements of these results in [12] impose closed graph assumptions on input multifunctions, while the proofs require merely a *local* closedness of the graphs. So we formulate the coderivative results under local closedness assumptions needed in what follows. First we present the sum rules corresponding to Theorem 4.1 and Corollary 4.4 in [12].

THEOREM 2.2. *Let F_1 and F_2 be given multifunctions from $\mathbb{R}^n$ into $\mathbb{R}^m$, and let $\bar{y} \in F_1(\bar{x}) + F_2(\bar{x})$. Assume that the graphs of F_1 and F_2 are closed whenever x is near $\bar{x}$, that the sets*

$$S(x,y) := \{(y_1,y_2) \in \mathbb{R}^m \times \mathbb{R}^m \,|\, y_1 \in F_1(x),\, y_2 \in F_2(x),\, y_1 + y_2 = y\}$$

are uniformly bounded around $(\bar{x},\bar{y})$*, and that the qualification condition*

$$D^*F_1(\bar{x},y_1)(0)\cap(-D^*F_2(\bar{x},y_2)(0))=\{0\}\quad\forall\,(y_1,y_2)\in S(\bar{x},\bar{y}) \tag{2.11}$$

is fulfilled. Then for all $y^*\in\mathbb{R}^m$ *one has*

$$\begin{aligned}D^*(F_1+F_2)(\bar{x},\bar{y})(y^*)&\\ \subset\bigcup_{(y_1,y_2)\in S(\bar{x},\bar{y})}&\left[D^*F_1(\bar{x},y_1)(y^*)+D^*F_2(\bar{x},y_2)(y^*)\right].\end{aligned} \tag{2.12}$$

Furthermore, if $F_1=f_1:\mathbb{R}^n\to\mathbb{R}^m$ *happens to be single-valued and strictly differentiable at* $\bar{x}$ *while* $F_2:\mathbb{R}^n\rightrightarrows\mathbb{R}^m$ *has the closed graph around* $(\bar{x},\bar{y}-f_1(\bar{x}))$*, then*

$$\begin{aligned}D^*(f_1+F_2)(\bar{x},\bar{y})(y^*)&\\ =(\nabla f_1(\bar{x}))^*\,y^*+D^*F_2(\bar{x},\bar{y}-f_1(\bar{x}))(y^*)&\ \ \forall y^*\in\mathbb{R}^m.\end{aligned} \tag{2.13}$$

Next we present two special cases of the general coderivative chain rule proved in [12, Theorem 5.1].

THEOREM 2.3. (i) *Let* $(f\circ G):\mathbb{R}^n\rightrightarrows\mathbb{R}^q$ *be the composition* $f(G(x))$ *of a single-valued mapping* $f:\mathbb{R}^m\to\mathbb{R}^q$ *and a set-valued mapping* $G:\mathbb{R}^n\rightrightarrows\mathbb{R}^m$ *and let* $(\bar{x},\bar{z})\in gph\,(f\circ G)$*. Assume that the multifunction* $M:\mathbb{R}^n\times\mathbb{R}^q\rightrightarrows\mathbb{R}^m$ *defined by*

$$M(x,z):=G(x)\cap f^{-1}(z) \tag{2.14}$$

is single-valued at $(\bar{x},\bar{z})$ *with* $M(\bar{x},\bar{z})=\{\bar{y}\}$ *and upper semicontinuous at this point. Assume also that* f *is strictly differentiable at* $\bar{y}$ *and that the graph of* G *is closed around* $(\bar{x},\bar{y})$*. Then*

$$D^*(f\circ G)(\bar{x},\bar{z})(z^*)\subset D^*G(\bar{x},\bar{y})((\nabla f(\bar{y}))^*\,z^*)\ \ \forall z^*\in\mathbb{R}^q. \tag{2.15}$$

(ii) *Let* $(F\circ g):\mathbb{R}^n\rightrightarrows\mathbb{R}^q$ *be the composition* $F(g(x))$ *of a set-valued mapping* $F:\mathbb{R}^m\rightrightarrows\mathbb{R}^q$ *and a single-valued mapping* $g:\mathbb{R}^n\to\mathbb{R}^m$ *strictly differentiable at* $\bar{x}$*. Take* $\bar{z}\in(F\circ g)(\bar{x})$ *and assume that the graph of* F *is closed around* $(g(\bar{x}),\bar{z})$ *and that*

$$D^*F(g(\bar{x}),\bar{z})(0)\cap ker(\nabla g(\bar{x}))^*=\{0\}. \tag{2.16}$$

Then one has

$$D^*(F\circ g)(\bar{x},\bar{z})(z^*)\subset(\nabla g(\bar{x}))^*D^*(F(g(\bar{x}),\bar{z})(z^*)\ \ \forall z^*\in\mathbb{R}^q. \tag{2.17}$$

3. Calculus rules for second-order subdifferentials. In this section we derive some calculus rules for both second-order subdifferentials defined in the previous section. Our goal is to express the second-order subdifferentials of sums and compositions for certain classes of functions in terms of the corresponding constructions involving their components. To furnish this, we are going to employ, based on definitions (2.8) and (2.9) of the second-order subdifferentials, calculus results for the coderivative (2.2) and the first-order subdifferentials (2.3) and (2.6). In this way we have to restrict ourselves to classes of functions for which the first-order subdifferential sum and chain rules hold as *equalities*, since the coderivative (2.2) does not possess any monotonicity properties. We begin with *sum rules* for the basic second-order subdifferential (2.8).

THEOREM 3.1. *Let $\varphi_i : \mathbb{R}^n \to \overline{\mathbb{R}}$, $i = 1, 2$, be finite at $\bar{x}$ and let $\bar{y} \in \partial(\varphi_1 + \varphi_2)(\bar{x})$. The following assertions hold.*

(i) *Assume that there is a neighborhood U of $\bar{x}$ such that*

$$\partial^\infty \varphi_1(x) \cap (-\partial^\infty \varphi_2(x)) = \{0\} \ \ \forall x \in U; \tag{3.1}$$

both functions φ_1 and φ_2 are l.s.c. on U and subdifferentially regular at every $x \in U$. Assume also that the graphs of $\partial\varphi_1$ and $\partial\varphi_2$ are closed whenever x is near $\bar{x}$ and that the sets

$$S(x, y) := \{(y_1, y_2) \in \mathbb{R}^n \times \mathbb{R}^n \,|\, y_1 \in \partial\varphi_1(x),\ y_2 \in \ \partial\varphi_2(x),\ y_1 + y_2 = y\} \tag{3.2}$$

are uniformly bounded around $(\bar{x}, \bar{y})$. Finally we impose the basic second-order qualification condition

$$\partial^2 \varphi_1(\bar{x}, y_1)\,(0) \cap \big(-\partial^2 \varphi_2(\bar{x}, y_2)\,(0)\big) = \{0\} \ \ \forall (y_1, y_2) \in S(\bar{x}, \bar{y}). \tag{3.3}$$

Then for all $y^* \in \mathbb{R}^n$ one has

$$\partial^2(\varphi_1 + \varphi_2)(\bar{x}, \bar{y})(y^*) \subset \bigcup_{(y_1, y_2) \in S(\bar{x}, \bar{y})} \Big[\partial^2 \varphi_1(\bar{x}, y_1)(y^*) + \partial^2 \varphi_2(\bar{x}, y_2)(y^*)\Big]. \tag{3.4}$$

(ii) *Assume that $\varphi_1 \in C^1$ around $\bar{x}$ while φ_2 is an arbitrary extended-real-valued function such that the graph of $\partial\varphi_2$ is closed around $(\bar{x}, \bar{y}_2)$, where $\bar{y}_2 := \bar{y} - \bar{y}_1$ and $\bar{y}_1 := \nabla\varphi_1(\bar{x})$. Assume also that the second-order qualification condition* (3.3) *holds at $(y_1, y_2) = (\bar{y}_1, \bar{y}_2)$. Then*

$$\partial^2(\varphi_1 + \varphi_2)(\bar{x}, \bar{y})(y^*) \subset \partial^2 \varphi_1(\bar{x}, \bar{y}_1)(y^*) + \partial^2 \varphi_2(\bar{x}, \bar{y}_2)(y^*) \ \ \forall y^* \in \mathbb{R}^n. \tag{3.5}$$

Moreover, if $\varphi_1 \in C^{1,1}$ (i.e., $\nabla\varphi_1$ is Lipschitz continuous around $\bar{x}$), then (3.3) *holds automatically and $\partial^2 \varphi_1(\bar{x}, \bar{y})(y^*) = \partial\langle y^*, \nabla\varphi_1\rangle(\bar{x})$ in* (3.5).

(iii) *Assume in addition to* (ii) *that $\nabla\varphi_1$ is strictly differentiable at $\bar{x}$ with $\nabla^2\varphi_1(\bar{x})$ denoting this strict derivative (in particular, $\varphi_1 \in C^2$ around $\bar{x}$). Then* (3.3) *holds and*

$$\partial^2(\varphi_1 + \varphi_2)(\bar{x}, \bar{y})(y^*) = (\nabla^2\varphi_1(\bar{x}))^* y^* + \partial^2 \varphi_2(\bar{x}, \bar{y}_2)(y^*) \ \ \forall y^* \in \mathbb{R}^n.$$

Proof. To justify (i), we first observe that under the first-order qualification condition (3.1) and the subdifferential regularity assumption on both φ_1 and φ_2 one has the equality

$$\partial(\varphi_1 + \varphi_2)(x) = \partial\varphi_1(x) + \partial\varphi_2(x) \ \ \forall x \in U; \tag{3.6}$$

see [12, Corollary 4.6]. Now assertion (i) follows directly from the inclusion sum rule (2.12) in Theorem 2.2 with $F_i = \partial\varphi_i$, $i = 1, 2$, and the definition of the basic second-order subdifferential.

To establish assertion (ii) of the theorem, we observe that if φ_1 is continuously differentiable around $\bar{x}$, then

$$\partial(\varphi_1 + \varphi_2)(x) = \nabla\varphi_1(x) + \partial\varphi_2(x) \ \ \forall x \in U \tag{3.7}$$

without *any* other assumptions; see [10, Corollary 4.1.2]. Again applying Theorem 2.2 with $F_1 = \nabla\varphi_1$ and $F_2 = \partial\varphi_2$, we arrive at (3.5). The mentioned refinement of (3.5)

for $\varphi_1 \in C^{1,1}$ follows from the scalarization formula (2.5). Note that the assumptions in (ii) do *not* require the subdifferential regularity of φ_2.

The proof of (iii) is similar to (ii). The only difference is that, instead of (2.12), we apply to (3.7) the equality sum rule (2.13) in Theorem 2.2. □

Note that the first-order qualification condition (3.1) automatically holds and the sets (3.2) are uniformly bounded around $(\bar{x},\bar{y})$ if one the functions φ_i is locally Lipschitzian around $\bar{x}$; cf. [12, Corollary 4.8].

Next let us derive sum rules for the semiconvex second-order subdifferential (2.9) similarly to Theorem 3.1. Observe that the way of proving Theorem 3.1(i) does not lead to new results in the case of (2.9) since then it would require the Clarke regularity of φ_i around $\bar{x}$, which implies that the second-order subdifferentials (2.9) and (2.8) coincide for φ_1, φ_2, and $\varphi_1+\varphi_2$. However, when $\varphi_1 \in C^1$ and φ_2 is general, we can obtain sum rules for $\overline{\partial}^2$ that are parallel to assertions (ii) and (iii) of Theorem 3.1 but are not implied by the latter.

THEOREM 3.2. *Let $\varphi_1 : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable around $\bar{x}$ and let $\varphi_2 : \mathbb{R}^n \to \overline{\mathbb{R}}$ be finite at $\bar{x}$ and l.s.c. around this point. Given $\bar{y} \in \overline{\partial}(\varphi_1+\varphi_2)(\bar{x})$ and $(\bar{y}_1,\bar{y}_2) := (\nabla\varphi_1(\bar{x}), \bar{y}-\nabla\varphi_1(\bar{x}))$, we assume that*

$$\overline{\partial}^2\varphi_1(\bar{x},\bar{y}_1)(0) \cap \Big(-\overline{\partial}^2\varphi_2(\bar{x},\bar{y}_2)(0)\Big) = \{0\} \tag{3.8}$$

and that the graph of $\overline{\partial}\varphi_2$ is closed around $(\bar{x},\bar{y}_2)$. Then

$$\overline{\partial}^2(\varphi_1+\varphi_2)(\bar{x},\bar{y})(y^*) \subset \overline{\partial}^2\varphi_1(\bar{x},\bar{y}_1)(y^*) + \overline{\partial}^2\varphi_2(\bar{x},\bar{y}_2)(y^*) \ \ \forall y^* \in \mathbb{R}^n, \tag{3.9}$$

where $\overline{\partial}^2\varphi_1(\bar{x},\bar{y})(y^*) = \partial\langle y^*, \nabla\varphi_1\rangle(\bar{x})$ with (3.8) holding automatically if $\varphi_1 \in C^{1,1}$ around $\bar{x}$. Moreover, if $\nabla\varphi_1$ has the strict derivative $\nabla^2\varphi_1(\bar{x})$ at $\bar{x}$, then

$$\overline{\partial}^2(\varphi_1+\varphi_2)(\bar{x},\bar{y})(y^*) = (\nabla^2\varphi_1(\bar{x}))^*y^* + \overline{\partial}^2\varphi_2(\bar{x},\bar{y}_2)(y^*) \ \ \forall y^* \in \mathbb{R}^n. \tag{3.10}$$

Proof. First let us show that

$$\overline{\partial}(\varphi_1+\varphi_2)(x) = \nabla\varphi_1(x) + \overline{\partial}\varphi_2(x) \ \ \forall x \in U, \tag{3.11}$$

where U is a neighborhood of $\bar{x}$ in which φ_1 is continuously differentiable and φ_2 is l.s.c. To furnish this, we use (3.7) and the equality

$$\partial^\infty(\varphi_1+\varphi_2)(x) = \partial^\infty\varphi_2(x) \ \ \forall x \in U$$

is valid in this setting; see [12, Corollary 4.6]. Employing these two properties and representation (2.6), we get

$$\begin{aligned}
\overline{\partial}(\varphi_1+\varphi_2)(x) &= clco\,[\partial(\varphi_1+\varphi_2)(x) + \partial^\infty(\varphi_1+\varphi_2)(x)]\\
&= clco\,[\nabla\varphi_1(x) + \partial\varphi_2(x) + \partial^\infty\varphi_2(x)]\\
&= \nabla\varphi_1(x) + clco\,[\partial\varphi_2(x) + \partial^\infty\varphi_2(x)]\\
&= \nabla\varphi_1(x) + \overline{\partial}\varphi_2(x) \ \ \forall x \in U,
\end{aligned}$$

which gives (3.11). Now using definition (2.9) and applying to (3.11) the coderivative sum rule (2.12) with the qualification condition (2.11), we obtain (3.9) under the second-order qualification condition (3.8). The refinement of this result for $\varphi_1 \in C^{1,1}$ follows from the scalarization formula (2.5). If $\nabla\varphi_1$ is strictly differentiable at $\bar{x}$, we

employ in (3.11) the equality sum rule (2.13) of Theorem 2.2 and arrive at (3.10). □

Remark 3.3. (i) Let us discuss the local closedness assumptions on the graph of the basic subdifferential imposed in Theorem 3.1 and used also in what follows. Given $\varphi : \mathbb{R}^n \to \overline{\mathbb{R}}$ finite at $\bar{x}$, the closedness of $\partial\varphi(x)$ for all x near $\bar{x}$ means that there is a neighborhood U of $\bar{x}$ such that the set $gph\,\partial\varphi$ is closed relative to $U \times \mathbb{R}^n$. Due to (2.4) this always happens when φ is *continuous* around $\bar{x}$. It also holds for every proper l.s.c. *convex* function and for a more general class of *amenable* functions that are especially important for the theory and applications of variational analysis; see [23], in particular, Definition 10.23 and Exercise 10.25(b) therein. The less restrictive requirement on the local closedness of $gph\,\partial\varphi$ around $(\bar{x}, \bar{y})$, imposed in Theorem 3.1(ii), means that $gph\,\partial\varphi$ is closed relative to a neighborhood of $(\bar{x}, \bar{y})$ and holds for *subdifferentially continuous* functions; see Definition 13.28 and the related discussion in [23]. The local closedness of $gph\,\partial\varphi$ may also be fulfilled in some other situations when the subdifferential continuity is violated as, e.g., in [23, Figure 13-3].

(ii) The local closed graph assumption on the convexified subdifferential $\overline{\partial}\varphi$ imposed in Theorem 3.2 is more restrictive and does not hold, in particular, for a locally continuous function unless it is assumed to be *directionally Lipschitzian*; see [22, Proposition 4R and the counterexample on p. 23]. However, it holds for every function amenable at $\bar{x}$ since such functions exhibit Clarke regularity at any point x in a neighborhood of $\bar{x}$, hence $\partial\varphi(x) = \overline{\partial}\varphi(x)$; see [23, Exercise 10.25(a) and (b)].

Next let us consider the composition

$$\varphi(x) = (\psi \circ h)(x) := \psi(h(x)) \tag{3.12}$$

of functions $h : \mathbb{R}^n \to \mathbb{R}^m$ and $\psi : \mathbb{R}^m \to \overline{\mathbb{R}}$. Our goal is to derive parallel *chain rules* for both second-order subdifferentials (2.8) and (2.9). First we examine the situation when the inner mapping h is smooth around the point in question while the outer function ψ is extended-real-valued.

THEOREM 3.4. *Given $\bar{x} \in \mathbb{R}^n$, we suppose that φ is finite around $\bar{x}$, that ψ is l.s.c. around $h(\bar{x})$, and that h is continuously differentiable around $\bar{x}$ and its Jacobian $\nabla h(\bar{x})$ has full row rank m. Suppose also that the mapping $\nabla h : \mathbb{R}^n \to \mathbb{R}^{m\times n}$ is strictly differentiable at $\bar{x}$. Then the following assertions hold.*

(i) *Let $\bar{y} \in \partial\varphi(\bar{x})$ for composition* (3.12) *and assume that the graph of $\partial\psi$ is closed around $(h(\bar{x}), \bar{v})$, where $\bar{v} \in \mathbb{R}^m$ is a unique vector satisfying the relations*

$$\bar{y} = (\nabla h(\bar{x}))^* \bar{v} \;\; \text{and} \;\; \bar{v} \in \partial\psi\,(h(\bar{x}))\,. \tag{3.13}$$

Then

$$\begin{aligned} &\partial^2\varphi(\bar{x}, \bar{y})(y^*) \\ &\quad \subset \big(\nabla^2\langle \bar{v}, h\rangle(\bar{x})\big)\, y^* + (\nabla h(\bar{x}))^* \partial^2\psi(h(\bar{x}), \bar{v})(\nabla h(\bar{x})\; y^*) \;\; \forall y^* \in \mathbb{R}^n. \end{aligned} \tag{3.14}$$

(ii) *Let $\bar{y} \in \overline{\partial}\varphi(\bar{x})$ for composition* (3.12) *and assume that the graph of $\overline{\partial}\psi$ is closed around $(h(\bar{x}), \bar{v})$, where $\bar{v}$ is uniquely determined by*

$$\bar{y} = (\nabla h(\bar{x}))^* \bar{v} \;\; \text{and} \;\; \bar{v} \in \overline{\partial}\psi\,(h(\bar{x}))\,.$$

Then

$$\begin{aligned} &\overline{\partial}^2\varphi(\bar{x}, \bar{y})(y^*) \\ &\quad \subset \big(\nabla^2\langle \bar{v}, h\rangle(\bar{x})\big)\, y^* + (\nabla h(\bar{x}))^* \overline{\partial}^2\psi(h(\bar{x}), \bar{v})(\nabla h(\bar{x})\; y^*) \;\; \forall y^* \in \mathbb{R}^n. \end{aligned} \tag{3.15}$$

Proof. First let us prove assertion (i). Observe that the second-order chain rule (3.14) can be equivalently rewritten as

$$\partial^2\varphi(\bar{x},\bar{y})\,(y^*) \subset \nabla_x((\nabla h(\bar{x}))^*\bar{v})\,y^* + \Big\{(\nabla h(\bar{x}))^*w \,|\, (w,-\nabla h(\bar{x})y^*) \in N((h(\bar{x}),\bar{v}); gph\,\partial\psi)\Big\}. \tag{3.16}$$

To establish (3.16), we start with the first-order equality chain rule

$$\partial\varphi(x) = (\nabla h(x))^*\,\partial\psi(h(x)) \tag{3.17}$$

that holds for all x from a neighborhood U of $\bar{x}$ under the assumptions made; see [23, Exercise 10.7]. This allows us to represent the multifunction $\partial\varphi$ as the composition

$$\partial\varphi(x) = (f\circ G)(x), \quad x\in U, \tag{3.18}$$

with the single-valued mapping $f:\mathbb{R}^n\times\mathbb{R}^m\to\mathbb{R}^n$ defined by

$$f(u,v) := (\nabla h(u))^*v \tag{3.19}$$

and the set-valued mapping $G:\mathbb{R}^n\rightrightarrows\mathbb{R}^n\times\mathbb{R}^m$ defined by

$$(u,v)\in G(x) \quad \text{iff} \quad u=x \text{ and } v\in\partial\psi(h(x)). \tag{3.20}$$

We are going to apply the coderivative chain rule of Theorem 2.3(i) to composition (3.18). To furnish this, let us define the multifunction $M:\mathbb{R}^n\times\mathbb{R}^n\rightrightarrows\mathbb{R}^m\times\mathbb{R}^n$ by

$$M(x,y) := \{(u,v)\in\mathbb{R}^n\times\mathbb{R}^m \,|\, u=x,\ v\in\partial\psi(h(x)),\ y=(\nabla h(u))^*v\},$$

which corresponds to (2.14) in Theorem 2.3. Since

$$M(x,y)\subset\{(u,v)\in\mathbb{R}^n\times\mathbb{R}^m \,|\, u=x,\ y=(\nabla h(u))^*v\}$$

and $M(\bar{x},\bar{y})\neq\emptyset$ due to $\bar{y}\in\partial\varphi(\bar{x})$, we get $M(\bar{x},\bar{y})=\{(\bar{x},\bar{v})\}$ with the vector $\bar{v}$ uniquely determined by (3.13). It is easy to see that all the assumptions of Theorem 2.3(i) are fulfilled for the above composition (3.18). Applying the coderivative chain rule (2.15) to composition (3.18) and taking into account the structure of (3.19), we obtain

$$\begin{aligned} D^*(f\circ G)(\bar{x},\bar{y})(y^*) &\subset D^*G(\bar{x},\bar{x},\bar{v})\,((\nabla f(\bar{x},\bar{v}))^*\,y^*) \\ &= D^*G(\bar{x},\bar{x},\bar{v})\left(\begin{bmatrix} \nabla_x((\nabla h(\bar{x}))^*\bar{v}) \\ (\nabla h(\bar{x}))^* \end{bmatrix} y^*\right). \end{aligned} \tag{3.21}$$

To compute the coderivative of G in (3.21), we observe that

$$D^*G(\bar{x},\bar{x},\bar{v})\,(u^*,v^*)\subset u^* + D^*(\partial\psi\circ h)\,(\bar{x},\bar{v})\,(v^*) \tag{3.22}$$

for all $(u^*,v^*)\in\mathbb{R}^n\times\mathbb{R}^m$. It follows from (3.20) and the inclusion

$$N((\bar{x},\bar{x},\bar{v}); gph\,G)\subset\Big\{(x^*,u^*,v^*)\in\mathbb{R}^n\times\mathbb{R}^n\times\mathbb{R}^m \,\Big|\, x^*=x_1^*+x_2^*,\ x_1^*=-u^*, \\ (x_2^*,v^*)\in N((\bar{x},\bar{v}); gph\,(\partial\psi\circ h))\Big\}$$

that holds due to

$$gph\, G = \Big\{(x,u,v) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \;\Big|\; u = x\Big\} \cap \Big\{(x,u,v) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \;\Big|\; v \in \partial\psi(h(x))\Big\}$$

and the normal cone representation for set intersections; see [12, Corollary 4.7].

It remains to compute the second term at the right-hand side of (3.22), which is the coderivative of a composition $F \circ g$ with $F := \partial\psi$ and $g := h$. To do it, we apply Theorem 2.3(ii) whose assumptions are fulfilled due the full rank condition $ker(\nabla h(\bar{x}))^* = \{0\}$ and the local closedness of $\partial\psi$. So we get

$$D^*(\partial\psi \circ h)\,(\bar{x},\bar{v})\,(v^*) \subset (\nabla h(\bar{x}))^*\, D^*\partial\psi(h(\bar{x}),\, \bar{v})\,(v^*) \;\; \forall v^* \in \mathbb{R}^m \tag{3.23}$$

from the coderivative chain rule (2.17). Now combining (3.21), (3.22), and (3.23), we arrive at the required inclusion (3.16) and finish the proof of assertion (i) in the theorem.

To prove (ii), we use the same procedure starting with the property

$$\overline{\partial}\varphi(x) = (\nabla h(x))^*\, \overline{\partial}\psi(h(x))$$

that follows, due to (2.6), from (3.17) and its counterpart for singular subgradients. □

Remark 3.5. If $n = m$ in Theorem 3.4, then the full rank condition means that the Jacobian matrix $\nabla h(\bar{x})$ is quadratic and nonsingular. According to the classical inverse mapping theorem, there is a single-valued local inverse h^{-1} that is strictly differentiable at the point $h(\bar{x})$. So applying Theorem 3.4 to $\psi = \varphi \circ h^{-1}$ in this case, one can get the opposite inclusions in (3.14) and (3.15), i.e., they hold as *equalities*. It was pointed out by Terry Rockafellar that the general case of Theorem 3.4 with the full rank condition could be reduced to the quadratic nonsingular case. It can be done similarly to the procedure in [23, Exercise 6.7]. Thus the second-order chain rules (3.14) and (3.15) in fact hold as equalities under the assumptions made.

Next let us consider compositions (3.12) involving nonsmooth inner mappings $h : \mathbb{R}^n \to \mathbb{R}^n$ while outer functions $\psi : \mathbb{R}^n \to \mathbb{R}$ are smooth. Suppose that each component h_i of h depends only on the ith component of the variable $x \in \mathbb{R}^n$. Given $\bar{x} \in \mathbb{R}^n$, we assume that ψ is continuously differentiable around $h(\bar{x})$ and introduce the index sets

$$I_+(\bar{x}) := \{i \in \{1,2,\ldots,n\} \,|\, (\nabla\psi(h(\bar{x})))_i \ge 0\} \;\; \text{and} \;\; I_-(\bar{x}) := \{1,2,\ldots,n\} \setminus I_+(\bar{x}).$$

THEOREM 3.6. *In addition to the assumptions above, we suppose that all the functions h_i, $i = 1,2,\ldots,n$, are Lipschitz continuous around $\bar{x}_i$ and that $\nabla\psi$ is strictly differentiable at $h(\bar{x})$ with the strict derivative denoted by $\nabla^2\psi(h(\bar{x}))$. Assume also that*

$$(\nabla\psi(h(\bar{x})))_i \ne 0 \;\; \textit{for all} \;\; i = 1,2,\ldots,n. \tag{3.24}$$

Then the following assertions hold.

(i) *Let $\bar{y} \in \partial\varphi(\bar{x})$. Then*

$$\partial^2\varphi(\bar{x},\bar{y})\,(y^*) \subset D^*h(\bar{x})\left(\nabla^2\psi(h(\bar{x}))\,(\bar{v} \bullet y^*)\right) + \Lambda(y^*) \tag{3.25}$$

for all $y^* \in \mathbb{R}^n$, where

$$\Lambda(y^*) := \left\{\lambda \in \mathbb{R}^n \;\Big|\; \lambda_i \in \left\langle \begin{array}{ll} \partial^2 h_i(\bar{x}_i,\bar{v}_i)\,((\nabla\psi(h(\bar{x})))_i\, y_i^*) & \textit{if} \quad i \in I_+(\bar{x}) \\ \partial^{+2} h_i(\bar{x}_i,\bar{v}_i)\,((\nabla\psi(h(\bar{x})))_i\, y_i^*) & \textit{if} \quad i \in I_-(\bar{x}) \end{array} \right\rangle \right\}$$

and $\bar{v} \in \mathbb{R}^n$ *is a unique vector satisfying the relations*

$$\bar{y} = (\nabla\psi(h(\bar{x}))) \bullet \bar{v}, \quad \bar{v}_i \in \begin{cases} \partial h_i(\bar{x}_i) & \text{if} \quad i \in I_+(\bar{x}), \\ \partial^+ h_i(\bar{x}_i) & \text{if} \quad i \in I_-(\bar{x}), \end{cases} \qquad i = 1, 2, \ldots, n.$$

(ii) *Let* $\bar{y} \in \overline{\partial}\varphi(\bar{x})$. *Then*

$$\overline{\partial}^2\varphi(\bar{x}, \bar{y})\,(y^*) \subset D^*h(\bar{x})\,(\nabla^2\psi(h(\bar{x}))\,(\bar{v} \bullet y^*)) + \overline{\Lambda}(y^*) \tag{3.26}$$

for all $y^* \in \mathbb{R}^n$, *where*

$$\overline{\Lambda}(y^*) := \left\{\lambda \in \mathbb{R}^n \mid \lambda_i = \overline{\partial}^2 h_i(\bar{x}_i, \bar{v}_i)\,((\nabla\psi(h(\bar{x})))_i\, y_i^*)\right\}$$

and $\bar{v} \in \mathbb{R}^n$ *is a unique vector satisfying*

$$\bar{y} = (\nabla\psi(h(\bar{x}))) \bullet \bar{v}, \quad \bar{v}_i \in \overline{\partial} h_i(\bar{x}_i), \quad i = 1, 2, \ldots, n.$$

Proof. First let us prove (i). We begin with the following first-order chain rule proved in [10, Theorem 4.7] (see also [12, Corollary 5.8]):

$$\partial\varphi(x) = \sum_{i=1}^{n} \partial(\alpha_i h_i)\,(\bar{x}) \quad \forall\ x \text{ close to } \bar{x}, \tag{3.27}$$

where $\alpha_i := (\nabla\psi(h(\bar{x})))_i,\ i = 1, 2 \ldots, n$. Due to (3.27) and the definitions of ∂ and ∂^+ in section 2 we get

$$\partial\varphi(\bar{x}) = \left\{ y \in \mathbb{R}^n \,\middle|\, y_i \in \begin{cases} (\nabla\psi(h(\bar{x})))_i\, \partial h_i(\bar{x}_i) & \text{if } i \in I_+(\bar{x}) \\ (\nabla\psi(h(\bar{x})))_i\, \partial^+ h_i(\bar{x}_i) & \text{if } i \in I_-(\bar{x}) \end{cases} \right\}. \tag{3.28}$$

Given $\bar{y} \in \partial\varphi(\bar{x})$, we now proceed similarly to the proof of Theorem 3.4 and represent $\partial\varphi$ in the composition form (3.18) with the single-valued mapping $f : \mathbb{R}^{2n} \to \mathbb{R}^n$ defined by

$$f(u, v) := \nabla\psi(u) \bullet v \tag{3.29}$$

and the set-valued mapping $G : \mathbb{R}^n \rightrightarrows \mathbb{R}^{2n}$ defined by

$$(u, v) \in G(x) \text{ iff } u = h(x) \quad \text{and} \quad v_i \in \begin{cases} \partial h_i(\bar{x}_i) & \text{if} \quad i \in I_+(\bar{x}), \\ \partial^+ h_i(\bar{x}_i) & \text{if} \quad i \in I_-(\bar{x}). \end{cases} \tag{3.30}$$

For this composition, the corresponding mapping (2.14) is given by

$$\begin{aligned} M(x, y) &= \left\{ (u, v) \in \mathbb{R}^n \times \mathbb{R}^n \mid y = \nabla\psi(u) \bullet v,\ u = h(x), v_i \in \begin{cases} \partial h_i(x_i) & \text{if } i \in I_+(x) \\ \partial^+ h_i(x_i) & \text{if } i \in I_-(x) \end{cases} \right\} \\ &= \left\{ (h(x), v) \in \mathbb{R}^n \times \mathbb{R}^n \mid y = \nabla\psi(h(x)) \bullet v,\ v_i \in \begin{cases} \partial h_i(x_i) & \text{if } i \in I_+(x) \\ \partial^+ h_i(x_i) & \text{if } i \in I_-(x) \end{cases} \right\}. \end{aligned}$$

Due to condition (3.24) the set $M(\bar{x}, \bar{y})$ reduces to the singleton $\{h(\bar{x}, \bar{v})\}$, where $\bar{v}$ is defined above. It is easy to see that the other assumptions of Theorem 2.3(i) hold as

well for composition (3.18) under consideration. Using this result and the structure of (3.29), we obtain the inclusion

$$D^*(f\circ G)\,(\bar{x},\bar{y})\,(y^*)\subset D^*G(\bar{x},h(\bar{x}),\,\bar{v})\;((\nabla f(h(\bar{x}),\bar{v})))^*\,y^*)\,, \tag{3.31}$$

where the Jacobian of f is computed by

$$(\nabla f(h(\bar{x}),\bar{v}))^*\;y^*=\left[\begin{array}{c}\nabla^2\psi(h(\bar{x}))\,\mathrm{Diag}(\bar{v})\\ \mathrm{Diag}(\nabla\psi(h(\bar{x})))\end{array}\right]y^*.$$

It remains to compute the coderivative of the mapping G in (3.31). Using the structure of (3.30) and the definition of ∂^{+2} in (2.10), we get

$$\begin{aligned}D^*G(\bar{x},h(\bar{x}),\bar{v})\,(u^*,v^*)\subset{}& D^*h(\bar{x})(u^*)\\ &+\left\{\lambda\in\mathbb{R}^n\;\middle|\;\lambda_i\in\left\langle\begin{array}{lll}\partial^2h_i(\bar{x}_i,\bar{v}_i)\,(v_i^*) & \text{if} & i\in I_+(\bar{x})\\ \partial^{+2}h_i(\bar{x}_i,\bar{v}_i)\,(v_i^*) & \text{if} & i\in I_-(\bar{x})\end{array}\right.\right\}\end{aligned}$$

for all $(u^*,v^*)\in\mathbb{R}^n\times\mathbb{R}^n$. Finally we arrive at (3.25) and finish the proof of assertion (i).

To prove assertion (ii), we proceed similarly to (i) starting with the equality

$$\overline{\partial}\varphi(x)=\sum_{i=1}^n\overline{\partial}(\alpha_ih_i)\,(\bar{x})\quad\forall\;x\text{ close to }\bar{x},$$

which immediately follows from (3.27) due to (2.6) and the Lipschitz continuity of φ around $\bar{x}$. Using the symmetry property (2.7) of the convexified subdifferential, we have

$$\overline{\partial}\varphi(\bar{x})=\big\{y\in\mathbb{R}^n|\;y_i\in(\nabla\psi(h(\bar{x})))_i\overline{\partial}h_i(\bar{x}_i),\;\;i=1,2,\ldots,n\big\}$$

instead of (3.28), which leads to the difference between the sets $\Lambda(y^*)$ and $\overline{\Lambda}(y^*)$ in the second-order chain rules (3.25) and (3.26). □

In some applications (see, e.g., section 5) one needs to compute the coderivative of multifunctions given by

$$Q(x,y):=\{q\in\mathbb{R}^m\,|\,q=h(x)\bullet v,\;v\in\partial\varphi(y)\}\quad\text{and} \tag{3.32}$$

$$\overline{Q}(x,y):=\big\{q\in\mathbb{R}^m\,|\,q=h(x)\bullet v,\;v\in\overline{\partial}\varphi(y)\big\}\,,$$

where $h:\mathbb{R}^n\to\mathbb{R}^m$ and $\varphi:\mathbb{R}^m\to\overline{\mathbb{R}}$. In these cases we do not compute the second-order subdifferentials of a function while the resulting formulas contain the corresponding second-order subdifferentials of φ, and thus they can be viewed as a part of the second-order calculus.

THEOREM 3.7. *Given $(\bar{x},\bar{y})\in\mathbb{R}^n\times\mathbb{R}^m$, we assume that φ is l.s.c. around $\bar{y}$ and that h is strictly differentiable at $\bar{x}$ with $h_i(\bar{x})\neq0$ for all $i=1,2,\ldots,m$. Then the following hold.*

(i) *Let $\bar{q}\in Q(\bar{x},\bar{y})$ and let $\bar{v}$ be a (unique) vector satisfying the relations*

$$\bar{q}=h(\bar{x})\bullet\bar{v},\quad\bar{v}\in\partial\varphi(\bar{y}). \tag{3.33}$$

Assume that the graph of $\partial\varphi$ is closed around $(\bar{y},\bar{v})$. Then

$$D^*Q(\bar{x},\bar{y},\bar{q})\,(q^*)\subset\left[\begin{array}{c}(\nabla h(\bar{x}))^*(\bar{v}\bullet q^*)\\ \partial^2\varphi(\bar{y},\bar{v})\,(h(\bar{x})\bullet q^*)\end{array}\right]\ \ \forall\ \ q^*\in\mathbb{R}^m. \tag{3.34}$$

(ii) *Let $\bar{q}\in\overline{Q}(\bar{x},\bar{y})$ and let $\bar{v}$ be a (unique) vector satisfying the relations*

$$\bar{q}=h(\bar{x})\bullet\bar{v},\quad \bar{v}\in\overline{\partial}\varphi(\bar{y}).$$

Assume that the graph of $\overline{\partial}\varphi$ is closed around $(\bar{y},\bar{v})$. Then

$$D^*\overline{Q}(\bar{x},\bar{y},\bar{q})\,(q^*)\subset\left[\begin{array}{c}(\nabla h(\bar{x}))^*(\bar{v}\bullet q^*)\\ \overline{\partial}^2\varphi(\bar{y},\bar{v})\,(h(\bar{x})\bullet q^*)\end{array}\right]\ \ \forall\ \ q^*\in\mathbb{R}^m.$$

Proof. The proof of this theorem is similar to the case of Theorem 3.6, so we present only the main points in proving assertion (i).

Clearly, multifunction (3.32) is represented as the composition

$$Q(x,y)=(f\circ G)(x,y), \tag{3.35}$$

where $f:\mathbb{R}^n\times\mathbb{R}^m\to\mathbb{R}^m$ is a smooth function defined by

$$f(u,v):=h(u)\bullet v$$

and where G maps $\mathbb{R}^n\times\mathbb{R}^m$ into subsets of $\mathbb{R}^n\times\mathbb{R}^m$ so that

$$(u,v)\in G(x,y)\text{ iff }u=x\text{ and }v\in\partial\varphi(y).$$

By the assumptions made on h, the vector $\bar{v}$ is indeed uniquely determined in (3.33). Applying Theorem 2.3(i) to composition (3.35) and taking into account the structure of the mappings involved, we get

$$D^*(f\circ G)\,(\bar{x},\bar{y},\bar{q})\,(q^*)\subset D^*G(\bar{x},\bar{y},\bar{x},\bar{v})\left(\left[\begin{array}{c}(\mathrm{Diag}(\bar{v})\,\nabla h(\bar{x}))^*\\ \mathrm{Diag}(h(\bar{x}))\end{array}\right]q^*\right)\ \text{ and}$$

$$D^*G(\bar{x},\bar{y},\bar{x},\bar{v})\,(u^*,v^*)=\left[\begin{array}{c}u^*\\ \partial^2\varphi(\bar{y},\bar{v})\,(v^*)\end{array}\right],$$

which implies (3.34). □

4. Computation of second-order subdifferentials. The value of the second-order subdifferential theory depends on the possibility to compute efficiently the second-order subdifferentials (2.8) and (2.9) for attractive classes of nonsmooth functions important for applications. In [4], it was done for the class of indicator functions of polyhedral convex sets that naturally appear in many important applications of variational analysis and optimization; see, in particular, [4], [20], and [23]. Note that all the functions of this class are fully amenable [23] and do not distinguish between constructions (2.8) and (2.9).

In this section we efficiently compute the second-order subdifferentials (2.8) and (2.9) for a new class of functions that are especially important for the study of mathematical programs with equilibrium constraints (cf., in particular, [18]) and frequently arise, e.g., in the modeling of some mechanical equilibria; see section 5. Functions of

this class do not generally exhibit subdifferential regularity and have different second-order subdifferentials (2.8) and (2.9), both of which are computed in what follows. Using the calculus results of section 3, we can compute the second-order subdifferentials for more general classes of functions via various compositions.

The attention is paid first to an extended-real-valued function $\varphi : \mathbb{R} \to \overline{\mathbb{R}}$ having the form

$$\varphi(\cdot) = \vartheta(\cdot) + \delta(\cdot;\Gamma). \tag{4.1}$$

In (4.1), $\delta(\cdot;\Gamma)$ is the indicator function of the closed interval $\Gamma := [\alpha,\beta] \subset \overline{\mathbb{R}}$ with α possibly equal to $-\infty$ and β possibly equal to $+\infty$; the function ϑ is *piecewise* C^2 in the following sense:

(i) ϑ is continuous on an open set $\mathcal{O}$ containing Γ;

(ii) there exist points $\kappa^1, \kappa^2, \ldots, \kappa^k$ in Γ with

$$\alpha < \kappa^1 < \kappa^2 < \cdots < \kappa^k < \beta$$

and twice continuously differentiable functions $\vartheta^j : \mathcal{O} \to \mathbb{R},\ j = 0, 1, \ldots, k$, such that

$$\vartheta(\xi) = \begin{cases} \vartheta^0(\xi) & \text{for} \quad \xi \in [\alpha, \kappa^1], \\ \vartheta^j(\xi) & \text{for} \quad \xi \in [\kappa^j, \kappa^{j+1}], \quad j = 1, 2, \ldots, k-1, \\ \vartheta^k(\xi) & \text{for} \quad \xi \in [\kappa^k, \beta]. \end{cases}$$

Example 4.1. Consider the functions

$$\varphi(p) = |p| + \mu(\max\{0,p\})^2 + \nu(\max\{0,-p\})^2 + \delta(p;\Gamma),$$
$$\tilde{\varphi}(p) = -|p| + \mu(\max\{0,p\})^2 + \nu(\max\{0,-p\})^2 + \delta(p;\Gamma),$$

where μ, ν are given parameters and $\Gamma = [-1,1]$. Both these functions can be easily converted to form (4.1). In particular, for the function φ we have $k = 1,\ \alpha = -1,\ \kappa^1 = 0,\ \beta = 1$, and the corresponding function ϑ attains the form

$$\vartheta(p) = \begin{cases} -p + \nu p^2 & \text{for} \quad p \in [-1, 0], \\ p + \mu p^2 & \text{for} \quad p \in [0, 1]. \end{cases}$$

The graphs of φ and $\partial\varphi$ are depicted on Figure 4.1(a) ($gph\,\partial\varphi = gph\,\overline{\partial}\varphi$) while the graphs of $\tilde{\varphi}$, $\partial\tilde{\varphi}$, and $\overline{\partial}\tilde{\varphi}$ are depicted on Figure 4.1(b) (for $\mu = 0.5,\ \nu = -0.5$).

It is easy to see that for φ given by (4.1) the normal cones to $gph\,\partial\varphi$ and $gph\,\overline{\partial}\varphi$ at any point can be computed directly from definition (2.1). Consequently, the respective second-order subdifferentials can be expressed in terms of the function data. It doesn't seem to be possible to have one universal formula that describes all the situations occurring at kink points of the subdifferential graphs. The general case of functions (4.1) is covered by formulas (4.4)–(4.10). For the reader's convenience, we mention in (4.4)–(4.10) some characteristic points at Figure 4.1, which illustrate the application of these general formulas to the case of simple functions considered in Example 4.1.

In the computation of the basic second-order subdifferential of functions φ in form (4.1) we use the following auxiliary statement, where

$$M := \left\{ j \in \{1, 2, \ldots, k\} \,\middle|\, \nabla\vartheta^{j-1}(\kappa^j) \le \nabla\vartheta^j(\kappa^j) \right\}.$$

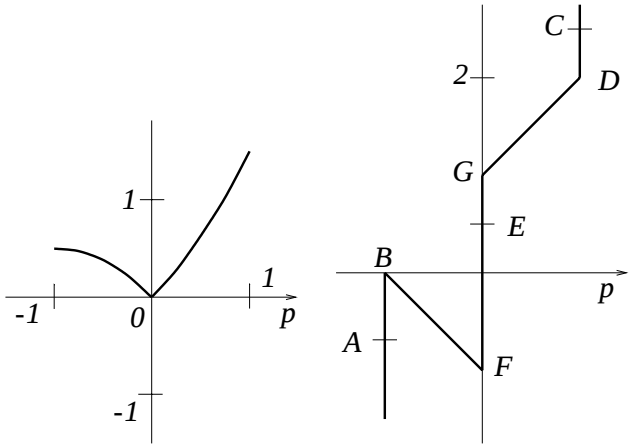


Fig. 4.1(a). $gph\,\phi$ and $gph\,\partial\phi$.

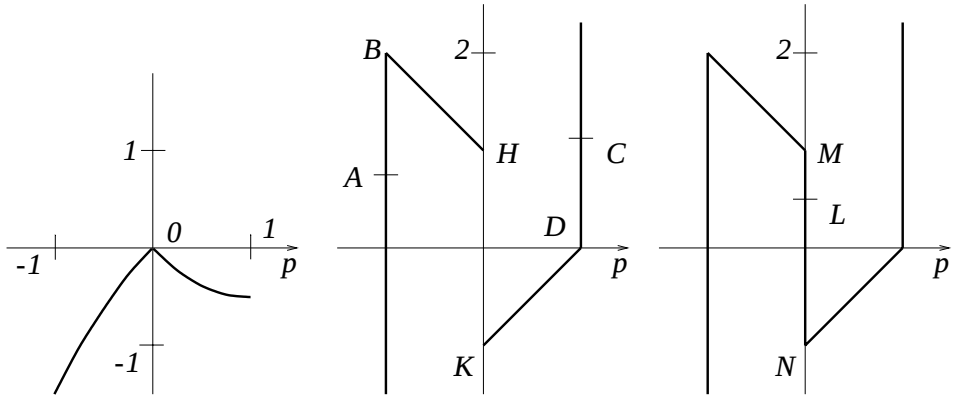


Fig. 4.1(b). $gph\,\tilde{\phi}$, $gph\,\partial\tilde{\phi}$, and $gph\,\overline{\partial}\tilde{\phi}$.

PROPOSITION 4.2. *Assume that $\alpha < \beta$ and both these numbers are finite. Denote*

$$A := \left\{(\xi,\eta) \in \mathbb{R}^2 \,|\, \xi = \alpha,\ \eta \in (-\infty,\ \nabla\vartheta^0(\alpha)]\right\},$$
$$B := \left\{(\xi,\eta) \in \mathbb{R}^2 \,|\, \xi = \beta,\ \eta \in [\nabla\vartheta^k(\beta),\ +\infty)\right\}.$$

Then one has

$$\begin{aligned} gph\,\partial\varphi \;&=\; \left\{(\xi,\eta) \in \mathbb{R}^2 \,|\, \alpha \le \xi \le \kappa^1,\ \eta = \nabla\vartheta^0(\xi)\right\} \\ &\cup \bigcup_{j=1}^{k-1} \left\{(\xi,\eta) \in \mathbb{R}^2 \,|\, \kappa^j \le \xi \le \kappa^{j+1},\ \eta = \nabla\vartheta^j(\xi)\right\} \\ &\cup \left\{(\xi,\eta) \in \mathbb{R}^2 \,|\, \kappa^k \le \xi \le \beta,\ \eta = \nabla\vartheta^k(\xi)\right\} \cup A \cup B \\ &\cup \bigcup_{j\in M} \left\{(\xi,\eta) \in \mathbb{R}^2 \,|\, \xi = \kappa^j,\ \eta \in \left[\nabla\vartheta^{j-1}(\kappa^j),\ \nabla\vartheta^j(\kappa^j)\right]\right\}. \end{aligned} \tag{4.2}$$

If $\alpha = -\infty$ or $\beta = +\infty$, then (4.2) *holds true with $A = \emptyset$ or $B = \emptyset$, respectively.*

Proof. Since ϑ is continuously differentiable around α and β, we get

$$\partial\varphi(\alpha) = \nabla\vartheta^0(\alpha) + \mathbb{R}_- \quad \text{and} \quad \partial\varphi(\beta) = \nabla\vartheta^k(\beta) + \mathbb{R}_+$$

due to (3.7) with $\partial\varphi(\xi)=\partial\vartheta^j(\xi)$ for $\xi\in\operatorname{int}\Gamma$ and the appropriate j from above. Thus φ is continuously differentiable on the intervals (α,κ^1), (κ^j,κ^{j+1}), $j=1,2,\ldots,k-1$, and (κ^k,β). Moreover, its gradient equals to the gradient of the appropriate function ϑ^j. It remains to analyze the points κ^j, $j=1,2,\ldots,k$. One can easily observe from the definitions of φ and the basic subdifferential that

$$\partial\varphi(\kappa^j)=\begin{cases}\big[\nabla\vartheta^{j-1}(\kappa^j),\,\nabla\vartheta^j(\kappa^j)\big] & \text{if } j\in M,\\ \big\{\nabla\vartheta^{j-1}(\kappa^j),\,\nabla\vartheta^j(\kappa^j)\big\} & \text{otherwise.}\end{cases}\tag{4.3}$$

This completes the proof. □

Using the structure of each set of $gph\,\partial\varphi\subset\mathbb{R}^2$ in (4.2), we are able to compute the normal cone (2.1) to these sets at any pair $(\bar p,\bar v)\in gph\,\partial\varphi$. To facilitate the notation, let us put

$$\mathcal{A}^j(\xi):=\left\{(w,z)\in\mathbb{R}^2\,|\,z=-\frac{1}{\nabla^2\vartheta^j(\xi)}\,w \text{ if } \nabla^2\vartheta^j(\xi)\neq 0 \text{ and } w=0 \text{ otherwise}\right\}$$

for all $\xi\in\mathcal{O}$, $j=0,1,\ldots,k$. Based on the construction of the normal cone (2.1) and Proposition 4.2, we get

$$N((\bar p,\bar v);gph\,\partial\varphi)=\begin{cases}\mathcal{A}^0(\bar p) & \text{if} \quad \bar p\in(\alpha,\kappa^1),\\ \mathcal{A}^j(\bar p) & \text{if} \quad \bar p\in(\kappa^j,\kappa^{j+1}),\\ \mathcal{A}^k(\bar p) & \text{if} \quad \bar p\in(\kappa^k,\beta).\end{cases}\qquad j=1,2,\ldots,k-1.\tag{4.4}$$

For $\bar p=\alpha$ and $\bar p=\beta$ one has, respectively,

$$N((\alpha,\bar v);gph\,\partial\varphi)=\begin{cases}\{(w,z)\in\mathbb{R}^2\,|\,z=0\} \text{ if } \bar v<\nabla\vartheta^0(\alpha) \text{ (cf. points } A),\\ \{(w,z)\in\mathbb{R}^2\,|\,z=0\}\cup\mathcal{A}^0(\alpha)\\ \cup\{(w,z)\in\mathbb{R}^2\,|\,w+\nabla^2\vartheta^0(\alpha)\,z\le 0,\ z\ge 0\}\\ \qquad\qquad\text{if } \bar v=\nabla\vartheta^0(\alpha) \text{ (cf. points } B);\end{cases}\tag{4.5}$$

$$N((\beta,\bar v);gph\,\partial\varphi)=\begin{cases}\{(w,z)\in\mathbb{R}^2\,|\,z=0\} \text{ if } \bar v>\nabla\vartheta^k(\beta) \text{ (cf. points } C),\\ \{(w,z)\in\mathbb{R}^2\,|\,z=0\}\cup\mathcal{A}^k(\beta)\\ \cup\{(w,z)\in\mathbb{R}^2\,|\,w+\nabla^2\vartheta^k(\beta)\,z\ge 0,\ z\le 0\}\\ \qquad\qquad\text{if } \bar v=\nabla\vartheta^k(\beta) \text{ (cf. points } D).\end{cases}\tag{4.6}$$

Finally taking $\bar p=\kappa^j$ with $j\in\{1,2,\ldots,k\}$, we have to distinguish between the following two situations.

(a) Let $j\in M$. Then

$$N((\kappa^j,\bar v);gph\,\partial\varphi)=\begin{cases}\{(w,z)\in\mathbb{R}^2\,|\,z=0\}\text{ if }\nabla\vartheta^{j-1}(\kappa^j)<\bar v<\nabla\vartheta^j(\kappa^j)\\ \qquad\qquad\text{(cf. point }E\text{)},\\ \{(w,z)\in\mathbb{R}^2\,|\,z=0\}\cup\mathcal{A}^{j-1}(\kappa^j)\\ \cup\{(w,z)\in\mathbb{R}^2\,|\,w+\nabla^2\vartheta^{j-1}(\kappa^j)\,z\ge 0,\ z\le 0\}\\ \qquad\qquad\text{if }\bar v=\nabla\vartheta^{j-1}(\kappa^j)\text{ (cf. point }F\text{)},\\ \{(w,z)\in\mathbb{R}^2\,|\,z=0\}\cup\mathcal{A}^{j}(\kappa^j)\\ \cup\{(w,z)\in\mathbb{R}^2\,|\,w+\nabla^2\vartheta^{j}(\kappa^j)\,z\le 0,\ z\ge 0\}\\ \qquad\qquad\text{if }\bar v=\nabla\vartheta^{j}(\kappa^j)\text{ (cf. point }G\text{)},\end{cases}\tag{4.7}$$

provided that $\nabla\vartheta^{j-1}(\kappa^j)<\nabla\vartheta^j(\kappa^j)$, and

$$\begin{aligned}N((\kappa^j,\bar v);gph\,\partial\varphi)&=\mathcal{A}^{j-1}(\kappa^j)\cup\mathcal{A}^j(\kappa^j)\\ &\cup\left\{(w,z)\in\mathbb{R}^2|\,-\nabla^2\vartheta^{j-1}(\kappa^j)\,z\le w\le-\nabla^2\vartheta^j(\kappa^j)\,z\right\},\end{aligned}\tag{4.8}$$

provided that $\bar v=\nabla\vartheta^{j-1}(\kappa^j)=\nabla\vartheta^j(\kappa^j)$.

(b) Let $j\notin M$. Then $\bar v$ cannot lie between $\nabla\vartheta^{j-1}(\kappa^j)$ and $\nabla\vartheta^j(\kappa^j)$, and hence one has

$$\begin{aligned}&N((\kappa^j,\bar v);gph\,\partial\varphi)\\ &=\begin{cases}\{(w,z)\in\mathbb{R}^2\,|\,w+\nabla^2\vartheta^{j-1}(\kappa^j)\,z\ge 0\}\text{ if }\bar v=\nabla\vartheta^{j-1}(\kappa^j)\text{ (cf. point }H\text{)},\\ \{(w,z)\in\mathbb{R}^2\,|\,w+\nabla^2\vartheta^{j}(\kappa^j)\,z\le 0\}\text{ if }\bar v=\nabla\vartheta^{j}(\kappa^j)\text{ (cf. point }K\text{)}.\end{cases}\end{aligned}\tag{4.9}$$

Next let us observe from (2.6) and Proposition 4.2 that $N((\bar p,\bar v);gph\,\overline{\partial}\varphi)=N((\bar p,\bar v);gph\,\partial\varphi)$ whenever formulas (4.4)–(4.8) apply. The only difference between these cones occurs in the case of $\bar p=\kappa^j$ and $j\in\{1,2,\ldots,k\}\setminus M$. In this case

$$N((\kappa^j,\bar v);gph\,\overline{\partial}\varphi)=\begin{cases}\{(w,z)\in\mathbb{R}^2\,|\,z=0\}\text{ if }\nabla\vartheta^{j-1}(\kappa^j)>\bar v>\nabla\vartheta^j(\kappa^j)\\ \qquad\qquad\text{(cf. point }L\text{)},\\ \{(w,z)\in\mathbb{R}^2\,|\,z=0\}\cup\mathcal{A}^{j-1}(\kappa^j)\\ \cup\{(w,z)\in\mathbb{R}^2\,|\,w+\nabla^2\vartheta^{j-1}(\kappa^j)\,z\ge 0,\ z\ge 0\}\\ \qquad\qquad\text{if }\bar v=\nabla\vartheta^{j-1}(\kappa^j)\text{ (cf. point }M\text{)},\\ \{(w,z)\in\mathbb{R}^2\,|\,z=0\}\cup\mathcal{A}^{j}(\kappa^j)\\ \cup\{(w,z)\in\mathbb{R}^2\,|\,w+\nabla^2\vartheta^{j}(\kappa^j)\,z\le 0,\ z\le 0\}\\ \qquad\qquad\text{if }\bar v=\nabla\vartheta^{j}(\kappa^j)\text{ (cf. point }N\text{)}.\end{cases}\tag{4.10}$$

Taking into account the above calculations, we establish the main result of this section that concerns *separable* extended-real-valued functions of many variables $\psi:\mathbb{R}^\ell\to\overline{\mathbb{R}}$ given by

$$\psi(p)=\sum_{i=1}^{\ell}\varphi_i(p_i),\tag{4.11}$$

where each $\varphi_i:\mathbb{R}\to\overline{\mathbb{R}}$ has structure (4.1) and satisfies all the assumptions posed at the beginning of this section. Correspondingly, we associate with each function φ_i

the respective interval $\Gamma_i = [\alpha_i, \beta_i]$, the piecewise C^2 function ϑ_i, the junction points κ_i^j, $j = 1, 2, \ldots, k_i$, and the index set M_i defined above. In the next statement we provide the exact formulas for computing both second-order subdifferentials (2.8) and (2.9) for functions ψ of class (4.11) in terms of their initial data.

THEOREM 4.3. *Let $\psi : \mathbb{R}^\ell \to \overline{\mathbb{R}}$ be a function of type* (4.11)*, where all the summands φ_i have structure* (4.1) *and satisfy the respective assumptions. Then the following assertions hold.*

(i) *Given $(\bar{p}, \bar{v}) \in \mathbb{R}^\ell \times \mathbb{R}^\ell$ with $\bar{v} \in \partial\psi(\bar{p})$, one has*

$$\partial^2\psi(\bar{p}, \bar{v})(z) = \big\{w \in \mathbb{R}^\ell \,|\, (w_i, -z_i) \in N((\bar{p}_i, \bar{v}_i); gph\, \partial\varphi_i),\ i = 1, 2, \ldots, \ell\big\} \tag{4.12}$$

for any $z \in \mathbb{R}^\ell$, where the cones $N((\bar{p}_i, \bar{v}_i); gph\, \partial\varphi_i)$, $i = 1, 2, \ldots, \ell$, are computed in (4.4)–(4.9).

(ii) *Given $(\bar{p}, \bar{v}) \in \mathbb{R}^\ell \times \mathbb{R}^\ell$ with $\bar{v} \in \overline{\partial}\psi(\bar{p})$, one has*

$$\overline{\partial}^2\psi(\bar{p}, \bar{v})(z) = \big\{w \in \mathbb{R}^\ell \,|\, (w_i, -z_i) \in N((\bar{p}_i, \bar{v}_i); gph\, \overline{\partial}\varphi_i),\ \ i = 1, 2, \ldots, \ell\big\} \tag{4.13}$$

for any $z \in \mathbb{R}^\ell$, where $N((\bar{p}_i, \bar{v}_i); gph\, \overline{\partial}\varphi_i) = N((\bar{p}_i, \bar{v}_i); gph\, \partial\varphi_i)$ when $(\bar{p}_i, \bar{v}_i) \in gph\, \overline{\partial}\varphi_i$ and formulas (4.4)–(4.8) *apply, and where $N((\kappa_i^j, \bar{v}_i); gph\, \overline{\partial}\varphi_i)$ are computed by formula* (4.10) *when $j \in \{1, 2, \ldots, k_i\} \setminus M_i$.*

Proof. First let us justify (i). Using the separable structure of l.s.c. functions (4.11), we get, due to [23, Proposition 10.5], that

$$\partial\psi(\bar{p}) = \sum_{i=1}^{\ell} \partial\varphi_i(\bar{p}_i), \tag{4.14}$$

which implies that

$$\bar{v} = (\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_\ell) \quad \text{with} \quad \bar{v}_i \in \partial\varphi_i(\bar{p}_i).$$

Employing the projection rule for the normal cones in (4.14), we arrive at

$$N((\bar{p}, \bar{v}); gph\, \partial\psi) = \sum_{i=1}^{\ell} N((\bar{p}_i, \bar{v}_i); gph\, \partial\varphi_i).$$

Now the second-order subdifferential formula (4.12) follows directly from definition (2.8).

Next let us justify assertion (ii). Taking into account the discussion before Theorem 4.3, it remains to prove that

$$\overline{\partial}\psi(\bar{p}) = \sum_{i=1}^{\ell} \overline{\partial}\varphi_i(\bar{p}_i) \tag{4.15}$$

for the functions ψ and φ_i in (4.11) and (4.1). To see it, we observe that (4.15) can be violated only if for some $i \in \{1, 2, \ldots, \ell\}$ one has $\overline{\partial}\varphi_i(\bar{p}_i) \neq \partial\varphi_i(\bar{p}_i)$, i.e., if $\bar{p}_i = \kappa_i^j$ for some $j \in \{1, 2, \ldots, k_i\} \setminus M_i$. Let L denote the collection of indices $i \in \{1, 2, \ldots, \ell\}$ for which this happens. It follows from the definitions that

$$\overline{\partial}\psi(\bar{p}) \subset \sum_{i \notin L} \partial\varphi_i(\bar{p}_i) \times \sum_{i \in L} [\nabla\vartheta_i^{j-1}(\kappa_i^j),\, \nabla\vartheta_i^j(\kappa_i^j)]. \tag{4.16}$$

On the other hand, we have

$$\overline{\partial}\psi(\bar{p}) \supset clco\, \partial\psi(\bar{p}) = \sum_{i \notin L} \partial\varphi_i(\bar{p}_i) \times \sum_{i \in L} [\nabla\vartheta_i^{j-1}(\kappa_i^j),\, \nabla\vartheta_i^j(\kappa_i^j)]$$

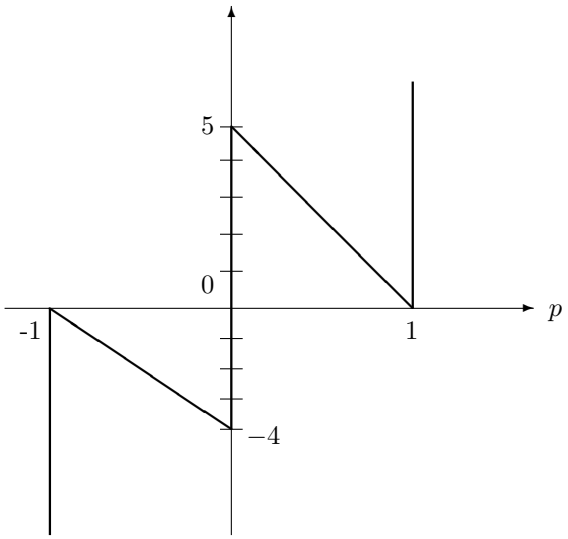


FIG. 4.2. $gph\,\partial\psi$.

due to (2.6) and (4.3). Thus we establish the equality in (4.16) and justify (4.15) in the general case under consideration. This finishes the proof of the theorem. □

Using the calculus results of the previous section and the formulas of Theorem 4.3, we can substantially extend the class of functions for which the second-order subdifferentials (2.8) and (2.9) can be efficiently computed. Let us consider several examples that illustrate the application of the chain rules in Theorems 3.4, 3.6, and 3.7 combined with the calculations presented above. Note that for functions ψ given by (4.11) and (4.1), the graphs of $\partial\psi$ and $\overline{\partial}\psi$ are closed (see Proposition 4.2 and the proof of Theorem 4.3); thus for such functions the results of section 3 can be readily applied. For brevity we present calculations only for the basic second-order subdifferential (2.8).

Example 4.4. Let

$$\psi(p) := \begin{cases} 5p - 2.5p^2 & \text{for } p \in [0,1], \\ -4p - 2p^2 & \text{for } p \in [-1,0], \\ +\infty & \text{otherwise,} \end{cases} \tag{4.17}$$

which corresponds to (4.11), (4.1) with $\ell = 1$, $k_1 = 1$, $\alpha_1 = -1$, $\kappa_1^1 = 0$, and $\beta_1 = 1$. The graph of the subdifferential mapping $\partial\psi$ for (4.17) is shown in Figure 4.2. Consider the composition $\varphi(x) = (\psi \circ h)(x)$ of the function $\psi : \mathbb{R} \to \mathbb{R}$ in (4.17) and a smooth mapping $h : \mathbb{R}^2 \to \mathbb{R}$ given by

$$h(x_1, x_2) := (x_1)^2 + x_1 + 2x_2.$$

Using Theorem 3.4 for this composition with $\bar{x} = (0,0)$ and $\bar{y} = (-4,-8) \in \partial\varphi(\bar{x})$, we get $\bar{v} = -4$ from (3.13) and obtain the inclusion

$$\begin{aligned} \partial^2\varphi(\bar{x},\bar{y})\,(y^*) \subset &\begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}(-4)\begin{bmatrix} y_1^* \\ y_2^* \end{bmatrix} \\ &+ \left\{ \begin{bmatrix} w \\ 2w \end{bmatrix} \middle| \, (w, -y_1^* - 2y_2^*) \in N((0,-4); gph\,\partial\psi) \right\} \end{aligned}$$

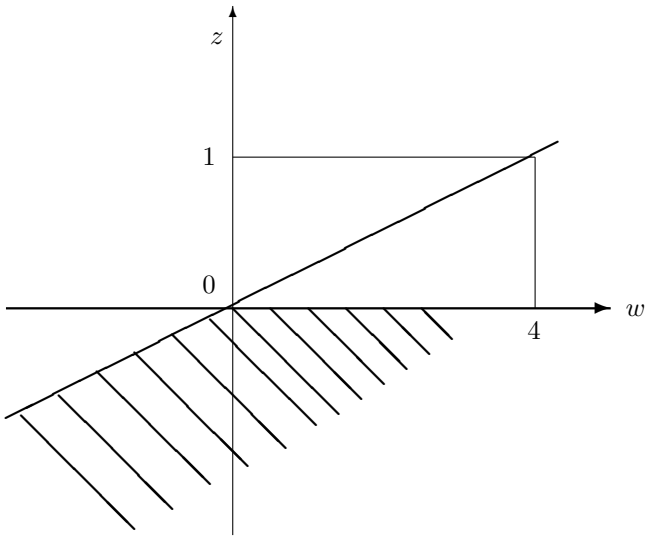


FIG. 4.3. $N((0,-4);gph\,\partial\psi)$.

for all $y^*=(y_1^*,y_2^*)\in\mathbb{R}^2$, where the normal cone $N((0,-4);gph\,\partial\psi)$ is computed by

$$N((0,-4);gph\,\partial\psi)=\big\{(w,z)\in\mathbb{R}^2\,|\,z=0\big\}\cup\left\{(w,z)\in\mathbb{R}^2\,|\,z=\frac{1}{4}w\right\}$$
$$\cup\big\{(w,z)\in\mathbb{R}^2\,|\,w-4z\ge 0,\ z\le 0\big\}$$

due to (4.7) and is depicted in Figure 4.3. Note that the above formula for $\partial^2\varphi$ actually holds as equality; see Remark 3.5.

Example 4.5. Consider the composition $\varphi=\psi\circ h$ in Theorem 3.6, where

$$\psi(p):=\frac{1}{2}(p_1)^2+p_1p_2+(p_2)^2+2p_1+p_2,\quad p=(p_1,p_2)\in\mathbb{R}^2,$$

and $h=(h_1,h_2):\mathbb{R}^2\to\mathbb{R}^2$ is given by

$$h_1(x_1):=-|x_1|,\quad h_2(x_2)=|x_2|.$$

Taking $\bar{x}=(0,0)$ and $\bar{y}=(-2,-1)$, we check the assumption (3.24) and compute

$$\nabla^2\psi(h(\bar{x}))=\begin{bmatrix}1&1\\1&2\end{bmatrix},\quad \bar{v}=\begin{bmatrix}-1\\-1\end{bmatrix},\quad \bar{v}\bullet y^*=\begin{bmatrix}-y_1^*\\-y_2^*\end{bmatrix}$$

in (3.25) for any $y^*=(y_1^*,y_2^*)\in\mathbb{R}^2$. Thus (3.25) gives the inclusion

$$\partial^2\varphi(\bar{x},\bar{y})\,(y^*)\subset D^*h(\bar{x})\begin{bmatrix}-y_1^*-y_2^*\\-y_1^*-2y_2^*\end{bmatrix}\tag{4.18}$$
$$+\Big\{\lambda\in\mathbb{R}^2\,\Big|\,\lambda_1\in\partial^2h_1(0,-1)\,(2y_1^*),\ \lambda_2\in\partial^2h_2(0,-1)\,(y_2^*)\Big\}.$$

Applying Theorem 4.3(i) to the above functions h_1 and h_2, we get

$$\partial^2\,h_1(0,-1)\,(2y_1^*)=\mathbb{R}_-$$

$$\partial^2\,h_2(0,-1)\,(y_2^*)=\left\{w\in\mathbb{R}\,\middle|\,w\in\begin{cases}\{0\} & \text{if}\quad y_2^*<0\\ \mathbb{R} & \text{if}\quad y_2^*=0\\ \mathbb{R}_+ & \text{if}\quad y_2^*>0\end{cases}\right\}.$$

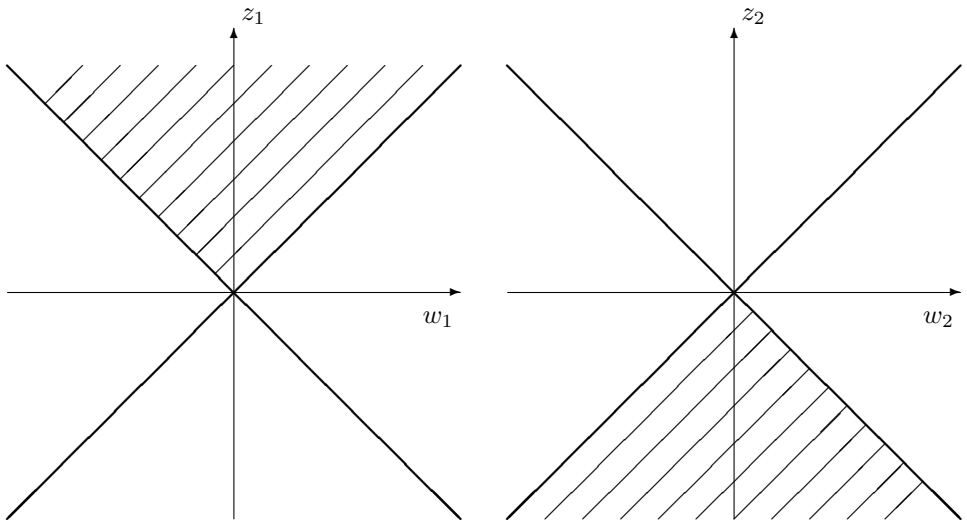


FIG. 4.4. $N((0,0);gph\,h)$.

So inclusion (4.18) reduces to

$$\partial^2\varphi(\bar{x},\bar{y})\,(y^*) \subset D^*h(\bar{x})\begin{bmatrix} -y_1^*-y_2^* \\ -y_1^*-2y_2^* \end{bmatrix} + \left\{ w\in\mathbb{R}^2 \,\middle|\, w_1\le 0, w_2 \in \begin{cases} \{0\} & \text{if} \quad y_2^*<0 \\ \mathbb{R} & \text{if} \quad y_2^*=0 \\ \mathbb{R}_+ & \text{if} \quad y_2^*>0 \end{cases} \right\}.$$

It remains to compute the coderivative of h at $\bar{x}=(0,0)$, which requires the computation of the normal cone (2.1) to the graph of h at $(0,0)$. Employing the definition, we obtain

$$N((0,0);gph\,h) = \Big(\Big\{(w_1,z_1)\in\mathbb{R}^2 \,|\, z_1=w_1\Big\}\cup\Big\{(w_1,z_1)\in\mathbb{R}^2\,|\,z_1=-w_1\Big\}\cup epi|\cdot|\Big) \times \Big(\Big\{(w_2,z_2)\in\mathbb{R}^2\,|\,z_2=w_2\Big\}\cup\Big\{(w_2,z_2)\in\mathbb{R}^2\,|\,z_2=-w_2\Big\}\cup hypo(-|\cdot|)\Big)$$

that is depicted in Figure 4.4. Combining these results, we arrive at an efficient upper approximation of $\partial^2\varphi(\bar{x},\bar{y})\,(y^*)$ for any $y^*\in\mathbb{R}^2$ on the basis of Theorem 3.6. In particular, for $y^*=(2,\,-0.5)$ we get

$$\partial^2\varphi(\bar{x},\bar{y})\,(y^*) \subset \begin{bmatrix} [-1.5,\,1.5] \\ \{-1,\,1\} \end{bmatrix} + \begin{bmatrix} \mathbb{R}_- \\ 0 \end{bmatrix} = \begin{bmatrix} (-\infty,\,1.5] \\ \{-1,\,1\} \end{bmatrix}.$$

Example 4.6. Consider a multifunction $Q:\mathbb{R}^3\times\mathbb{R}^2\rightrightarrows\mathbb{R}^2$ of the form (3.32) and compute its coderivative using Theorems 3.7 and 4.3. Let

$$h(x) := \begin{bmatrix} 2x_1+x_2 \\ x_2+x_3 \end{bmatrix}, \quad \varphi(y) := |y_1|+|y_2|$$

in (3.32) and take $\bar{x}=(1,1,1)$, $\bar{y}=(0,0)$, and $\bar{q}=(-3,2)$. All the assumptions of

Theorem 3.7(i) hold, and we have

$$D^*Q(\bar{x},\bar{y},\bar{q})\,(q^*) \subset \left[\begin{array}{l} -2q_1^* \\ -q_1^*+q_2^* \\ q_2^* \\ \partial^2\varphi(0,\bar{v})\left[\begin{array}{l} 3q_1^* \\ 2q_2^* \end{array}\right] \end{array}\right] \quad \text{with } \bar{v}=\left[\begin{array}{r} -1 \\ 1 \end{array}\right]$$

for each $q^* \in \mathbb{R}^2$. Using Theorem 4.3(i), we can easily compute the basic second-order subdifferential for the function φ under consideration:

$$\partial^2\varphi(0,\bar{v})\left[\begin{array}{l} 3q_1^* \\ 2q_2^* \end{array}\right] = \left\{ w\in\mathbb{R}^2 \,\middle|\, w_1 \in \left\{\begin{array}{lll} \{0\} & \text{if} & q_1^*<0, \\ \mathbb{R} & \text{if} & q_1^*=0, \\ \mathbb{R}_+ & \text{if} & q_1^*>0, \end{array}\right. \right. \left. w_2 \in \left\{\begin{array}{lll} \{0\} & \text{if} & q_2^*>0 \\ \mathbb{R} & \text{if} & q_2^*=0 \\ \mathbb{R}_- & \text{if} & q_2^*<0 \end{array}\right. \right\}.$$

Thus we get an efficient upper approximation for the coderivative $D^*Q(\bar{x},\bar{y},\bar{q})\,(q^*)$.

5. Applications. In the final section of the paper we present some applications of the second-order subdifferential theory to stability (sensitivity) analysis of parametric variational systems described by GEs in the form

$$0 \in f(x,y) + Q(x,y), \tag{5.1}$$

where $y \in \mathbb{R}^m$ is the so-called *decision* variable, $x \in \mathbb{R}^n$ is a perturbation vector (parameter), $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^m$ is a continuously differentiable vector function, and $Q : \mathbb{R}^n \times \mathbb{R}^m \rightrightarrows \mathbb{R}^m$ is a multifunction. Note that, in contrast to the classical framework of Robinson [21] and subsequent publications, the perturbation parameter x enters not only the single-valued term f but also the set-valued operator Q in (5.1). It has been well recognized that parametric generalized equations provide a convenient ground for the study of sensitivity and stability questions in many areas of nonlinear programming, complementarity, equilibrium theory, economic models, etc. In particular, (5.1) reduces to the standard form of variational inequalities when $Q(y) = N(y;\Omega)$ is the classical normal cone operator for a convex set Ω.

In what follows we consider more general structures of Q in (5.1) given in one of the forms

$$Q(x,y) = \left\langle \begin{array}{l} \partial\varphi(g(x,y)) \text{ if } g(x,y) \in dom\,\varphi, \\ \emptyset \text{ otherwise,} \end{array}\right. \tag{5.2}$$

$$Q(x,y) = \left\langle \begin{array}{l} \overline{\partial}\varphi(g(x,y)) \text{ if } g(x,y) \in dom\,\varphi, \\ \emptyset \text{ otherwise} \end{array}\right. \tag{5.3}$$

by using the basic and convexified subdifferentials of the outer function. We always assume that the extended-real-valued function $\varphi : \mathbb{R}^m \to \overline{\mathbb{R}}$ is proper and that the vector function $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^m$ is continuously differentiable at the points in question. Note that generalized equations (5.1) with structures (5.2) and (5.3) contain various types of variational and hemivariational inequalities being particularly useful in some mechanical applications; see the examples below.

Let us define the (multivalued) *solution map* $S : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$ to (5.1) by

$$S(x) := \{y \in \mathbb{R}^m \,|\, 0 \in f(x,y) + Q(x,y)\} \tag{5.4}$$

and extend the results of [13] on robust Lipschitzian stability of (5.4) to the class of generalized equations with structures (5.2) and (5.3). Given a reference point $(\bar{x}, \bar{y}) \in gph\, S$, we obtain efficient conditions ensuring the so-called *pseudo-Lipschitzian* property of S around $(\bar{x}, \bar{y})$ in the sense of Aubin [1], which means that there are neighborhoods $\mathcal{U}$ of $\bar{x}$ and $\mathcal{V}$ of $\bar{y}$ and a modulus $L \geq 0$ satisfying

$$S(x_1) \cap \mathcal{V} \subset S(x_2) + L\|x_1 - x_2\|\,\mathbb{B} \quad \forall\, x_1,\, x_2 \in \mathcal{U}. \tag{5.5}$$

Property (5.5) reduces to the classical local Lipschitz continuity if S is single-valued around $\bar{x}$; in general it is equivalent to the fundamental properties of metric regularity and openness at a linear rate for the inverse mapping S^{-1} (see [13] and [23] for more discussions and references).

THEOREM 5.1. (i) *Let* $(\bar{x}, \bar{y})$ *satisfy the GE* (5.1) *with* Q *given by* (5.2). *Assume that* $gph\, \partial\varphi$ *is closed around* $(g(\bar{x}, \bar{y}), -f(\bar{x}, \bar{y}))$ *and that the conditions*

(A) *the adjoint GE*

$$0 \in (\nabla_y f(\bar{x}, \bar{y}))^*\, u + (\nabla_y g(\bar{x}, \bar{y}))^*\, \partial^2\varphi\,(g(\bar{x}, \bar{y}),\, -f(\bar{x}, \bar{y}))\,(u)$$

possesses only the trivial solution $u = 0$*;*

(B)

$$ker\,(\nabla_y g(\bar{x}, \bar{y}))^* \cap \partial^2\varphi\,(g(\bar{x}, \bar{y}),\, -f(\bar{x}, \bar{y}))\,(0) = \{0\}$$

are fulfilled. Then the corresponding solution map S *is pseudo-Lipschitzian around* $(\bar{x}, \bar{y})$.

(ii) *Let* $(\bar{x}, \bar{y})$ *satisfy the GE* (5.1) *with* Q *given by* (5.3). *Assume that* $gph\, \overline{\partial}\varphi$ *is closed around* $(g(\bar{x}, \bar{y}), -f(\bar{x}, \bar{y}))$ *and that the conditions*

($\overline{\text{A}}$) *the adjoint GE*

$$0 \in (\nabla_y f(\bar{x}, \bar{y}))^*\, u + (\nabla_y g(\bar{x}, \bar{y}))^*\, \overline{\partial}^2\varphi\,(g(\bar{x}, \bar{y}),\, -f(\bar{x}, \bar{y}))\,(u)$$

possesses only the trivial solution $u = 0$*;*

($\overline{\text{B}}$)

$$ker\,(\nabla_y g(\bar{x}, \bar{y}))^* \cap \overline{\partial}^2\varphi\,(g(\bar{x}, \bar{y}),\, -f(\bar{x}, \bar{y}))\,(0) = \{0\}$$

are fulfilled. Then the corresponding solution map S *is pseudo-Lipschitzian around* $(\bar{x}, \bar{y})$.

Proof. It is sufficient to prove only assertion (i) since the proof of (ii) is similar. Let us observe that

$$S(x) = \{y \in \mathbb{R}^m \,|\, s(x,y) \in \Lambda\} \tag{5.6}$$

for S defined by (5.4) and (5.2), where

$$s(x,y) := \begin{bmatrix} g(x,y) \\ -f(x,y) \end{bmatrix} \quad \text{and} \quad \Lambda := gph\, \partial\varphi.$$

To ensure the pseudo-Lipschitzian property of the multifunction S, we are going to employ the coderivative criterion from [13, Theorem 3.2]. In order to furnish

this, we need to obtain an efficient upper approximation of the coderivative of S at the reference point $(\bar{x}, \bar{y})$. Let us do it by applying Theorem 6.10 from [12] to the multifunction S in form (5.6). One can easily check that our assumptions (A) and (B) guarantee the satisfaction of both qualification conditions in the latter theorem. So, using that result and the structure of $s(\cdot)$ and Λ in (5.6), we arrive at the coderivative estimate

$$D^*S(\bar{x}, \bar{y})\,(y^*) \subset \Big\{x^* \in \mathbb{R}^n \,|\, x^* \in (\nabla_x f(\bar{x}, \bar{y}))^* v + (\nabla_x g(\bar{x}, \bar{y}))^* \partial^2 \varphi(g(\bar{x}, \bar{y}),\, -f(\bar{x}, \bar{y}))\,(v),$$
$$-y^* \in (\nabla_y f(\bar{x}, \bar{y}))^* v + (\nabla_y g(\bar{x}, \bar{y}))^* \partial^2 \varphi(g(\bar{x}, \bar{y}),\, -f(\bar{x}, \bar{y}))\,(v)\Big\}.$$

Now we check that the assumptions (A) and (B) imply that $D^*S(\bar{x}, \bar{y})\,(0) = \{0\}$, which ensures the pseudo-Lipschitzian property of S around $(\bar{x}, \bar{y})$ due to [13, Theorem 3.2]. □

Remark 5.2. To verify both conditions (A) and (B) of Theorem 5.1, one has to be able to evaluate the second-order subdifferentials of φ. It has been done in [17] and [19] in connection with necessary optimality conditions for mathematical programs where a nonlinear or a mixed complementarity problem arises among constraints. This corresponds to $g(x, y) = y$ and φ given in form (4.11), (4.1) with $\vartheta_i(\cdot) \equiv 0$ and either $\Gamma_i = \mathbb{R}_+$ or Γ_i equal to a bounded closed interval, respectively.

Using the theory and computations presented above, we can enlarge a class of mathematical programs with equilibrium constraints where optimality and stability conditions can be efficiently derived. In particular, Theorem 5.1 directly leads to verifiable conditions ensuring robust Lipschitzian stability of solution maps to the following class of perturbed *implicit complementarity problems*: given $x \in \mathbb{R}^n$, find $y \in \mathbb{R}^m$ such that

$$f(x, y) \geq 0, \quad y \geq b(x, y), \quad \langle f(x, y),\, y - b(x, y)\rangle = 0, \tag{5.7}$$

where $b : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^m$ is continuously differentiable. One can easily see that (5.7) reduces to (5.1) and (5.2) with $g(x, y) = y - b(x, y)$ and the same φ as in the nonlinear complementarity problem mentioned above. Note that form (5.7) is useful in equilibrium models corresponding to filtration through porous media [15] as well as to contact problems with compliant obstacles.

Next let us consider generalized equations (5.1) with multifunctions Q given in one of the following forms:

$$Q(x, y) = \{q \in \mathbb{R}^m \,|\, q = h(x) \bullet v,\, v \in \partial\varphi(y)\}, \tag{5.8}$$

$$Q(x, y) = \big\{q \in \mathbb{R}^m \,|\, q = h(x) \bullet v,\, v \in \overline{\partial}\varphi(y)\big\}, \tag{5.9}$$

where $h : \mathbb{R}^n \to \mathbb{R}^m$ is continuously differentiable and $\varphi : \mathbb{R}^m \to \overline{\mathbb{R}}$ is l.s.c. around reference points. Using a procedure similar to the proof of Theorem 5.1 and applying the second-order calculus rule of Theorem 3.7, we arrive at the following stability conditions.

THEOREM 5.3. (i) *Let $(\bar{x}, \bar{y})$ satisfy the GE (5.1) with Q given by (5.8). Assume that $h_i(\bar{x}) \neq 0$ for all $i = 1, 2, \ldots, m$ and that $\bar{v}$ is a unique vector satisfying the relations*

$$f(\bar{x}, \bar{y}) + h(\bar{x}) \bullet \bar{v} = 0, \quad \bar{v} \in \partial\varphi(\bar{y}).$$

Assume also that $\operatorname{gph} \partial\varphi$ is closed around $(\bar{y}, \bar{v})$ and that

(C) *the adjoint GE*

$$0 \in (\nabla_y f(\bar{x}, \bar{y}))^* u + \partial^2 \varphi(\bar{y}, \bar{v})\,(h(\bar{x}) \bullet u)$$

possesses only the trivial solution $u = 0$.

Then the corresponding solution map S *is pseudo-Lipschitzian around* $(\bar{x}, \bar{y})$.

(ii) *Let* $(\bar{x}, \bar{y})$ *satisfy the GE* (5.1) *with* Q *given by* (5.9). *Assume that* $h^i(\bar{x}) \neq 0$ *for all* $i = 1, 2, \ldots, m$ *and* $\bar{v}$ *is a unique vector satisfying*

$$f(\bar{x}, \bar{y}) + h(\bar{x}) \bullet \bar{v} = 0, \quad \bar{v} \in \overline{\partial}\varphi(\bar{y}).$$

Assume also that $\mathrm{gph}\,\overline{\partial}\varphi$ *is closed around* $(\bar{y}, \bar{v})$ *and that*

($\overline{\mathrm{C}}$) *the adjoint GE*

$$0 \in (\nabla_y f(\bar{x}, \bar{y}))^* u + \overline{\partial}^2 \varphi(\bar{y}, \bar{v})\,(h(\bar{x}) \bullet u)$$

possesses only the trivial solution $u = 0$.

Then the corresponding solution map S *is pseudo-Lipschitzian around* $(\bar{x}, \bar{y})$.

Proof. Let us sketch the main points in proving (i), which works also for (ii) in the same way. Clearly, the solution map S to (5.1) is represented as

$$S(x) = \{y \in \mathbb{R}^m \mid (x, y, -f(x, y)) \in \mathrm{gph}\, Q\}. \tag{5.10}$$

To compute the coderivative of the above mapping, we use [12, Theorem 6.10]. Taking into account the structure of Q in (5.8) and employing Theorem 3.7(i), we compute an efficient upper approximation of the coderivative of Q and thus the corresponding upper approximation of the coderivative of S in (5.10) due to [12, Theorem 6.10]. Note that the condition (C) alone ensures the fulfillment of all the qualification conditions in the latter theorem by virtue of (3.34). Moreover, it implies that $D^*S(\bar{x}, \bar{y})(0) = \{0\}$ for the multifunction S in (5.10). Due to the coderivative criterion in [13, Theorem 3.2], we justify the pseudo-Lipschitzian property of the solution map to the GE (5.1) with Q given by (5.8). □

In the concluding part of this paper we present applications of the results obtained to some problems of continuum mechanics. For these problems, our results lead to efficient conditions ensuring robust solution stability with respect to perturbations that are expressed in terms of problem data.

First let us consider a discretized hemivariational inequality corresponding to a *contact problem with nonmonotone friction* taken from [5]. The underlying mechanical problem (see Figure 5.1) can be described as follows. There is an elastic body Ω supported from below by a rigid obstacle and exposed to external forces that represent our perturbation vector x. Vectors y_t, y_n represent, respectively, tangential and normal displacements of the discretization nodes lying on the contact boundary Γ_c. In many situations it is possible to replace the "nonpenetrability condition" $y_n \geq 0$ with the equality $y_n = 0$. Then we put $y := y_t \in \mathbb{R}^m$ and describe the equilibrium in this mechanical problem by the following generalized equation of type (5.1):

$$0 \in Ay + p(x) + \partial\psi(Dy), \tag{5.11}$$

where m is the number of nodes on Γ_c, n is the dimension of external forces $x \in \mathbb{R}^n$, A is an $m \times m$ positively definite "stiffness" matrix, $p : \mathbb{R}^n \to \mathbb{R}^m$ is a continuously differentiable vector function related to external forces, and D is an $m \times m$ nonsingular

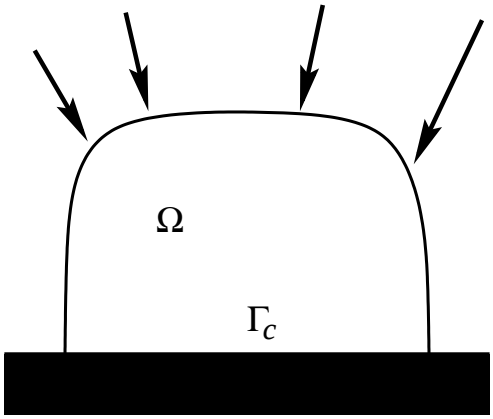


FIG. 5.1. *Contact problem with nonmonotone friction.*

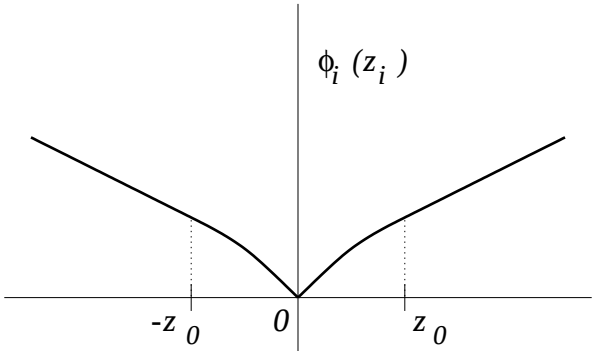


FIG. 5.2. *A nonmonotone friction law.*

matrix defined by a quadrature formula that is used for the boundary integral along Γ_c. The function ψ in (5.11) is given in the form

$$\psi(z) = \sum_{i=1}^{m} \varphi_i(z_i) \quad \text{with} \quad z \in \mathbb{R}^m, \tag{5.12}$$

where $\varphi_i : \mathbb{R} \to \mathbb{R}$ represents the *nonmonotone friction law* depicted in Figure 5.2.

In [5] we can find the following analytic formula for φ_i:

$$\varphi_i(z_i) = \begin{cases} (-k_1 + k_2 z_0)\, z_i + \frac{k_2}{2}(z_0)^2 & \text{if} \quad z_i < -z_0, \\ -k_1 z_i - \frac{k_2}{2}(z_i)^2 & \text{if} \quad z_i \in [-z_0, 0), \\ k_1 z_i - \frac{k_2}{2}(z_i)^2 & \text{if} \quad z_i \in [0, z_0), \\ (k_1 - k_2 z_0)\, z_i + \frac{k_2}{2}(z_0)^2 & \text{if} \quad z_i \geq z_0, \end{cases} \tag{5.13}$$

where $z_0 > 0$, $k_1 > 0$, and $k_2 > 0$ are given parameters. Since the function φ defined by (5.12) and (5.13) is obviously of form (4.11) and (4.1), its second-order subdifferentials can be computed by the formulas of Theorem 4.3. It is easy to observe that the subdifferentials (2.8) and (2.9) coincide for this function.

Let us explicitly express the stability conditions of Theorem 5.1 for the mechanical system under consideration using the above calculations. Note that condition (B) of Theorem 5.1 automatically holds for (5.11) due to the nonsingularity of the matrix D. To efficiently express condition (A), let us employ the following nine index sets assigned to the reference pair $(\bar{x}, \bar{y})$:

$$
\begin{aligned}
I_1(\bar{x},\bar{y}) &:= \{i \in \{1,2,\dots,m\} | (D\bar{y})_i < -z_0\},\\
I_2(\bar{x},\bar{y}) &:= \{i \in \{1,2,\dots,m\} | (D\bar{y})_i = -z_0\},\\
I_3(\bar{x},\bar{y}) &:= \{i \in \{1,2,\dots,m\} | (D\bar{y})_i \in (-z_0,0)\},\\
I_4(\bar{x},\bar{y}) &:= \{i \in \{1,2,\dots,m\} | (D\bar{y})_i = 0, (-A\bar{y}-p(\bar{x}))_i = -k_1\},\\
I_5(\bar{x},\bar{y}) &:= \{i \in \{1,2,\dots,m\} | (D\bar{y})_i = 0, (-A\bar{y}-p(\bar{x}))_i \in (-k_1,k_1)\},\\
I_6(\bar{x},\bar{y}) &:= \{i \in \{1,2,\dots,m\} | (D\bar{y})_i = 0, (-A\bar{y}-p(\bar{x}))_i = k_1\},\\
I_7(\bar{x},\bar{y}) &:= \{i \in \{1,2,\dots,m\} | (D\bar{y})_i \in (0,z_0)\},\\
I_8(\bar{x},\bar{y}) &:= \{i \in \{1,2,\dots,m\} | (D\bar{y})_i = z_0\},\\
I_9(\bar{x},\bar{y}) &:= \{i \in \{1,2,\dots,m\} | (D\bar{y})_i > z_0\}.
\end{aligned}
\tag{5.14}
$$

They completely describe the position of $(D\bar{y}, -A\bar{y}-p(\bar{x}))$ on $gph\,\partial\psi$ and, clearly, $\bigcup_{j=1}^{9} I_j(\bar{x},\bar{y}) = \{1,2,\dots,m\}$. With these index sets we associate by formulas (4.4)–(4.9) the following nine normal cones to $gph\,\partial\varphi_i$ computed at the points $((D\bar{y})_i, (-A\bar{y}-p(\bar{x}))_i)$. To simplify the notation, we give (as a subscript) only the number of the index set to which the respective component of $(D\bar{y}, -A\bar{y}-p(\bar{x}))$ belongs:

$$
\begin{aligned}
N_1 &= N_9 = \{0\} \times \mathbb{R},\\
N_2 &= N_1 \cup \left\{(w,u) \in \mathbb{R}^2 | u = \tfrac{1}{k_2}\, w\right\} \cup \left\{(w,u) \in \mathbb{R}^2 | 0 \le w \le k_2 u\right\},\\
N_3 &= N_7 = \left\{(w,u) \in \mathbb{R}^2 | u = \tfrac{1}{k_2}\, w\right\},\\
N_4 &= N_3 \cup \left\{(w,u) \in R^2 | u = 0\right\} \cup \left\{(w,u) \in R^2 | w - k_2 u \ge 0,\ u \le 0\right\},\\
N_5 &= \left\{(w,u) \in R^2 | u = 0\right\},\\
N_6 &= N_3 \cup N_5 \cup \left\{(w,u) \in R^2 | w - k_2 u \le 0,\ u \ge 0\right\},\\
N_8 &= N_1 \cup N_3 \cup \left\{(w,u) \in R^2 | k_2 u \le w \le 0\right\}.
\end{aligned}
\tag{5.15}
$$

On the basis of these calculations and Theorem 5.1, we arrive at verifiable conditions for robust Lipschitzian stability of the solution map to the nonmonotone friction problem described by (5.11).

PROPOSITION 5.4. *Let* $(\bar{x}, \bar{y})$ *satisfy the GE* (5.11) *with* ψ *given in* (5.12) *and* (5.13). *Consider the adjoint GE*

$$
0 \in A^* u + \Xi(\bar{x}, \bar{y}, u), \tag{5.16}
$$

where the set

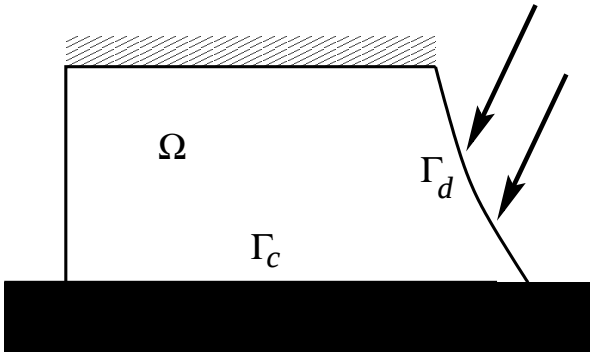


FIG. 5.3. *Shape design problem with nonmonotone friction.*

$$\Xi(\bar{x},\bar{y}) = \sum_{i=1}^{m} \Xi_i(\bar{x},\bar{y},u_i)$$

is generated by

$$\Xi_i(\bar{x},\bar{y},u_i) = \{w_i \in \mathbb{R} | (w_i,-u_i) \in N_j\}$$

with N_j computed in (5.15) *and where j is a uniquely determined index from* $\{1,2,\dots,m\}$ *for which $i \in I_j(\bar{x},\bar{y})$ in* (5.14). *Then the solution map to* (5.11) *is pseudo-Lipschitzian around $(\bar{x},\bar{y})$ if the adjoint GE* (5.16) *possesses only the trivial solution $u = 0$.*

Next let us consider the following *shape design problem with nonmonotone friction* that can be examined by using Theorem 5.3. The elastic body Ω in Figure 5.3 is now fixed at the upper part of the boundary. Furthermore, the outer forces act only on the right-hand side of the boundary Γ_d whose shape is described by the *design variable* $x \in \mathbb{R}^n$. As in the previous problem, $y = y_t$ is the vector of tangential displacements of m discretization nodes lying on the contact boundary Γ_c. Positions of these nodes depend now on the design variable. Imposing the nonpenetrability condition $y_n = 0$, we get the model described by the discretized hemivariational inequality (cf. [5])

$$0 \in A(x)\,y + b(x) + h(x) \bullet \partial\psi(Dy), \tag{5.17}$$

where A is an $m \times m$ matrix depending on the design variable, $b : \mathbb{R}^n \to \mathbb{R}^m$ is a continuously differentiable vector function representing the outer forces, and $h : \mathbb{R}^n \to \mathbb{R}^m$ is a continuously differentiable vector function that reflects the influence of the variable boundary Γ_d on the discretization nodes lying on Γ_c. The friction function ψ and the nonsingular matrix D are the same as in the previous problem (5.11). For simplicity we suppose that D is the unit $m \times m$ matrix.

Model (5.17) is represented in the GE form (5.1) with Q of the composite structure (5.8). So we apply Theorem 5.3 to derive efficient conditions ensuring robust Lipschitzian stability of the solution map at the reference point $(\bar{x},\bar{y})$. These conditions can be given in a verifiable form using the second-order subdifferential formulas

of Theorem 4.3. To furnish this, let us define the index sets

$$\begin{aligned}
J_1(\bar{x},\bar{y}) &:= \{i\in\{1,2,\dots,m\}|\bar{y}_i<-z_0\},\\
J_2(\bar{x},\bar{y}) &:= \{i\in\{1,2,\dots,m\}|\bar{y}_i=-z_0\},\\
J_3(\bar{x},\bar{y}) &:= \{i\in\{1,2,\dots,m\}|\bar{y}_i\in(-z_0,0)\},\\
J_4(\bar{x},\bar{y}) &:= \{i\in\{1,2,\dots,m\}|\bar{y}_i=0,\ \bar{v}^i=-k_1\},\\
J_5(\bar{x},\bar{y}) &:= \{i\in\{1,2,\dots,m\}|\bar{y}_i=0,\ \bar{v}^i\in(-k_1,k_1)\},\\
J_6(\bar{x},\bar{y}) &:= \{i\in\{1,2,\dots,m\}|\bar{y}_i=0,\ \bar{v}^i=k_1\},\\
J_7(\bar{x},\bar{y}) &:= \{i\in\{1,2,\dots,m\}|\bar{y}_i\in(0,z_0)\},\\
J_8(\bar{x},\bar{y}) &:= \{i\in\{1,2,\dots,m\}|\bar{y}_i=z_0\},\\
J_9(\bar{x},\bar{y}) &:= \{i\in\{1,2,\dots,m\}|\bar{y}_i>z_0\},
\end{aligned}\tag{5.18}$$

where $\bar{v}$ is a unique vector satisfying the relations

$$0\in A(\bar{x})\,\bar{y}+b(\bar{x})+h(\bar{x})\bullet\bar{v},\quad \bar{v}\in\partial\psi(\bar{y}).\tag{5.19}$$

The corresponding normal cones are computed in (5.15).

PROPOSITION 5.5. *Let $(\bar{x},\bar{y})$ satisfy the GE* (5.17) *with the unit matrix D and with ψ given by* (5.12) *and* (5.13). *Assume that $h_i(\bar{x})\neq 0$ for all $i=1,2,\dots,m$ and that $\bar{v}$ is a unique vector satisfying relations* (5.19). *Consider the adjoint GE*

$$0\in(A(\bar{x}))^*u+\Theta(\bar{x},\bar{y},u),\tag{5.20}$$

where the set

$$\Theta(\bar{x},\bar{y},u)=\sum_{i=1}^{m}\Theta_i(\bar{x},\bar{y},u_i)$$

is generated by

$$\Theta_i(\bar{x},\bar{y},u_i)=\{w_i\in\mathbb{R}|(w_i,-u_i)\in N_j\}$$

with N_j computed in (5.15) *and where j is a uniquely determined index from $\{1,2,\dots,m\}$ for which $i\in I_j(\bar{x},\bar{y})$ in* (5.18). *Then the solution map to* (5.17) *is pseudo-Lipschitzian around $(\bar{x},\bar{y})$ if the adjoint GE* (5.20) *possesses only the trivial solution $u=0$.*

Note that both mechanical models considered above rely on the classical concept of *given friction* [16]. They are mechanically justified provided that $y_n=0$ at all equilibrium pairs (x,y) for x from a neighborhood of $\bar{x}$. Otherwise the equality $y_n=0$ has to be replaced by the inequality $y_n\geq 0$, and the models become more complicated.

Finally let us present a simple two-dimensional example illustrating the usage of Proposition 5.4.

Example 5.6. Consider the GE (5.11) with $n=2$, $m=2$, $A=\begin{bmatrix}5&4\\4&5\end{bmatrix}$, $p(x)=\begin{bmatrix}x^1\\x^2\end{bmatrix}$, $D=\begin{bmatrix}1&0\\0&1\end{bmatrix}$, and ψ given by (5.12) and (5.13) with $k_1=1$, $k_2=\frac{1}{2}$, and $z_0=1$. As the reference point we take $(\bar{x}_1,\bar{x}_2,\bar{y}_1,\bar{y}_2)=\left(3,\frac{13}{4},0,-\frac{1}{2}\right)$ and clearly get from (5.14) that $I_3(\bar{x},\bar{y})=\{2\}$ and $I_4(\bar{x},\bar{y})=\{1\}$; all the other index sets are empty. The adjoint GE (5.16) attains the form

$$\begin{aligned}
0 &\in 5u_1+4u_2+\{w_1\in\mathbb{R}|(w_1,-u_1)\in N_4\},\\
0 &\in 4u_1+5u_2+\{w_2\in\mathbb{R}|u_2=-2w_2\},
\end{aligned}\tag{5.21}$$

where the corresponding cone N_4 is computed in (5.15). The second relation in (5.21) is a linear equation from which we obtain $u_2 = -\frac{8}{9}\,u_1$. By inserting this into the first relation of (5.21), one gets

$$0 \in \frac{13}{9}\,u_1 + \{w_1 \in \mathbb{R} | (w_1, -u_1) \in N_4\}.$$

It follows from (5.15) that the above GE has a unique solution $u_1 = 0$. This implies that $u_2 = 0$, which ensures the pseudo-Lipschitzian property of the corresponding solution map S around the reference point $(\bar{x}, \bar{y})$ by virtue of Proposition 5.4.

Acknowledgments. The authors gratefully acknowledge fruitful discussions with Terry Rockafellar regarding calculus rules and with Michal Kočvara regarding mechanical applications. We also thank Alexander Kruger and two anonymous referees for their valuable remarks that helped us to improve the original presentation.

REFERENCES

[1] J.-P. Aubin, *Lipschitz behavior of solutions to convex minimization problems*, Math. Oper. Res., 9 (1984), pp. 87–111.
[2] J. F. Bonnans and A. Shapiro, *Perturbation Analysis of Optimization Problems*, Springer, New York, 2000.
[3] F. H. Clarke, *Optimization and Nonsmooth Analysis*, Wiley, New York, 1983.
[4] A. L. Dontchev and R. T. Rockafellar, *Characterization of strong regularity for variational inequalities over polyhedral convex sets*, SIAM J. Optim., 6 (1996), pp. 1087–1105.
[5] J. Haslinger, M. Miettinen, and P. D. Panagiotopoulos, *Finite Element Methods for Hemivariational Inequalities*, Kluwer, Dordrecht, The Netherlands, 1999.
[6] A. B. Levy, R. A. Poliquin, and R. T. Rockafellar, *Stability of locally optimal solutions*, SIAM J. Optim., 10 (2000), pp. 580–604.
[7] Z.-Q. Luo, J.-S. Pang, and D. Ralph, *Mathematical Programs with Equilibrium Constraints*, Cambridge University Press, Cambridge, UK, 1996.
[8] B. S. Mordukhovich, *Maximum principle in problems of time optimal control with nonsmooth constraints*, J. Appl. Math. Mech., 40 (1976), pp. 960–969.
[9] B. S. Mordukhovich, *Metric approximations and necessary optimality conditions for general classes of nonsmooth extremal problems*, Soviet Math. Dokl., 22 (1980), pp. 526–530.
[10] B. S. Mordukhovich, *Approximation Methods in Problems of Optimization and Control*, Nauka, Moscow, 1988.
[11] B. S. Mordukhovich, *Sensitivity analysis in nonsmooth optimization*, in Theoretical Aspects of Industrial Design, D. A. Field and V. Komkov, eds., Proceedings in Applied Mathematics 58, SIAM, Philadelphia, 1992, pp. 32–46.
[12] B. S. Mordukhovich, *Generalized differential calculus for nonsmooth and set-valued mappings*, J. Math. Anal. Appl., 183 (1994), pp. 250–288.
[13] B. S. Mordukhovich, *Lipschitzian stability of constraint systems and generalized equations*, Nonlinear Anal., 22 (1994), pp. 173–206.
[14] B. S. Mordukhovich, *Stability theory for parametric generalized equations and variational inequalities via nonsmooth analysis*, Trans. Amer. Math. Soc., 343 (1994), pp. 609–658.
[15] U. Mosco, *Implicit variational problems and quasi-variational inequalities*, in Nonlinear Operators and the Calculus of Variations, Lecture Notes in Math. 543, Springer-Verlag, Berlin, 1976, pp. 83–156.
[16] J. Nečas and I. Hlaváček, *Mathematical Theory of Elastic and Elasto-Plastic Bodies: An Introduction*, Elsevier, Amsterdam, 1981.
[17] J. V. Outrata, *Optimality conditions for a class of mathematical programs with equilibrium constraints*, Math. Oper. Res., 24 (1999), pp. 627–644.
[18] J. V. Outrata, *A generalized mathematical program with equilibrium constraints*, SIAM J. Control Optim., 38 (2000), pp. 1623–1638.
[19] J. V. Outrata, *On constraint qualifications for mathematical programs with mixed complementarity constraints*, in Applications and Algorithms of Complementarity, M. C. Ferris, O. Mangasarian, and J.-S. Pang, eds., Kluwer, Dordrecht, The Netherlands, 2001, pp. 253–271.

[20] R. A. Poliquin and R. T. Rockafellar, *Tilt stability of a local minimum*, SIAM J. Optim., 8 (1998), pp. 287–299.
[21] S. M. Robinson, *Generalized equations and their solutions.* I. *Basic theory. Point-to-set maps and mathematical programming*, Math. Programming Stud., 10 (1979), pp. 128–141.
[22] R. T. Rockafellar, *The Theory of Subgradients and Its Applications to Problems of Optimization. Convex and Nonconvex Functions*, Heldermann, Berlin, 1981.
[23] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*, Springer, Berlin, 1998.
[24] J. S. Treiman, *Lagrange multipliers for nonconvex generalized gradients with equality, inequality, and set constraints*, SIAM J. Control Optim., 37 (1999), pp. 1313–1329.
[25] J. J. Ye, *Optimality conditions for optimization problems with complementarity constraints*, SIAM J. Optim., 9 (1999), pp. 374–387.
[26] J. J. Ye and X. Y. Ye, *Necessary optimality conditions for optimization problems with variational inequality constraints*, Math. Oper. Res., 22 (1997), pp. 977–997.
[27] R. Zhang, *Problems of hierarchical optimization in finite dimensions*, SIAM J. Optim., 4 (1994), pp. 521–536.

SIAM J. OPTIM.
Vol. 12, No. 1, pp. 170–187

AMPLE PARAMETERIZATION OF VARIATIONAL INCLUSIONS*

A. L. DONTCHEV† AND R. T. ROCKAFELLAR‡

Abstract. For a general category of variational inclusions in finite dimensions, a class of parameterizations, called "ample" parameterizations, is identified that is rich enough to provide a full theory of Lipschitz-type properties of solution mappings without the need to resort to the auxiliary introduction of canonical parameters. Ample parameterizations also support a detailed description of the graphical geometry that underlies generalized differentiation of solutions mappings. A theorem on proto-derivatives is thereby obtained. The case of a variational inequality over a polyhedral convex set is given special treatment along with an application to minimizing a parameterized function over such a set.

Key words. variational inequalities, calmness, Aubin continuity, Lipschitzian localizations, graphical derivatives, sensitivity of minimizers, variational analysis

AMS subject classifications. 49K40, 90C31, 49J52

PII. S1052623400371016

1. Introduction.

This paper is concerned with implicit function type results for parameterized variational inclusions (generalized equations) of the broad form

$$f(w,x) + F(x) \ni 0, \tag{1.1}$$

where $w \in \mathbb{R}^d$ is the parameter, $x \in \mathbb{R}^n$ is the solution, $f : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}^m$ is a smooth (i.e., $\mathcal{C}^1$) function, and $F : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$ is a set-valued mapping with closed graph. The focus is on local properties of the solution mapping

$$S : w \mapsto S(w) = \big\{x \,\big|\, f(w,x) + F(x) \ni 0\big\} \tag{1.2}$$

at a pair (w_*, x_*) with $x_* \in S(w_*)$. We investigate Lipschitz-type properties such as calmness, Aubin continuity, and Lipschitzian localization, as well as graphical properties connected with generalized differentiation.

It is well understood that in order to make progress in this area the parameterization has to be "rich enough." A standard technique for ensuring such richness is to introduce explicitly, alongside of w, the so-called *canonical parameter* y that corresponds to perturbing the right side in (1.1) to

$$f(w,x) + F(x) \ni y, \tag{1.3}$$

and then to work with extended mapping $\widetilde{S} : \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}^n$ given by

$$\widetilde{S} : (w,y) \mapsto \widetilde{S}(w,y) = \big\{x \,\big|\, f(w,x) + F(x) \ni y\big\}. \tag{1.4}$$

Results obtained for $\widetilde{S}$ can be specialized to S by taking $y = 0$. That approach seems inefficient, though, since the extended inclusion in (1.3) could also be written like (1.1):

$$\tilde{f}(\tilde{w},x) + F(x) \ni 0, \quad \text{where } \tilde{w} = (w,y) \text{ and } \tilde{f}(\tilde{w},x) = f(w,x) - y. \tag{1.5}$$

*Received by the editors April 21, 2000; accepted for publication February 8, 2001; published electronically July 2, 2001. This research was undertaken under grant DMS–9803089 from the National Science Foundation.

http://www.siam.org/journals/siopt/12-1/37101.html

†Mathematical Reviews, Ann Arbor, MI 48107-8604 (ald@ams.org).

‡Department of Mathematics, University of Washington, Seattle, WA 98195 (rtr@math.washington.edu).

It would be preferable to capture the needed richness of the parameterization through an assumption on (1.1) itself, moreover in a manner that provides more flexibility by being merely local. We accomplish that here through the following concept.

DEFINITION 1.1 (ample parameterization). *The variational inclusion* (1.1) *will be called* amply parameterized *at a pair* $(w_*, x_*) \in \operatorname{gph} S$ *if the partial Jacobian matrix* $\nabla_w f(w_*, x_*)$ *for* f *with respect to* w *at* (w_*, x_*) *has full rank:*

$$\operatorname{rank} \nabla_w f(w_*, x_*) = m, \qquad \nabla_w f(w_*, x_*) \in \mathbb{R}^{m \times d}. \tag{1.6}$$

Obviously this condition is fulfilled at every point $(\tilde{w}_*, x_*) = (w_*, y_*, x_*)$ in the graph of the extended mapping $\tilde{S}$ in (1.4), viewed as in (1.5). Hence ample parameterization can always be enforced by passing from S to $\widetilde{S}$, in confirmation of the standard technique.

Supplied with this concept, we begin by studying the relationship between S and an auxiliary mapping S_* at (w_*, x_*) of the general type

$$S_* : y \mapsto S_*(y) = \big\{ x \,\big|\, f_*(x) + F(x) \ni y \big\}, \tag{1.7}$$

where f_* denotes any (smooth) *first-order approximation* to $f(w_*, \cdot)$ at x_* in the sense that

$$f_*(x_*) = f(w_*, x_*) \quad \text{and} \quad \nabla f_*(x_*) = \nabla_x f(w_*, x_*). \tag{1.8}$$

Among the prime candidates for f_* are the simple restriction $f_*(x) = f(w_*, x)$ or its linearization $f_*(x) = f(w_*, x_*) + \nabla_x f(w_*, x_*)(x - x_*)$. Our results, however, depend only on the assumption in (1.7) that (1.8) holds, so in stating them in terms of S_* we achieve a more efficient presentation which emphasizes what is truly essential.

Note that S_* can itself be viewed as a solution mapping in this context, namely one in which there is only a canonical parameterization. Indeed, the choice $f_*(x) = f(w_*, x)$ corresponds to $S_*(y) = \widetilde{S}(w_*, y)$. In comparing properties of S and S_* we continue a long tradition coming from the classical implicit function theorem, where $F = 0$ and the mapping $w \mapsto \big\{ x \,\big|\, f(w, x) = 0 \big\}$ is compared to the mapping $y \mapsto \big\{ x \,\big|\, f(w_*, x) = y \big\}$ or its linearization. Our contribution is to develop the comparison definitively not just for one, but for several key properties in our general setting, while employing the concept of ample parameterization to achieve statements that are more succinct and convenient.

Sections 2, 3, and 4 follow this pattern for the properties of calmness, Aubin continuity, and Lipschitzian localization, respectively. In each case, under ample parameterization, the property in question holds for S if and only if it holds for S_*. Even without ample parameterization, if the property holds for S_* it must hold for S as well.

In section 5 we show, again under ample parameterization, that S is graphically Lipschitzian if and only if F is graphically Lipschitzian. Furthermore, we demonstrate in section 6 that such equivalence carries over to proto-differentiability of S versus that of F, and we obtain a corresponding formula for the proto-derivatives, which reveals that they are given as solutions to an auxiliary variational inclusion.

In section 7 we specialize to the case of F being the normal cone mapping N_C to a convex set C; that is, the case where (1.1) is a *variational inequality*. We take advantage of the fact that N_C is then graphically Lipschitzian, and when C is polyhedral, N_C is proto-differentiable. From the resulting formula for proto-derivatives,

we show that when the derivative mapping is convex-valued the proto-differentiability turns into the stronger property of semidifferentiability.

Finally, in section 8 we apply our results to an optimization problem with perturbations only in the cost function. We show that the standard second-order sufficient optimality condition is equivalent to the combination of optimality at the reference point and calmness of the stationary point mapping. Moreover the strong second-order sufficient condition is equivalent to the Lipschitzian localization property of the mapping that gives local minimizers. A formula for semiderivatives of this mapping is also provided.

A separate paper [6] is devoted to applications of these results to the perturbation of saddle points in convex optimization.

Throughout, any norm is denoted by $\|\cdot\|$ and $B_a(x)$ is the closed ball of radius a centered at x. The graph of a set-valued mapping $\Gamma : \mathbb{R}^p \rightrightarrows \mathbb{R}^n$ is the set $\operatorname{gph}\Gamma = \big\{(z,x) \in \mathbb{R}^p \times \mathbb{R}^n \,\big|\, x \in \Gamma(z)\big\}$ and the inverse of Γ is $\Gamma^{-1} : x \mapsto \big\{z \in \mathbb{R}^p \,\big|\, x \in \Gamma(z)\big\}$.

2. Calmness. To start, we consider a graphically localized version of the "upper-Lipschitz continuity" property introduced for set-valued mappings by Robinson [21]. For functions, the property goes back earlier to Clarke [1], who called it "calmness," and that is the term we prefer here in line with the recent book [23].

DEFINITION 2.1 (calmness). *A mapping $\Gamma : \mathbb{R}^p \rightrightarrows \mathbb{R}^n$ is said to be calm at z_* for isolated x_* when $(z_*, x_*) \in \operatorname{gph}\Gamma$ and there exist neighborhoods U of x_* and V of z_* along with a constant γ such that*

$$\|x - x_*\| \le \gamma \|z - z_*\| \text{ for all } z \in V \text{ and } x \in \Gamma(z) \cap U.$$

This condition implies that $\Gamma(z_*) \cap U = \{x_*\}$, so x_* is an isolated point of $\Gamma(z_*)$, hence the terminology; but calmness can also be defined in a broader sense which reduces to the present one when x_* is an isolated point, yet has meaning even when x_* is not isolated (cf. [23, p. 399]). The broader concept will not enter here. For single-valued mappings, there is no difference.

The calmness in Definition 2.1 was formally introduced by Dontchev [3] as the "local upper-Lipschitz property at a point in the graph" of a mapping. Earlier, without giving it a name, Rockafellar [22] characterized it in terms of the graphical derivatives of the set-valued mapping. That result will be applied in section 6. For recent studies of calmness in the context of mathematical programming, see Klatte [9] and Levy [12]. Note that in the latter paper the term "calmness" is used for a different property.

The following theorem for variational inclusions furnishes a general result of implicit function type for the calmness property.

THEOREM 2.2 (criterion for calmness). *The mapping S is calm at w_* for isolated x_* when the mapping S_* is calm at 0 for isolated x_*. Under the ample parameterization condition* (1.6)*, moreover, the two assertions are equivalent.*

We will deduce Theorem 2.2 from another result which we state next.

THEOREM 2.3 (calmness in composition). *Consider a mapping $N : \mathbb{R}^d \rightrightarrows \mathbb{R}^n$ of the form $N(w) = \big\{x \,\big|\, x \in M(h(w,x))\big\}$, where $M : \mathbb{R}^p \rightrightarrows \mathbb{R}^n$ is set-valued and $h : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}^p$ is $\mathcal{C}^1$. Let (w_*, x_*) be such that $\nabla_x h(w_*, x_*) = 0$. If M is calm at $z_* = h(w_*, x_*)$ for isolated x_*, then N is calm at w_* for isolated x_*.*

Proof. First, since $\nabla_x h(w_*, x_*) = 0$, we know that for any real $\lambda > 0$ and neighborhoods W of w_*, U of x_* and V of z_* there exist positive reals a, b, and c such that the balls $B_a(w_*)$, $B_b(x_*)$, and $B_c(z_*)$ are contained in W, U, and V, respectively, and for any fixed $w \in B_a(w_*)$ the function $x \mapsto h(w,x)$ is Lipschitz continuous on $B_b(x_*)$ with a Lipschitz constant λ. Of course, the radii a and b can be chosen

arbitrarily small, and then c can be made arbitrary small as well, independently of the initial choice of λ. Let κ be an associated Lipschitz constant of the function $w \mapsto h(w,x)$ on $B_a(w_*)$, independent of $x \in B_b(x_*)$.

Suppose M is calm at z_* for isolated x_* with neighborhoods V' of z_* and U' of x_* and constant γ. Choose

$$0 < \lambda < 1/\gamma. \tag{2.1}$$

By the property of h just mentioned, there exist a, b, and c such that $B_b(x_*) \subset U'$ and $B_c(z_*) \subset V'$, and moreover with the property that for any $w \in B_a(w_*)$ the function $h(w,\cdot)$ is Lipschitz continuous on $B_b(x_*)$ with a Lipschitz constant λ. Choose a and b smaller if necessary so that

$$\lambda a + \kappa b \le c. \tag{2.2}$$

Let $w \in B_a(w_*)$ and $x \in N(w) \cap B_b(x_*)$. Then $x \in M(h(w,x)) \cap B_b(x_*)$. Using (2.2) we have $\|h(w,x) - z_*\| = \|h(w,x) - h(w_*,x_*)\| \le \lambda a + \kappa b \le c$. From the calmness of M we then have $\|x - x_*\| \le \gamma\|h(w,x) - z_*\| \le \gamma\lambda\|x - x_*\| + \gamma\kappa\|w - w_*\|$; hence

$$\|x - x_*\| \le \frac{\gamma\kappa}{1 - \lambda\kappa}\|w - w_*\|.$$

Therefore the mapping N is calm at w_* for x_* with constant $\gamma\kappa/(1 - \lambda\kappa)$. □

Theorem 2.3 is a purely metric result and can be formulated in terms only of the constants involved. Accordingly, there is no real need to have $\nabla_x h(w_*,x_*) = 0$ or even to have h be differentiable. All that is required, as seen through the proof, is for h to be Lipschitz continuous in x with a "sufficiently small" Lipschitz constant. In fact the result can be stated in a context of metric spaces.

In the proof of Theorem 2.2, still ahead, we will also employ the following lemma, where the classical implicit function theorem comes in.

LEMMA 2.4 (reparameterization). *Under the ample parameterization condition* (1.6)*, and for a function f_* satisfying the condition* (1.8)*, there exist neighborhoods U, V, and W of x_*, $y = 0$ and w_*, respectively, and a $\mathcal{C}^1$ function $\omega : U \times V \to W$ such that*

(i) *$y + f(\omega(x,y),x) = f_*(x)$ for every $y \in V$ and $x \in U$,*

(ii) *$\omega(x_*,0) = w_*$ and $\nabla_x \omega(x_*,0) = 0$.*

Proof. Let $B := \nabla_w f(w_*,x_*)$; by assumption, this matrix in $\mathbb{R}^{m\times d}$ has full row rank m. In terms of the transpose $B^\top$, consider the system of equations

$$\begin{aligned} w - w_* + B^\top z &= 0,\\ y + f(w,x) - f_*(x) &= 0, \end{aligned} \tag{2.3}$$

where (w,z) is the variable and (x,y) is the parameter. Clearly, $(w_*,0)$ is a solution of (2.3) for the parameter choice $(x_*,0)$. The Jacobian J at $(w_*,0,x_*,0)$ of the function of (w,z) on left side of (2.3) has the form

$$J = \begin{bmatrix} I & B^\top \\ B & 0 \end{bmatrix},$$

where I is the identity. It is well known that when B has full row rank the matrix J is nonsingular. Hence, from the classical implicit function theorem, we conclude that,

locally around $(w_*, 0, x_*, 0)$, there exists a $\mathcal{C}^1$ function $\Omega : (x, y) \mapsto (\omega(x, y), \zeta(x, y))$ such that

$$\begin{aligned} &\omega(x, y) - w_* + B^\top \zeta(x, y) = 0, \\ &y + f(\omega(x, y), x) - f_*(x) = 0 \end{aligned} \tag{2.4}$$

with $\Omega(x_*, 0) = (w_*, 0)$. This yields (i) and the first condition in (ii). By differentiating the system we see further that $J\nabla_x\Omega(x, y)$ must vanish locally, and since J is nonsingular this implies that $\nabla_x\Omega(x, y)$ vanishes locally. In particular, then, $\nabla_x\omega(x_*, 0) = 0$. □

Proof of Theorem 2.2. From the definitions of S and S_* in (1.2) and (1.7) we have $x \in S(w)$ if and only if $x \in S_*(y)$ for $y = f_*(x) - f(w, x)$. Thus, we can write

$$S(w) = \big\{x \,\big|\, x \in S_*\big(f_*(x) - f(w, x)\big)\big\}. \tag{2.5}$$

By taking $h(w, x) = f_*(x) - f(w, x)$, which has $\nabla_x h(w_*, x_*) = 0$ by virtue of (1.8), we can put this in the framework of Theorem 2.3 with $M = S_*$. This lets us conclude that calmness of S_* implies calmness of S.

Assume now that the ample parameterization condition (1.6) holds and consider a mapping ω as guaranteed in Lemma 2.4 with respect to certain neighborhoods U, V, and W. Fix $y \in V$. If $x \in S_*(y) \cap U$ and $w = \omega(x, y)$, then $w \in W$ and $y + f(w, x) = f_*(x)$; hence $x \in S(w) \cap U$. Conversely, if $x \in S(w(x, y)) \cap U$, then clearly $x \in S_*(y) \cap U$. Thus,

$$S_*(y) \cap U = \big\{x \,\big|\, x \in S(\omega(x, y)) \cap U\big\}. \tag{2.6}$$

Since calmness of S at w_* for isolated x_* is local property of the graph of S relative to the point (w_*, x_*), this holds if and only if the same holds for the truncated mapping $S_U : w \mapsto S(w) \cap U$. That equivalence is valid for S_* as well. Applying Theorem 2.3 now in the context of (2.6) with $h = \omega$, we get the desired equivalence for S versus S_*. □

3. Aubin property. The idea behind the Aubin property, which Aubin called "pseudo-Lipschitz continuity," can be traced back to the original proofs of the Lyusternik and Graves theorems; see [2], [4], [7], [11], and [23] for discussions. This property is known to correspond, with respect to taking inverses of mappings, to "metric regularity," a condition which plays a major role in optimization.

Definition 3.1 (Aubin property). *A mapping $\Gamma : \mathbb{R}^p \rightrightarrows \mathbb{R}^n$ is said to have the Aubin continuity property at z_* for x_* when $(z_*, x_*) \in \operatorname{gph}\Gamma$ and there exist neighborhoods U of x_* and V of z_* along with a constant γ such that*

$$z', z'' \in V,\ x' \in \Gamma(z') \cap U \implies \exists\, x'' \in \Gamma(z'') \text{ with } \|x' - x''\| \le \gamma\|z' - z''\|.$$

Keeping the pattern of the preceding section, we establish a result about the Aubin property that is completely parallel to the one about calmness in the preceding section.

Theorem 3.2 (criterion for Aubin property). *The mapping S has the Aubin property at w_* for x_* when the mapping S_* has the Aubin property at 0 for x_*. Under the ample parameterization condition* (1.6)*, moreover, the two assertions are equivalent.*

Not only is the statement of Theorem 3.2 completely parallel to that of Theorem 2.2, the proofs are parallel as well. The key is a composition rule that can be regarded as a version of the Lyusternik–Graves theorem.

THEOREM 3.3 (Aubin property in composition). *Consider a mapping $N : \mathbb{R}^d \rightrightarrows \mathbb{R}^n$ of the form $N(w) = \{x \,|\, x \in M(h(w,x))\}$, where $M : \mathbb{R}^p \rightrightarrows \mathbb{R}^n$ is set-valued with closed graph and $h : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}^p$ is $\mathcal{C}^1$. Let (w_*, x_*) be such that $\nabla_x h(w_*, x_*) = 0$. If M has the Aubin property at $z_* = h(w_*, x_*)$ for x_*, then N has the Aubin property at w_* for x_*.*

Proof. Let the mapping M have the Aubin property at z_* for x_* with neighborhoods V' of z_* and U' of x_* and a constant γ. Let λ satisfy (2.1) and choose the constants a, b, and c as in the proof of Theorem 2.3. Choose a smaller if necessary so that

$$\frac{4\gamma\kappa a}{1-\gamma\lambda} \le b. \tag{3.1}$$

Let $w', w'' \in B_a(w_*)$ and let $x' \in N(w') \cap B_{b/2}(x_*)$. Then $x' \in M(h(w', x')) \cap B_{b/2}(x_*)$. We get from the Aubin property of M the existence of $x_1 \in M(h(w'', x'))$ such that $\|x_1 - x'\| \le \gamma \|h(w', x') - h(w'', x')\| \le \gamma\kappa \|w' - w''\|$. Also, through (3.1),

$$\|x_1 - x_*\| \le \|x_1 - x'\| + \|x' - x_*\| \le \gamma\kappa\|w' - w''\| + \|x' - x_*\| \le \gamma\kappa(2a) + \frac{b}{2} \le b,$$

and consequently $\|h(w'', x_1) - z_*\| \le \lambda a + \kappa b \le c$, from (2.2). Hence, from the Aubin property of M there exists $x_2 \in M(h(w'', x_1))$ such that

$$\|x_2 - x_1\| \le \gamma \|h(w'', x_1) - h(w'', x')\| \le \gamma\lambda \|x_1 - x'\| \le (\gamma\lambda)\gamma\kappa \|w' - w''\|.$$

By induction, we obtain a sequence $x_1, x_2, \ldots, x_k, \ldots$ with $x_k \in M(h(w'', x_{k-1}))$ and $\|x_k - x_{k-1}\| \le (\gamma\lambda)^{k-1}\gamma\kappa\|w' - w''\|$. Setting $x_0 = x'$ and using (3.1), we get

$$\begin{aligned} \|x_k - x_*\| &\le \|x_0 - x_*\| + \sum_{j=1}^{k} \|x_j - x_{j-1}\| \\ &\le \frac{b}{2} + \sum_{j=0}^{k-1} (\gamma\lambda)^j \gamma\kappa \|w' - w''\| \le \frac{b}{2} + \frac{2a\gamma\kappa}{1-\gamma\lambda} \le b; \end{aligned}$$

hence $\|h(w'', x_k) - z_*\| \le \lambda a + \kappa b \le c$. Then there exists $x_{k+1} \in M(h(w'', x_k))$ such that $\|x_{k+1} - x_k\| \le \gamma \|h(w'', x_k) - h(w'', x_{k-1})\| \le \gamma\lambda \|x_k - x_{k-1}\| \le (\gamma\lambda)^k \gamma\kappa \|w' - w''\|$, and the induction step is complete.

The sequence $\{x_k\}$ is Cauchy and hence convergent to some $x'' \in B_a(x_*) \subset U'$. From the closedness of $\operatorname{gph} M$ that has been assumed and the continuity of h we deduce that $x'' \in M(h(w'', x'')) \cap U'$; hence, $x'' \in N(w'')$. Furthermore, using the estimate

$$\|x_k - x'\| \le \sum_{j=1}^{k} \|x_j - x_{j-1}\| \le \sum_{j=0}^{k-1} (\gamma\lambda)^j \gamma\kappa \|w' - w''\| \le \frac{\gamma\kappa}{1-\gamma\lambda} \|w' - w''\|$$

we obtain, on passing to the limit with respect to $k \to \infty$, that $\|x'' - x'\| \le \gamma' \|w' - w''\|$. Thus, N has the Aubin property at 0 for x_* with constant $\gamma' = (\gamma\kappa)/(1-\gamma\lambda)$. □

Proof of Theorem 3.2. Repeat the argument in the proof of Theorem 2.2, simply replacing the composition rule in Theorem 2.3 by the one in Theorem 3.3. □

4. Lipschitzian localization. The Lipschitzian localization property is a looser form of the smooth localization property that appears in the classical implicit function theorem. In the context of variational inequalities, Lipschitzian localization is the property in Robinson's "strong regularity" theorem [20]; see [10], [11], [17], and [23] for more on this subject.

DEFINITION 4.1 (Lipschitzian localization). *A mapping $\Gamma : \mathbb{R}^p \rightrightarrows \mathbb{R}^n$ is said to have a single-valued Lipschitzian localization at z_* for x_* when $(z_*, x_*) \in \operatorname{gph}\Gamma$ and there exist neighborhoods U of x_* and V of z_* such that the mapping $V \ni z \mapsto \Gamma(z) \cap U$ is single-valued and Lipschitz continuous.*

For this property we have an analog of Theorems 2.2 and 3.2 in the following mode.

THEOREM 4.2 (criterion for Lipschitzian localization). *The mapping S has a single-valued Lipschitzian localization at w_* for x_* when the mapping S_* has a single-valued Lipschitzian localization at 0 for x_*. Under the ample parameterization condition* (1.6)*, moreover, the two assertions are equivalent.*

Again we establish this by way of a composition rule.

THEOREM 4.3 (Lipschitzian localization in composition). *Consider $N : \mathbb{R}^d \rightrightarrows \mathbb{R}^n$ of the form $N(w) = \{x \,|\, x \in M(h(w,x))\}$ where $M : \mathbb{R}^p \rightrightarrows \mathbb{R}^n$ is set-valued mapping with closed graph and $h : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}^p$ is $\mathcal{C}^1$. Let (w_*, x_*) be such that $\nabla_x h(w_*, x_*) = 0$. If M has a single-valued Lipschitzian localization at $z_* = h(w_*, x_*)$ for x_*, then N has a single-valued Lipschitzian localization at w_* for x_*.*

Proof. Suppose M has a single-valued Lipschitzian localization at z_* for x_* with neighborhoods U and V and a constant γ. In particular then, M has the Aubin property at z_* for x_* with the same constant γ and consequently, as already proved, N has the Aubin property at w_* for x_*. It is sufficient therefore to verify that there exist neighborhoods U' of x_* and W' of w_* such that $N(w) \cap U'$ is a singleton for every $w \in W'$.

Observe that we can choose a neighborhood W of w_* and shrink U if necessary so that the Lipschitz constant λ of the function $h(w, \cdot)$ on U works for any $w \in W$. Suppose that there exist two sequences, x_k^1 and x_k^2, converging to x_* and a sequence w_k converging to w_*, such that $x_k^i \in N(w_k)$, $i = 1, 2$, and $x_k^1 \neq x_k^2$ for a sufficiently large k so that $x_k^i \in U$, $w_k \in W$, and $h(w_k, x_k^i) \in V$. Since $M(h(w_k, x_k^i)) \cap U$ is a singleton for large k, we have $x_k^i = M(h(w_k, x_k^i)) \cap U$, $i = 1, 2$. From the Lipschitz continuity of both $M(\cdot) \cap U$ and $h(w_k, \cdot)$ we finally obtain

$$0 \neq \|x_k^1 - x_k^2\| \leq \gamma \|h(w_k, x_k^1) - h(w_k, x_k^2)\| \leq \gamma\lambda \|x_k^1 - x_k^2\| < \|x_k^1 - x_k^2\|.$$

This contradiction demonstrates that N has the property claimed. □

Proof of Theorem 4.2. Repeat the argument in the proof of Theorem 2.2, simply replacing the composition rule in Theorem 2.3 by the one in Theorem 4.3. □

5. Lipschitzian graphical geometry. Beyond the property of Lipschitzian localization treated in section 4, there is a more subtle kind of Lipschitzian behavior which is especially common for solution mappings without single-valuedness but which, unlike the Aubin property of section 3 or even the calmness property of section 2, does not revolve around comparing values of the mapping at two different points. Instead, this property centers on Lipschitzian geometry of the graph of the mapping. It has strong implications for generalized differentiability.

DEFINITION 5.1 (graphically Lipschitzian mappings). *A mapping $\Gamma : \mathbb{R}^p \rightrightarrows \mathbb{R}^n$ is said to be graphically Lipschitzian at z_* for x_*, and of dimension k in this respect, when $(z_*, x_*) \in \operatorname{gph}\Gamma$ and there is a change of coordinates in $\mathbb{R}^p \times \mathbb{R}^n$ around (z_*, x_*)*

that is $\mathcal{C}^1$ in both directions, under which $\operatorname{gph}\Gamma$ *can be identified locally with the graph in $\mathbb{R}^k \times \mathbb{R}^{p+n-k}$ of a Lipschitz continuous mapping defined around a point $u_* \in \mathbb{R}^k$.*

Background on graphically Lipschitzian mappings can be found in [23]. As a special case, of course, if Γ has a single-valued Lipschitzian localization around $z_* \in \mathbb{R}^p$, then Γ is graphically Lipschitzian of dimension p at z_* for $x_* = \Gamma(z_*)$. The point of Definition 5.1, however, is that many mappings of fundamental interest in variational analysis and optimization can fail to be single-valued and Lipschitz continuous and yet possess hidden properties of Lipschitzian character which deserve to be recognized and placed in service.

An important class of graphically Lipschitzian mappings which by no means need to be single-valued and Lipschitz continuous is furnished by the *maximal monotone* mappings $F : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$; the theory of maximal monotonicity is available in detail in Chapter 12 of [23]. Within this category are the normal cone mappings N_C associated with the nonempty, closed, convex sets C in $\mathbb{R}^n$ and more generally the subgradient mappings $\partial\varphi$ associated with the lower semicontinuous, proper, convex functions φ on $\mathbb{R}^n$. A normal cone mapping will be the focus in the next section. When $F : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is maximal monotone, $\operatorname{gph} F$ is in fact an n-dimensional Lipschitzian manifold in a global sense.

Maximal monotonicity is not the only source of examples. A broad class of normal cone mappings N_C and subgradient mappings $\partial\varphi$ for which the graphical Lipschitzian property prevails without C or φ having to be convex has been developed by Poliquin and Rockafellar [16] under the heading of "prox-regularity" and more specially "strong amenability" (see also 10.24 and 13.46 of [23]). Such sets C and functions φ arise very commonly in optimization. For instance, a set C given by finitely many $\mathcal{C}^2$ equality and inequality constraints is strongly amenable at any point satisfying the Mangasarian–Fromovitz constraint qualification; a function φ is sure to be strongly amenable when it is the sum of the indicator of a strongly amenable set and a function that is $\mathcal{C}^2$ or the maximum of finitely many $\mathcal{C}^2$ functions. The associated mappings N_C and $\partial\varphi$ then likewise furnish choices of F that are graphically Lipschitzian.

The next theorem shows that, under ample parameterization, graphically Lipschitzian properties of the solution mapping S can be derived from those of F by way of the natural correspondence between the graphs of these mappings:

$$(x, -f(w,x)) \in \operatorname{gph} F \iff (w,x) \in \operatorname{gph} S. \tag{5.1}$$

THEOREM 5.2 (criterion for Lipschitzian geometry). *Under the ample parameterization condition* (1.6)*, the mapping S is graphically Lipschitzian of dimension q at w_* for x_* if and only if the mapping F is graphically Lipschitzian of dimension k at x_* for y_*, where*

$$y_* = -f(w_*, x_*), \qquad q = k + d - m.$$

Proof. Define $Q : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^m$ by

$$Q(w,x) = (x, -f(w,x)). \tag{5.2}$$

Then from (5.1), $\operatorname{gph} S = Q^{-1}(\operatorname{gph} F)$. Under the ample parameterization condition the Jacobian $\nabla Q(w_*, x_*)$ of Q at (w_*, x_*) has full rank $n + m$; in particular this requires $d + n \geq n + m$, i.e., $d - m \geq 0$. Therefore, with respect to a neighborhood O of (w_*, x_*), Q^{-1} has the effect of transforming any graphically Lipschitzian manifold

of dimension k in $\mathbb{R}^n \times \mathbb{R}^m$ into one of dimension $k + (d - m)$ in $\mathbb{R}^d \times \mathbb{R}^n$. The equivalence is now immediate. □

COROLLARY 5.3 (maximal monotonicity). *Under the ample parameterization condition, if F is a maximal monotone mapping, $F : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$, then S is graphically Lipschitzian of dimension d at w_* for x_*.*

Proof. When F is maximal monotone, it is everywhere graphically Lipschitzian of dimension n (cf. [23, 12.15]). Then, by virtue of Theorem 5.2, S is graphically Lipschitzian of dimension $n + d - n = d$ at w_* for x_*. □

COROLLARY 5.4 (strong amenability). *Under the ample parameterization condition, if F is a normal cone mapping N_C or subgradient mapping $\partial\varphi$ for a set C or function φ that is strongly amenable at x_*, then S is graphically Lipschitzian of dimension d at w_* for x_*.*

Proof. Here we rely on the graphically Lipschitzian behavior of such normal cone mappings and subgradient mappings as noted prior to the statement of Theorem 5.2. □

In order to tie Theorem 5.2 in with the patterns of equivalence in the preceding sections, it is also worth stating the following elementary consequence.

COROLLARY 5.5 (equivalent geometries in approximation). *The mapping S_* is graphically Lipschitzian of dimension k at 0 for x_* if and only if F is graphically Lipschitzian of dimension k at x_* for y_*, where $y_* = -f_*(x_*)$. Thus, under the ample parameterization condition* (1.6)*, S is graphically Lipschitzian of dimension q at w_* for x_* if and only if S_* is graphically Lipschitzian of dimension k at 0 for x_*, where $q = k + d - m$.*

Proof. Theorem 5.2 can be applied to S_* as a special kind of solution mapping, which corresponds to replacing $f(w, x)$ by $g(y, x) = f_*(x) - y$ with y as the new parameter, in $\mathbb{R}^m$ instead of $\mathbb{R}^d$. For g, the condition of ample parameterization is satisfied trivially at $(0, x_*)$. Moreover, $-g(0, x_*) = -f(w_*, x_*) = y_*$. Therefore, S_* is graphically Lipschitzian of dimension q_* at 0 for x_* if and only if F is graphically Lipschitzian of dimension k at x_* for y_*, the relation between q_* and k being like that between q and k in Theorem 5.2, except that d is replaced by m. Then $q_* = k + m - m = k$.

In combination now with the statement about S and F in Theorem 5.2, this observation yields the claimed relationship between S and S_*. □

6. Generalized differentiation. In the graphical context of Theorem 5.2, there is a powerful geometric notion of generalized differentiation which can be used even though S may only be set-valued. One says that S is *proto-differentiable at w_* for x_** when $x_* \in S(w_*)$ and the difference quotient mappings

$$\Delta_\tau S(w_* \,|\, x_*) : w' \;\mapsto\; \tau^{-1}[S(w_* + \tau w') - x_*], \quad \tau > 0,$$

converge graphically as $\tau \searrow 0$; in other words, there is a mapping $D : \mathbb{R}^d \to \mathbb{R}^n \times \mathbb{R}^m$ such that $\operatorname{gph} \Delta_\tau S(w_* \,|\, x_*)$ converges to $\operatorname{gph} D$ as $\tau \to 0$. Proto-differentiability was introduced in [22], and much about it can be found now also in [23]; see [13] and [14] as well, where special properties in the case of a graphically Lipschitzian mapping are laid out.

Proto-differentiability is closely involved with the tangent cone $T_{\operatorname{gph} S}(w_*, x_*)$ to $\operatorname{gph} S$ at (w_*, x_*). This cone is the graph of the mapping $DS(w_* \,|\, x_*) : \mathbb{R}^d \rightrightarrows \mathbb{R}^n$ that in general is called the *graphical derivative* of S at w_* for x_*; by definition,

$$x' \in DS(w_* \,|\, x_*)(w') \quad \Longleftrightarrow \quad (w', x') \in T_{\operatorname{gph} S}(w_*, x_*). \tag{6.1}$$

The graphs of the mappings $\Delta_\tau S(w_* \,|\, x_*)$ are the sets $\tau^{-1}[\operatorname{gph} S - (w_*, x_*)]$, which have $T_{\operatorname{gph} S}(w_*, x_*)$ as their outer set limit ("lim sup") as $\tau \searrow 0$. What makes the property of proto-differentiability special is that the outer limit is required to equal the inner set limit ("lim inf") and thus be a true set limit. As translated to the language of tangent cones, proto-differentiability of S at w_* for x_* means that $\operatorname{gph} S$ is *geometrically derivable* at (w_*, x_*). See [23] for more on this subject. It is clear that when the graphical limit D in the definition of proto-differentiability exists it has to be $DS(w_* \,|\, x_*)$, although the latter has meaning (and uses) even in the absence of proto-differentiability.

The power of proto-differentiability in the presence of Lipschitzian graphical geometry comes from the tight mode of local approximation it affords, in a manner reminiscent of classical differentiability. To appreciate this, consider first the case where S happens to be single-valued and Lipschitz continuous around w_*, with x_* the unique element of $S(w_*)$. Proto-differentiability implies then that the mapping $DS(w_* \,|\, x_*)$ (which in this case could simply be denoted by $DS(w_*)$) is likewise single-valued and Lipschitz continuous and

$$S(w) = S(w_*) + DS(w_* \,|\, x_*)(w - w_*) + o(|w - w_*|), \tag{6.2}$$

where $o(t)$ denotes a term such that $o(t)/t \to 0$ as $t \searrow 0$. This is ordinary differentiability precisely when the mapping $DS(w_* \,|\, x_*)$ is, in addition, linear.

In general, when S and $DS(w_* \,|\, x_*)$ are single-valued (but $DS(w_* \,|\, x_*)$ might not be linear), we speak of the property in (6.2) as the *semidifferentiability* of S at w_* for $x_* = S(w_*)$. For more discussion of semidifferentiability, see [23].

In moving next to the case where S is not necessarily single-valued and Lipschitz continuous but merely graphically Lipschitzian at (w_*, x_*), it is crucial to observe that although the type of approximation in (6.2) depends strongly on the particular coordinate system on the graph, specifically the decomposition into components w and x, the notion of proto-differentiability does not. Because it is based on set convergence in the graph space, proto-differentiability is preserved under changes of coordinates. Therefore, *proto-differentiability of a graphically Lipschitzian mapping S corresponds to the tight mode of local approximation to* $\operatorname{gph} S$ *as in* (6.2)*, but applied obliquely, to a different coordinate system than the* (w, x) *system.*

Note that the mapping $DS(w_* \,|\, x_*)$ is always *positively homogeneous*, since its graph is a cone; one has $DS(w_* \,|\, x_*)(0) \ni 0$ and $DS(w_* \,|\, x_*)(\lambda w') = \lambda DS(w_* \,|\, x_*)(w')$ for all w' when $\lambda > 0$.

Proto-differentiability has only been described so far in terms of S, but of course the concept also applies to F, and this now comes on stage as well. For a pair $(x_*, y_*) \in \operatorname{gph} F$ we have

$$y' \in DF(x_*, y_*)(x') \iff (x', y') \in T_{\operatorname{gph} F}(x_*, y_*). \tag{6.3}$$

If F happens, for example, to be single-valued and, at x_*, is differentiable in the usual sense, then F is proto-differentiable at x_* for $y_* = F(x_*)$ with $DF(x_* \,|\, y_*)$ being the usual derivative mapping (for which $DF(x_*)$ is then a simpler notation).

THEOREM 6.1 (proto-derivative formula). *Under the ample parameterization condition* (1.6)*, the mapping S is proto-differentiable at w_* for x_* if and only if the mapping F is proto-differentiable at x_* for $y_* = -f(w_*, x_*)$. Then*

$$\begin{aligned} &DS(w_* \,|\, x_*)(w') = \big\{x' \,\big|\, g(w', x') + G(x') \ni 0\big\}, \ \textit{where} \\ &g(w', x') = \nabla_w f(w_*, x_*)w' + \nabla_x f(w_*, x_*)x' \ \textit{ and } \ G(x') = DF(x_* \,|\, y_*)(x'). \end{aligned} \tag{6.4}$$

Proof. We appeal again to the setup in the proof of Theorem 5.2, where $\operatorname{gph} S = Q^{-1}(\operatorname{gph} F)$ for the mapping Q in (5.2). Because the Jacobian of Q has full rank under ample parameterization, we can determine the tangent cone $T_{\operatorname{gph} S}(w_*, x_*)$ by the general rule of variational analysis given in 6.7 of [23], obtaining

$$T_{\operatorname{gph} S}(w_*, x_*) = \big\{(w', x') \,\big|\, \nabla Q(w_*, x_*)(w', x') \in T_{\operatorname{gph} F}(x_*, y_*)\big\}. \tag{6.5}$$

This furnishes, through the formulas for $DS(w_*, x_*)$ and $DF(x_*, y_*)$ in (6.1) and (6.3), the formula in (6.4). A parallel formula holds for the corresponding "derivable cones" to $\operatorname{gph} S$ and $\operatorname{gph} F$, which are defined with outer set limits replaced by inner set limits. The geometric derivability of $\operatorname{gph} F$ at (x_*, y_*) thus corresponds to the geometric derivability of $\operatorname{gph} S$ at (w_*, x_*). Hence we have the equivalence between proto-differentiability of S and that of F. □

COROLLARY 6.2 (derivative criterion for calmness). *Under the ample parameterization condition* (1.6) *and the assumption that F is proto-differentiable at x_* for $y_* = -f(w_*, x_*)$, the mapping S is calm at w_* for isolated x_* if and only if*

$$\nabla_x f(w_*, x_*)x' + DF(x_* \,|\, y_*)(x') \ni 0 \implies x' = 0. \tag{6.6}$$

Proof. According to the characterization of calmness of set-valued mappings developed in [22, Theorem 4.1] in terms of graphical derivatives, S is calm at w_* for isolated x_* if and only if $DS(w_* \,|\, x_*)(0) = \{0\}$. This criterion translates to (6.6) through the derivative formula in Theorem 6.1. □

The especially attractive feature of Theorem 6.1 is that the graphical derivative of the solution mapping S turns out itself to be a solution mapping in our framework, namely one that corresponds to g and G in place of f and F, with w' as the parameter and x' as the solution. A derivative formula in this pattern was originally exhibited in [22] for a variational inequality with canonical perturbations. That case will be elaborated below.

To make the best use of Theorem 6.1 and Corollary 6.2, one needs to recognize situations where F is proto-differentiable. The example of F single-valued and differentiable has already been mentioned. Other examples emerge from the second-order variational analysis of sets and functions that are *fully amenable*, this being a refinement of the strong amenability in [16] that had a role in the preceding section. For the theory of full amenability and the graphical derivative formulas it provides, along with examples, we refer to [23] and restrict ourselves here to recording the following consequence of Theorem 6.1.

COROLLARY 6.3 (full amenability). *Under the ample parameterization condition* (1.6), *if $F = N_C$ or $F = \partial\varphi$ for a set C or function φ that is fully amenable at x_*, then S is not only graphically Lipschitzian at w_* for x_* but also proto-differentiable there.*

Proof. The graphically Lipschitzian property is implied by Corollary 5.4, inasmuch as full amenability is a special case of strong amenability. The rest comes out of Theorem 6.1 and the fact, just cited, that F is proto-differentiable at x_* for $y_* \in F(x_*)$ when F is of the form described. □

7. Application to variational inequalities. We concentrate now on the special case where S is the solution mapping for a parameterized variational inequality,

$$S(w) = \big\{x \,\big|\, f(w, x) + N_C(x) \ni 0\big\} \tag{7.1}$$

with respect to a nonempty convex set $C \subset \mathbb{R}^n$ that is *polyhedral*. This choice allows us to obtain a quite detailed picture of the geometry of proto-derivatives of S and to

provide a basis for their actual computation. Because of convexity, the vectors y in the normal cone $N_C(x)$ at any $x \in C$ are the ones that satisfy

$$\langle y,\, x' - x\rangle \le 0 \quad \text{for all } x' \in C.$$

Typically in the literature on variational inequalities this condition, with $y = -f(w,x)$, is written in place of the condition $f(w,x) + N_C(x) \ni 0$, but the normal cone version helps to put things into the right framework of set-valued mappings. When $x \notin C$, $N_C(x)$ is interpreted as $\emptyset$.

Our goal is to apply the theory of the preceding sections to $F = N_C$ and make the most of the special properties that follow from C being polyhedral. We say that a mapping is *piecewise polyhedral* when its graph is the union of a collection of finitely many polyhedral (convex) sets. If the mapping is single-valued, this is the same as it being piecewise linear (see [23, 2.48]). For a vector y, we let $y^\perp = \big\{u \,\big|\, \langle y,u\rangle = 0\big\}$. This notation is used in the next theorem in defining the cone K_* that is known as the *critical cone* associated with the variational inequality in (7.1) for $w = w_*$ and $x = x_*$.

THEOREM 7.1 (proto-derivatives for variational inequalities). *Let $F = N_C$ for a polyhedral convex set $C \subset \mathbb{R}^n$ and assume that the ample parameterization condition* (1.6) *holds. Then S is both graphically Lipschitzian of dimension d and proto-differentiable at w_* for x_*, with its proto-derivatives given by an auxiliary variational inequality, namely*

$$\begin{aligned} &DS(w_*\,|\,x_*)(w') = \big\{x' \,\big|\, g(w',x') + N_{K_*}(x') \ \ni\ 0\big\}, \ \ \textit{where}\\ &g(w',x') = \nabla_w f(w_*,x_*)w' + \nabla_x f(w_*,x_*)x' \ \ \textit{and}\ \ K_* = T_C(x_*) \cap f(w_*,x_*)^\perp. \end{aligned} \tag{7.2}$$

Furthermore, the mapping $DS(w_*\,|\,x_*)$ is itself graphically Lipschitzian of dimension d everywhere and is piecewise polyhedral.

Proof. This mainly constitutes a further specialization of Theorems 5.2 and 6.1 along the lines of Corollaries 5.4 and 6.3. When $F = N_C$ with C polyhedral (and nonempty since by blanket assumption we are working with a pair $(w_*, x_*) \in \operatorname{gph} S$), we have F maximal monotone and everywhere proto-differentiable, with the proto-derivative mapping being itself a normal cone mapping; specifically, $DF(x_*\,|\,y_*) = N_{K_*}$ for $K_* = T_C(x_*) \cap y_*^\perp$, which we apply here to $y_* = -f(w_*,x_*)$. (This reduction of $DF(x_*\,|\,y_*)$ to a normal cone mapping depends crucially on C being polyhedral; for details see [21] or the reduction lemma in [5].)

Because the tangent cones to a polyhedral set C are themselves polyhedral, the cone K_* is polyhedral and the mapping N_{K_*} is therefore piecewise polyhedral (see [18] or [23, 12.31]). Recall now the general way that the graph of S corresponded to that of F through a mapping Q as in (5.1) and (5.2). In the context of the auxiliary variational inequality in (7.2), the same holds for $\operatorname{gph} DS(w_*\,|\,x_*)$ versus $\operatorname{gph} N_{K_*}$, and furthermore with a replacement for Q that is a linear mapping. From this it is apparent that $\operatorname{gph} DS(w_*\,|\,x_*)$ inherits the piecewise polyhedrality of $\operatorname{gph} N_{K_*}$. ∎

A proto-derivative formula akin to the one in Theorem 7.1 was originally established in [22], but in terms of canonical parameters. Here we have extended it in terms of ample parameterization as well as provided new information about the graph of the derivative mapping, its piecewise polyhedrality.

COROLLARY 7.2 (piecewise linear geometry). *In the setting of Theorem* 7.1*, the graph of $DS(w_*\,|\,x_*)$ is a piecewise linear manifold of dimension d in the sense of being a Lipschitzian manifold formed as the union of a finite collection of d-dimensional polyhedral sets.*

Proof. Theorem 7.1 reveals that $DS(w_* \,|\, x_*)$ is a mapping of the sort to which Corollary 5.2 applies. Hence $\operatorname{gph} DS(w_* \,|\, x_*)$ is a d-dimensional Lipschitzian manifold, in fact "globally" because this graph is a cone and therefore determined by its properties around the origin. On the other hand, $DS(w_* \,|\, x_*)$ is piecewise polyhedral by Theorem 6.1. That supplies the piecewise linearity of the Lipschitzian mapping underlying the definition of the graphically Lipschitzian property (cf. [23, 12.31] again). In expressing the graph as the union of a finite collection of polyhedral sets, it can be arranged that none of these sets is included in any of the others, and they must then all be of dimension d. □

COROLLARY 7.3 (calmness of variational inequalities). *In the setting of Theorem* 7.1*, the mapping S is calm at w_* for isolated x_* if and only if*

$$\nabla_x f(w_*, x_*)x' + N_{K_*}(x') \ni 0 \quad \Longrightarrow \quad x' = 0.$$

Proof. We get this immediately from Corollary 6.2. □

Especially of interest for proto-differentiability is the case of Theorem 7.1 where S is locally single-valued and Lipschitz continuous. When that holds, the proto-differentiability turns into a stronger property. A critical role in reaching that conclusion can be played by the result in Theorem 4.2, this being an extended version of Robinson's strong regularity theorem [19]. In other work which is closely related, King and Rockafellar [8] obtained a graphical-derivative characterization of single-valuedness for set-valued mappings with a "subinvertibility" property which in particular can be guaranteed through monotonicity. The next theorem could largely be derived as a specialization of that work, but because of a difference in contexts we find it more expedient and illuminating to proceed directly.

Recall here the concept of *semidifferentiability* that was described for single-valued S and $DS(w_* \,|\, x_*)$ in terms of the approximation in (6.2).

THEOREM 7.4 (single-valuedness relations). *Let $F = N_C$ for a polyhedral convex set $C \subset \mathbb{R}^n$ and assume that the ample parameterization condition* (1.6) *holds. Suppose further that S is convex-valued around w_*, in the sense that $S(w)$ is a convex set for all w in some neighborhood of w_*. Then the following properties are equivalent:*

(a) *S is single-valued and Lipschitz continuous on some neighborhood of w_*;*

(b) *$DS(w_* \,|\, x_*)$ is single-valued on some neighborhood of* 0 *(hence everywhere).*

Moreover, then S is semi-differentiable at w_* for x_*, and $DS(w_* \,|\, x_*)$ is not only Lipschitz continuous and positively homogeneous but also piecewise linear.

Proof. Since S is convex-valued, it is single-valued and Lipschitz continuous around w_* if and only it has a single-valued Lipschitzian localization at w_* for x_*. This is critical because this localization property is all that we are able to relate to $DS(w_* \,|\, x_*)$, inasmuch as $DS(w_* \,|\, x_*)$ depends only on the geometry of $\operatorname{gph} S$ at (w_*, x_*).

The proto-differentiability of S at w_* for x_*, which we know from Theorem 7.1, reduces to the semidifferentiability in (6.2) when S is locally single-valued and Lipschitz continuous, as noted earlier (see [23]). Furthermore, from Theorem 7.1 (and Corollary 7.2), the mapping $DS(w_* \,|\, x_*)$, being piecewise polyhedral, must be piecewise linear when it is single-valued (cf. 2.48 and 9.57 of [23]). Thus, (a) implies (b) along with piecewise linear semidifferentiability.

To complete the proof of the theorem, we must show that if (b) holds, then S has a single-valued Lipschitzian localization at w_* for x_*. For this purpose we can invoke Theorem 4.2 in order to transform the task into one of showing that an auxiliary mapping S_* has a single-valued Lipschitzian localization at 0 for x_*, where S_* has the

form (1.7)–(1.8), as in the earlier parts of this paper, except that now $F = N_C$. We specifically choose the function f_* in (1.8) by $f_*(x) = f(w_*, x_*) + \nabla_x f(w_*, x_*)(x - x_*)$, so that

$$\begin{aligned} &S_*(y) = \big\{x \,\big|\, h(y,x) + N_C(x) \ni 0\big\}, \quad \text{where} \\ &h(y,x) = f(w_*, x_*) + \nabla_x f(w_*, x_*)(x - x_*) - y. \end{aligned} \tag{7.3}$$

Because C is polyhedral, the mapping N_C is piecewise polyhedral (cf. [23, 12.31]), and it follows then, because h is linear, that S_* is piecewise polyhedral. Theorem 7.1 is applicable to S_* in place of S, with minor adjustments of notation. It yields the formula

$$DS_*(0 \,|\, x_*)(y') = \big\{x' \,\big|\, -y' + \nabla_x f(w_*, x_*)x' + N_{K_*}(x') \ni 0\big\} \tag{7.4}$$

for the same critical cone K_* as in (7.2), along with the information that $DS_*(0 \,|\, x_*)$ is piecewise polyhedral.

Crucial now will be the general fact that when a set G is polyhedral its tangent cone $T_G(z)$ at a point $z \in G$ coincides in some neighborhood of the origin with the translated set $G - z$. This obviously carries over to piecewise polyhedral sets G as well. Applying it to $G = \operatorname{gph} S_*$ at $z = (0, x_*)$, and remembering that $DS_*(0 \,|\, x_*)$ is the mapping which has $T_{\operatorname{gph} S_*}(0, x_*)$ as its graph, we see that $\operatorname{gph} S_* - (0, x_*)$ coincides with $\operatorname{gph} DS_*(0 \,|\, x_*)$ in a neighborhood of the origin.

In light of this, it will suffice for us to demonstrate that $DS_*(0 \,|\, x_*)$ is single-valued when $DS(w_* \,|\, x_*)$ is single-valued, inasmuch as the single-valuedness of $DS_*(0 \,|\, x_*)$ in combination with its piecewise polyhedrality will imply its Lipschitz continuity (again cf. [23, 2.48 and 9.57]). For arbitrary y', is there one and only one x' satisfying in (7.4) the condition $-y' + \nabla_x f(w_*, x_*)x' + N_{K_*}(x') \ni 0$? Under the ample parameterization condition (1.6), it is possible to write $-y' = \nabla_w f(w_*, x_*)w'$ for some w'. The question then is whether there is one and only one x' satisfying

$$\nabla_w f(w_*, x_*)w' + \nabla_x f(w_*, x_*)x' + N_{K_*}(x') \ni 0.$$

Through our assumption that $DS(w_* \,|\, x_*)$ is single-valued, the answer from formula (7.2) is yes, and we are done. □

PROPOSITION 7.5 (example of convex-valuedness). *In particular, the solution mapping S in* (7.1) *is convex-valued, as postulated in Theorem* 7.4, *when $f(w, x)$ is monotone with respect to $x \in C$, in the sense that*

$$\langle f(w, x') - f(w, x''),\, x' - x'' \rangle \geq 0 \quad \textit{for } x', x'' \in C.$$

Proof. Under this assumption the variational inequality is of monotone type, in which case its set of solutions is convex, as is well known. □

8. Application to minimization over a polyhedral set. In this section we specialize further to the case of a parameterized variational inequality coming out of a minimization problem with fixed linear constraints. This will provide an illustration also of our results on calmness and show how they are related to second-order conditions for optimality. Applications to primal-dual aspects of convex optimization in a format allowing for constraint perturbations will be found in our forthcoming paper [6].

The basic problem we consider here has the form

$$\text{minimize } \varphi(w, x) \text{ over } x \in C, \tag{8.1}$$

where C is a nonempty *polyhedral* (convex) subset of $\mathbb{R}^n$ and the function $\varphi : \mathbb{R}^d \times \mathbb{R}^n$ is of class $\mathcal{C}^2$. For this problem, parameterized by w, the first-order optimality condition is

$$-\nabla_x \varphi(w, x) \in N_C(x), \tag{8.2}$$

and the points x satisfying it are the "quasi-optimal" solutions called *stationary points*. The mapping from w to such points x has the form

$$S : w \mapsto \big\{ x \,\big|\, \nabla_x \varphi(w, x) + N_C(x) \ni 0 \big\} \tag{8.3}$$

and fits our framework as the case of the general mapping S in (1.2) where $m = n$ and

$$f(w, x) = \nabla_x \varphi(w, x), \qquad F(x) = N_C(x). \tag{8.4}$$

The specialization of F to the normal cone mapping N_C for a polyhedral set C was already the topic in the preceding section, so what is new here is merely the specialization of f to $\nabla_x \varphi$. The assumption that $\varphi \in \mathcal{C}^2$ gives us $f \in \mathcal{C}^1$ as required, with

$$\nabla_w f(w, x) = \nabla^2_{xw} \varphi(w, x) \in \mathbb{R}^{n \times d}, \qquad \nabla_x f(w, x) = \nabla^2_{xx} \varphi(w, x) \in \mathbb{R}^{n \times n}, \tag{8.5}$$

and the ample parameterization condition (1.6) for a pair $(w_*, x_*) \in \operatorname{gph} S$ coming out as

$$\operatorname{rank} \nabla^2_{xw} \varphi(w_*, x_*) = n. \tag{8.6}$$

Furnished with this information, it is easy to apply to the stationary point mapping in (8.3) all the results obtained so far in this paper, in particular the ones in section 7, in which the critical cone becomes

$$K_* = T_C(x_*) \cap \nabla_x \varphi(w_*, x_*)^\perp. \tag{8.7}$$

Rather than recording the details of that, we aim here at exploring certain connections between second-order optimality and our results on calmness and Aubin property.

Recall that, in partnership with the first-order condition for optimality that we are now placing on our reference element (w_*, x_*) in taking it to belong to the graph of the mapping S in (8.3), the *standard second-order necessary condition* for local optimality is

$$\langle u, \nabla^2_{xx} \varphi(w_*, x_*) u \rangle \geq 0 \quad \text{for all } u \in K_* \tag{8.8}$$

for the critical cone K_* in (8.7), whereas the *standard second-order sufficient condition* is

$$\langle u, \nabla^2_{xx} \varphi(w_*, x_*) u \rangle > 0 \quad \text{for all nonzero } u \in K_*. \tag{8.9}$$

The *strong second-order sufficient condition* for local optimality is

$$\langle u, \nabla^2_{xx} \varphi(w_*, x_*) u \rangle > 0 \quad \text{for all nonzero } u \in K_* - K_*. \tag{8.10}$$

Because K_* is convex, $K_* - K_*$ is the smallest subspace of $\mathbb{R}^n$ that includes K_*; it is called the *critical subspace* associated with w_* and x_*.

THEOREM 8.1 (calmness of optimal solution mappings). *Under the ample parameterization condition* (8.6)*, the following properties of the stationary point mapping* S *in* (8.3) *are equivalent at the reference pair* $(w_*, x_*) \in \operatorname{gph} S$*:*

(i) *The standard second-order sufficient condition* (8.9) *holds;*

(ii) x_* *is a local minimizer in problem* (8.1) *for* w_**, and* S *is calm at* w_* *for isolated* x_**.*

Proof. According to Corollary 7.3 as applied to $f = \nabla_x \varphi$, we have calmness at w_* for isolated x_* if and only if

$$\nabla^2_{xx}\varphi(w_*, x_*)x' + N_{K_*}(x_* \,|\, y_*)(x') \ni 0 \implies x' = 0. \tag{8.11}$$

On the other hand, we have available the following description of normal vectors to a closed convex cone K in terms of the polar cone K^*, as applied to $K = K_*$:

$$v \in N_{K_*}(u) \iff u \in K_*, \quad v \in K_*^*, \quad u \perp v \tag{8.12}$$

(cf. 11.4(b) of [23]). Therefore, S is calm at w_* for isolated x_* if and only if

$$u \in K_*, \quad -\nabla^2_{xx}\varphi(w_*, x_*)u \in K_*^*, \quad \langle u, \nabla^2_{xx}\varphi(w_*, x_*)u\rangle = 0 \implies u = 0. \tag{8.13}$$

Let (i) hold. Then of course x_* is a local minimizer as described, but is S calm at w_* for x_*? If this were not true, there would exist by (8.11) some $u \neq 0$ satisfying the conditions in (8.13), and that would contradict the inequality $\langle u, \nabla_{xx}\varphi(w_*, x_*)u\rangle > 0$ known from the supposition in (i) that (8.9) is satisfied.

Conversely now, let (ii) hold. Because x_* is a local minimizer, the second-order necessary condition (8.8) must be fulfilled; this can be written as

$$u \in K_* \implies -\nabla^2_{xx} g(w_*, x_*)u \in K_*^*.$$

The calmness of S, as identified with (8.13), eliminates the possibility of there being a nonzero $u \in K_*$ such that the inequality in (8.8) fails to be strict. Thus, the necessary condition (8.8) turns into the sufficient condition (8.9), and (i) is satisfied. □

We investigate next, in association with the stationary point mapping S in (8.3), the mapping

$$S_* : y \mapsto \big\{x \,\big|\, \nabla_x \varphi(w_*, x) + N_C(x) \ni y\big\}, \tag{8.14}$$

which has the form in the general theory of the earlier parts of this paper with $f_*(x) = f(w_*, x) = \nabla_x \varphi(w_*, x)$. From Theorem 3.2 we know that, under the ample parameterization condition (8.6), S has the Aubin property at w_* for x_* if and only if this mapping S_* has that property at 0 for x_*. From Theorem 4.2, likewise under the ample parameterization condition (8.6), S has a single-valued Lipschitzian localization at w_* for x_* if and only if this S_* has such a localization at 0 for x_*.

Something else can be brought into this picture. In [5, Theorem 3] we proved that in a variational inequality like the current one, in which C is polyhedral, the Aubin property and the Lipschitzian localization property are equivalent for S and also for S_*. On the other hand, by a result of Poliquin and Rockafellar [17, Theorem 4.5], the strong second-order sufficient condition (8.10) holds if and only if S_* has the Lipschitzian localization property. By combining these results we arrive at the following characterization.

THEOREM 8.2 (Lipschitzian localization of optimal solution mappings). *Under the ample parameterization condition* (8.6)*, the following properties of the stationary point mapping* S *in* (8.3) *are equivalent at the reference pair* $(w_*, x_*) \in \operatorname{gph} S$*:*

(i) *The strong second-order sufficient condition* (8.10) *holds at* (w_*, x_*)*;*

(ii) S *has a single-valued Lipschitzian localization at* w_* *for* x_* *such that, for all* $(w, x) \in \operatorname{gph} S$ *near* (w_*, x_*)*,* x *is not only a stationary point but a local minimizer in problem* (8.1)*.*

This can be supplemented by a description of the resulting semiderivatives of the mapping S.

THEOREM 8.3 (perturbations of local minimizers). *In the context of the properties in Theorem* 8.2*, the mapping* S *is semidifferentiable at* w_**; thus* (6.2) *holds. Moreover, in this case* $DS(w_* \,|\, x_*)$ *is a piecewise linear mapping such that* $DS(w_* \,|\, x_*)(w')$ *is the unique solution* x' *to the variational inequality*

$$\nabla^2_{xw}\varphi(w_*, x_*)w' + \nabla^2_{xx}\varphi(w_*, x_*)x' + N_{K_*}(x') \ni 0, \tag{8.15}$$

or, equivalently, the unique optimal solution to the quadratic programming subproblem

$$\text{minimize } \langle x', \nabla^2_{xw}\varphi(w_*, x_*)w'\rangle + \frac{1}{2}\langle x', \nabla^2_{xx}\varphi(w_*, x_*)x'\rangle \text{ over } x' \in K_*. \tag{8.16}$$

Proof. We apply Theorem 7.4 and then get the description of $DS(w_* \,|\, x_*)(w')$ through (8.15) by specializing formula (7.2) of Theorem 7.1. Next, we observe that (8.15) is the first-order optimality condition for the problem in (8.16), and, because of the second-order sufficiency we have at hand, it gives local minimizers. □

REFERENCES

[1] F. H. CLARKE, *A new approach to Lagrange multipliers*, Math. Oper. Research, 1 (1976), pp. 165–174.

[2] A. L. DONTCHEV, *The Graves theorem revisited*, J. Convex Anal., 3 (1996), pp. 45–53.

[3] A. L. DONTCHEV, *Characterizations of Lipschitz stability in optimization*, in Recent Developments in Well-Posed Variational Problems, Math. Appl. 331, Kluwer, Dordrecht, The Netherlands, 1995, pp. 95–115.

[4] A. L. DONTCHEV AND W. W. HAGER, *An inverse mapping theorem for set-valued maps*, Proc. Amer. Math. Soc., 121 (1994), pp. 481–489.

[5] A. L. DONTCHEV AND R. T. ROCKAFELLAR, *Characterizations of strong regularity for variational inequalities over polyhedral convex sets*, SIAM J. Optim., 6 (1996), pp. 1087–1105.

[6] A. L. DONTCHEV AND R. T. ROCKAFELLAR, *Primal-dual solution perturbations in convex optimization*, Set-Valued Anal., to appear.

[7] A. D. IOFFE, *Metric regularity and subdifferential calculus*, Uspekhi Mat. Nauk, 55 (2000), pp. 103–162.

[8] A. J. KING AND R. T. ROCKAFELLAR, *Sensitivity analysis for nonsmooth generalized equations*, Math. Program. Ser. A, 55 (1992), pp. 193–212.

[9] D. KLATTE, *Upper Lipschitz behavior of solutions to perturbed* $C^{1,1}$ *programs*, Math. Program., 88 (2000), pp. 285–311.

[10] D. KLATTE AND B. KUMMER, *Strong stability in nonlinear programming revisited*, J. Austral. Math. Soc. Ser. B, 40 (1999), pp. 336–352.

[11] B. KUMMER, *Lipschitzian and pseudo-Lipschitzian inverse functions and applications to nonlinear optimization*, in Mathematical Programming with Data Perturbations, Lecture Notes in Pure and Appl. Math. 195, Dekker, New York, 1998, pp. 201–222.

[12] A. B. LEVY, *Calm minima in parameterized finite-dimensional optimization*, SIAM J. Optim., 11 (2000), pp. 160–178.

[13] A. B. LEVY AND R. T. ROCKAFELLAR, *Sensitivity analysis of solutions to generalized equations*, Trans. Amer. Math. Soc., 345 (1994), pp. 661–671.

[14] A. B. LEVY AND R. T. ROCKAFELLAR, *Proto-derivatives and the geometry of solution mappings in nonlinear programming*, in Nonlinear Optimization and Applications, G. Di Pillo and F. Giannessi, eds., Plenum, New York, 1996, pp. 249–260,

[15] A. B. LEVY, R. A. POLIQUIN, AND R. T. ROCKAFELLAR, *Stability of locally optimal solutions*, SIAM J. Optim., 10 (2000), pp. 580–604.

[16] R. A. Poliquin and R. T. Rockafellar, *Prox-regular functions in variational analysis*, Trans. Amer. Math. Soc., 348 (1996), pp. 1805–1838.
[17] R. A. Poliquin and R. T. Rockafellar, *Tilt stability of a local minimum*, SIAM J. Optim., 8 (1998), pp. 287–299.
[18] S. M. Robinson, *An Implicit Function Theorem for Generalized Variational Inequalities*, Technical summary report 1672, University of Wisconsin-Madison, 1976.
[19] S. M. Robinson, *Strongly regular generalized equations*, Math. Oper. Res., 5 (1980), pp. 43–62.
[20] S. M. Robinson, *Some continuity properties of polyhedral multifunctions*, Math. Programming Stud., 14 (1981), pp. 206–214.
[21] S. M. Robinson, *An implicit-function theorem for a class of nonsmooth functions*, Math. Oper. Res., 16 (1991), pp. 292–309.
[22] R. T. Rockafellar, *Proto-differentiability of set-valued mappings and its applications in optimization*, Ann. Inst. H. Poincaré Anal. Non Linéaire, 6 (1989), suppl., pp. 449–482.
[23] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*, Springer-Verlag, Berlin, 1998.

SIAM J. OPTIM.
Vol. 12, No. 1, pp. 188–208

MINIMIZING A QUADRATIC OVER A SPHERE*

WILLIAM W. HAGER†

Abstract. A new method, the sequential subspace method (SSM), is developed for the problem of minimizing a quadratic over a sphere. In our scheme, the quadratic is minimized over a subspace which is adjusted in successive iterations to ensure convergence to an optimum. When a sequential quadratic programming iterate is included in the subspace, convergence is locally quadratic. Numerical comparisons with other recent methods are given.

Key words. trust region subproblem, large-scale optimization, sparse optimization, quadratic optimization, quadratic programming, minimal residual, preconditioning, Krylov space, Arnoldi orthogonalization, symmetric successive overrelaxation, Gauss–Seidel

AMS subject classifications. 90C20, 65F10, 65Y20

PII. S1052623499356071

1. Introduction.

In this paper we consider the following problem of minimizing a quadratic over a sphere:

$$\text{minimize } \mathbf{x}^{\mathsf{T}}\mathbf{A}\mathbf{x} - 2\mathbf{b}^{\mathsf{T}}\mathbf{x} \quad \text{subject to } \|\mathbf{x}\| \le r, \tag{1.1}$$

where $\mathbf{A}$ is a symmetric $n \times n$ matrix, $\mathbf{b} \in \mathbf{R}^n$, T denotes transpose, and $\|\cdot\|$ is the Euclidean norm. This minimization problem is often called the trust region subproblem since it must be solved in each step of a trust region algorithm [1, 2, 3, 15, 19]. Problems of this form arise in many other applications including regularization methods for ill-posed problems [14, 26] and graph partitioning problems [10].

Although the solution to (1.1) can be expressed in terms of a diagonalization of $\mathbf{A}$, this representation is practical only when n is small. In this paper, we focus on the large-scale case. One approach to the large-scale case, developed by Golub and von Matt in [5] (also see [4]), is to (partially) tridiagonalize $\mathbf{A}$ using the Lanczos process and then solve tridiagonal problems to obtain an approximate solution to (1.1). For further developments of this approach, including preconditioning and a Fortran 90 implementation HSL_VF05 in the Harwell subroutine library, see Gould et al. [7]. For the method developed in this paper, we use an approach in the spirit of the Golub/von Matt/Gould et al. scheme to obtain a starting guess.

Parametric eigenvalue approaches to the sphere constrained problem (1.1) are developed by Sorensen [24] and by Rendl and Wolkowicz [20]. The relationship between these two approaches is discussed in detail in [20]. Roughly, Sorensen's approach involves constructing an approximation to the solution of (1.1) from the solution to a related eigenvalue problem. Since this approximation may not satisfy the bound on the norm of the solution, a series of eigenvalue problems are solved, and in the limit, the bound on the norm of the solution is fulfilled. In the approach of Rendl and Wolkowicz, the same eigenvalue problem is solved in each iteration; however, the bound on the norm of the solution is satisfied by maximizing a related dual function. The eigenvalue problems arising in either approach can be solved using Arnoldi

*Received by the editors May 10, 1999; accepted for publication (in revised form) November 7, 2000; published electronically July 2, 2001. This work was supported by the National Science Foundation.

http://www.siam.org/journals/siopt/12-1/35607.html

†Department of Mathematics, University of Florida, Gainesville, FL 32611 (hager@math.ufl.edu, http://www.math.ufl.edu/∼hager).

techniques such as those developed in [13]. In the "hard case" (see [16]), where $\mathbf{b}$ is orthogonal to the eigenvectors associated with the smallest eigenvalue of $\mathbf{A}$, Sorensen's approach needs to be modified. An efficient algorithm for the hard case is developed by Rojas in her thesis [21]. She also uses this algorithm to solve some difficult ill-posed problems of Hansen [11, 12]. The approach of Rendl and Wolkowicz does not need modification in the hard case; however, the convergence of algorithms for the eigenvalue problem may be slower when the computed eigenvalue is not simple.

The approach in this paper, which we call the sequential subspace method (SSM), involves solving (1.1) with the additional constraint that $\mathbf{x}$ is contained in a subspace. We show that convergence is locally quadratic (locally cubic when $\mathbf{b} = \mathbf{0}$) if the subspace contains the iterate generated by one step of the sequential quadratic programming (SQP) algorithm applied to (1.1). The convergence is quadratic even when the original problem is degenerate with multiple solutions and with a singular Jacobian for the first-order optimality system. Descent of the cost at a nonoptimal point can be ensured by including in the subspace either the cost gradient or an eigenvector associated with the smallest eigenvalue of $\mathbf{A}$. We observe in numerical experiments that appropriate small dimensional subspaces are generated by preconditioned Krylov space and minimum residual techniques. Comparisons with the algorithms of Sorensen [24], Rendl and Wolkowicz [20], and Gould et al. [7] are given in section 5.

A solution of the problem

$$\text{minimize } \mathbf{x}^{\mathsf{T}}\mathbf{A}\mathbf{x} \quad \text{subject to } \|\mathbf{x}\| = r \tag{1.2}$$

is any eigenvector associated with the smallest eigenvalue of $\mathbf{A}$. In comparing the SSM approach to algorithms for solving the eigenproblem, it follows from the discussion of Sleijpen and Van der Vorst in [22] that an SQP iterate for (1.2) is closely connected to the Rayleigh quotient iteration [18, p. 70], which is cubically convergent [18, p. 73]. In [22] approximate solutions to the SQP system are used to build up subspaces containing the approximation to the eigenvector. In this paper, we solve the SQP system relatively precisely, and we form a small dimensional subspace containing the SQP iterate. After computing the new approximation in the subspace, the previous information is discarded; hence, the computer memory requirements are relatively small.

2. Complete diagonalization. If there exists a solution $\mathbf{y}$ of (1.1) with $\|\mathbf{y}\| < r$, then $\mathbf{A}$ is positive semidefinite and $\mathbf{y}$ is the global minimizer of the quadratic $\mathbf{x}^{\mathsf{T}}\mathbf{A}\mathbf{x} - 2\mathbf{b}^{\mathsf{T}}\mathbf{x}$. Thus, when a minimizer of (1.1) lies in the interior of the constraining sphere, the constraint can be ignored and the optimization problem can be approached using techniques for unconstrained optimization. Consequently, we restrict our attention to the following equality constrained problem:

$$\text{minimize } \mathbf{x}^{\mathsf{T}}\mathbf{A}\mathbf{x} - 2\mathbf{b}^{\mathsf{T}}\mathbf{x} \quad \text{subject to } \|\mathbf{x}\| = r. \tag{2.1}$$

The solutions to (2.1) are characterized by the following result (see [23, Lemmas 2.4 and 2.8]).

LEMMA 2.1. *The vector* $\mathbf{x}$ *is a solution of* (2.1) *if and only if* $\|\mathbf{x}\| = r$ *and there exists* μ *such that* $\mathbf{A} + \mu\mathbf{I}$ *is positive semidefinite and* $(\mathbf{A} + \mu\mathbf{I})\mathbf{x} = \mathbf{b}$.

The solution to (2.1) can be expressed in terms of the eigenpairs of $\mathbf{A}$. Let $\mathbf{A} = \mathbf{\Phi}\mathbf{\Lambda}\mathbf{\Phi}^{\mathsf{T}}$ be a diagonalization of $\mathbf{A}$, where $\mathbf{\Lambda}$ is a diagonal matrix with diagonal elements $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$ and $\mathbf{\Phi}$ is the matrix whose columns $\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_n$ are orthonormal eigenvectors of $\mathbf{A}$. Defining $\beta_i = \mathbf{b}^{\mathsf{T}}\boldsymbol{\phi}_i$, $\mathcal{E}_1 = \{i : \lambda_i = \lambda_1\}$, and $\mathcal{E}_+ = \{i : \lambda_i > \lambda_1\}$, Lemma 2.1 yields the following.

LEMMA 2.2. *The vector* $\boldsymbol{\phi} = \sum_{i=1}^{n} c_i \boldsymbol{\phi}_i$ *is a solution of* (2.1) *if and only if* $\mathbf{c}$ *is chosen in the following way:*

(a) Degenerate case: *If* $\beta_i = 0$ *for all* $i \in \mathcal{E}_1$ *and*

$$\sum_{i \in \mathcal{E}_+} \frac{\beta_i^2}{(\lambda_i - \lambda_1)^2} \le r^2, \tag{2.2}$$

then $\mu = -\lambda_1$ *in Lemma* 2.1 *and* $c_i = \beta_i/(\lambda_i - \lambda_1)$ *for* $i \in \mathcal{E}_+$*; the* c_i *for* $i \in \mathcal{E}_1$ *are arbitrary scalars satisfying the condition*

$$\sum_{i \in \mathcal{E}_1} c_i^2 = r^2 - \sum_{i \in \mathcal{E}_+} \frac{\beta_i^2}{(\lambda_i - \lambda_1)^2}.$$

(b) Nondegenerate case: *If* (a) *does not hold, then* $c_i = \beta_i/(\lambda_i + \mu)$, $1 \le i \le n$, *where* $\mu > -\lambda_1$ *is chosen so that*

$$\sum_{i=1}^{n} \frac{\beta_i^2}{(\lambda_i + \mu)^2} = r^2. \tag{2.3}$$

Proof. Simply check that the sufficient optimality conditions of Lemma 2.1 are satisfied. The degenerate case, where the Jacobian of the first-order optimality system may be singular, coincides with the "hard case" of Moré and Sorensen [16], where $\mathbf{b}$ is orthogonal to the eigenspace associated with the smallest eigenvalue of $\mathbf{A}$ and the multiplier μ is equal to $-\lambda_1$. In the nondegenerate case, the multiplier μ is chosen so that $\mathbf{A} + \mu\mathbf{I}$ is positive definite and the solution $\mathbf{x} = \mathbf{x}(\mu)$ to $(\mathbf{A} + \mu\mathbf{I})\mathbf{x} = \mathbf{b}$ satisfies the constraint $\mathbf{x}^{\mathsf{T}}\mathbf{x} = r^2$. □

In the nondegenerate case, (2.3) leads to upper and lower bounds for the multiplier μ. Since $\lambda_i + \mu \ge \lambda_1 + \mu > 0$, $1 \le i \le n$, we have

$$r^2 = \sum_{i=1}^{n} \frac{\beta_i^2}{(\lambda_i + \mu)^2} \le \sum_{i=1}^{n} \frac{\beta_i^2}{(\lambda_1 + \mu)^2} = \frac{\|\mathbf{b}\|^2}{(\lambda_1 + \mu)^2}.$$

Since $\lambda_1 + \mu > 0$, it follows that

$$\mu \le \frac{\|\mathbf{b}\|}{r} - \lambda_1 := \mu_u. \tag{2.4}$$

To obtain a lower bound, observe that

$$r^2 = \sum_{i=1}^{n} \frac{\beta_i^2}{(\lambda_i + \mu)^2} \ge \frac{1}{(\lambda_1 + \mu)^2} \sum_{i \in \mathcal{E}_1} \beta_i^2,$$

which yields the relation

$$\mu \ge -\lambda_1 + \frac{1}{r} \left(\sum_{i \in \mathcal{E}_1} \beta_i^2 \right)^{1/2} := \mu_l. \tag{2.5}$$

Utilizing the upper and lower bounds μ_u and μ_l and the strict convexity of the left side of (2.3) on the interval $(\mu_l, \mu_u]$, it is easy to devise efficient algorithms to compute a solution μ of (2.3).

3. Incomplete diagonalization; local convergence. At iteration k in the SSM for (2.1), we impose the additional constraint that $\mathbf{x}$ lies in a subspace $\mathcal{S}_k$ of $\mathbf{R}^n$. Hence, the new iterate $\mathbf{x}_{k+1}$ is a solution of the problem

$$\text{minimize } \mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x} - 2\mathbf{b}^\mathsf{T}\mathbf{x} \quad \text{subject to } \|\mathbf{x}\| = r, \quad \mathbf{x} \in \mathcal{S}_k. \tag{3.1}$$

We show that the convergence is locally quadratic, even when the original problem (2.1) is degenerate, if we include an SQP iterate associated with $\mathbf{x}_k$ in $\mathcal{S}_k$.

If $\mathbf{V}$ is an $n \times l$ matrix with orthonormal columns that span $\mathcal{S}_k$, then (3.1) is equivalent to the problem

$$\text{minimize } \mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x} - 2\mathbf{b}^\mathsf{T}\mathbf{x} \quad \text{subject to } \|\mathbf{x}\| = r, \quad \mathbf{x} = \mathbf{V}\mathbf{y}. \tag{3.2}$$

After substituting for $\mathbf{x}$, (3.2) reduces to the following problem in $\mathbf{R}^l$:

$$\text{minimize } \mathbf{y}^\mathsf{T}\mathbf{B}\mathbf{y} - 2\mathbf{c}^\mathsf{T}\mathbf{y} \quad \text{subject to } \|\mathbf{y}\| = r, \tag{3.3}$$

where $\mathbf{B} = \mathbf{V}^\mathsf{T}\mathbf{A}\mathbf{V}$ and $\mathbf{c} = \mathbf{V}^\mathsf{T}\mathbf{b}$. If l is small, then (3.3) can be solved by complete diagonalization as in section 2 or, if $\mathbf{B}$ has a sparse factorization, then (3.3) can be solved quickly using the Newton approach developed in [16].

In theory, a tridiagonal $\mathbf{B}$ is generated using the Lanczos process [6]. In particular, if $\mathbf{v}_1$ is a unit vector and $\mathbf{v}_i$ is the ith column of $\mathbf{V}$, then the Lanczos process can be expressed as follows.

ALGORITHM 1 (LANCZOS).
$u_0 = 0$
for $j = 1 : l - 1$
  $\mathbf{s} \leftarrow \mathbf{A}\mathbf{v}_j$
  $d_j \leftarrow \mathbf{s}^\mathsf{T}\mathbf{v}_j$
  $\mathbf{s} \leftarrow \mathbf{s} - d_j\mathbf{v}_j - u_{j-1}\mathbf{v}_{j-1}$
  $u_j \leftarrow \|\mathbf{s}\|$
  $\mathbf{v}_{j+1} \leftarrow \mathbf{s}/u_j$
end
END ALGORITHM 1

Here $\mathbf{d}$ is the diagonal and $\mathbf{u}$ is the superdiagonal of the tridiagonal matrix $\mathbf{B}$. If $u_j = 0$ for some j, then the Lanczos process is terminated and the column spaces of $\mathbf{V}$ and $\mathbf{AV}$ coincide.

It is well known that the columns of $\mathbf{V}$ generated by this process may deviate significantly from orthogonality due to the propagation of rounding errors. When this happens, (3.2) is no longer equivalent to (3.3). Nonetheless, Gould et al. observe in [7] that the solution to (3.3) often provides a good approximation to the solution of (3.2) despite the loss of orthogonality. The Lanczos process can be repaired, in order to restore orthogonality, by using a Householder process to generate the columns of $\mathbf{V}$. This process, however, requires products between a vector and each of the previously computed columns of $\mathbf{V}$. Thus, the overhead needed to maintain orthogonality grows as nl^2 in the number of flops and as nl in storage. This overhead can be significant when n or l is large. On the other hand, to compute a high accuracy solution, we need to maintain orthogonality in order to obtain an equivalent problem (3.3). This leads us to focus on approaches that involve subspaces where l is much smaller than n. In particular, for an implementation (Algorithm 4) of the SSM proposed later, l is either 4 or 5.

Since SQP techniques often converge rapidly, with a good starting guess, we always include the SQP approximation in the subspace $\mathcal{S}_k$. The SQP method is equivalent to Newton's method applied to the nonlinear system

$$(\mathbf{A}+\mu\mathbf{I})\mathbf{x}-\mathbf{b}=\mathbf{0}, \quad \frac{1}{2}\mathbf{x}^{\mathsf{T}}\mathbf{x}-\frac{1}{2}r^2=0. \tag{3.4}$$

If $\mathbf{x}_k$ is the current iterate, which we assume satisfies the constraint $\|\mathbf{x}\|=r$, and μ_k is the current approximation to the multiplier associated with the constraint, then the Newton iterate can be expressed in the following way: $\mathbf{x}_{\rm SQP}=\mathbf{z}+\mathbf{x}_k$ and $\mu_{\rm SQP}=\mu_k+\nu$, where $\mathbf{z}$ and ν are solutions of the linear system

$$(\mathbf{A}+\mu_k\mathbf{I})\mathbf{z}+\mathbf{x}_k\nu=\mathbf{b}-(\mathbf{A}+\mu_k\mathbf{I})\mathbf{x}_k, \tag{3.5}$$

$$\mathbf{x}_k^{\mathsf{T}}\mathbf{z}=0. \tag{3.6}$$

When the coefficient matrix in (3.5)–(3.6) is singular, we let $(\mathbf{z},\nu)$ be the minimum residual/minimum norm solution; that is, $(\mathbf{z},\nu)$ is obtained (in theory) by multiplying the right side by the pseudoinverse of the coefficient matrix (see [8]).

A solution $\mathbf{x}_{k+1}$ to the subspace problem (3.1) is an approximation to the solution of (2.1). To obtain an estimate for the multiplier of Lemma 2.1, we minimize the Euclidean norm of the residual $\mathbf{b}-\mathbf{A}\mathbf{x}_{k+1}-\mu\mathbf{x}_{k+1}$ over the scalar μ. This works out to give

$$\mu_{k+1}=\rho(\mathbf{x}_{k+1}), \quad \text{where } \rho(\mathbf{x})=\frac{(\mathbf{b}-\mathbf{A}\mathbf{x})^{\mathsf{T}}\mathbf{x}}{\|\mathbf{x}\|^2}. \tag{3.7}$$

This is the standard least squares approximation to the solution of the overdetermined linear system $\mu\mathbf{x}_{k+1}=\mathbf{b}-\mathbf{A}\mathbf{x}_{k+1}$.

We now examine the local convergence of a solution $\mathbf{x}_{k+1}$ of (3.1) and the multiplier estimate (3.7) under the assumption that $\mathcal{S}_k$ contains $\mathbf{x}_{\rm SQP}=\mathbf{z}+\mathbf{x}_k$, where $(\mathbf{z},\nu)$ is a solution to (3.5). Let $\mathcal{S}^*$ denote the set of minimizers of (2.1), and let μ^* be the multiplier given by Lemma 2.1. In the nondegenerate setting, where $\mathbf{A}+\mu^*\mathbf{I}$ is positive definite, we show that the iteration is locally, quadratically convergent to the unique solution of (2.1). In the degenerate case $\mu^*=-\lambda_1$, where $\mathcal{S}^*$ has more than one element, we obtain local quadratic convergence to $\mathcal{S}^*$, where distance is measured in the usual way:

$$\operatorname{dist}(\mathbf{x},\mathcal{S}^*)=\inf\{\|\mathbf{x}-\mathbf{x}^*\| : \mathbf{x}^*\in\mathcal{S}^*\}.$$

In the nondegenerate-degenerate case, where $\mu^*=-\lambda_1$ but $\mathcal{S}^*$ contains a single element, we obtain local quadratic convergence for a "safe-guarded" choice of μ_k. Our convergence result in the special nondegenerate-degenerate case is given later in Lemma 3.4, while our local convergence result in either the nondegenerate case or the degenerate case with multiple solutions is as follows.

THEOREM 3.1. *Let μ^* be the multiplier of Lemma* 2.1 *associated with the set of solutions $\mathcal{S}^*$ of* (2.1)*, and suppose that either $\mathbf{A}+\mu^*\mathbf{I}$ is positive definite or $\mu^*=-\lambda_1$ with* (2.2) *a strict inequality. Then there exist positive constants η and C with the property that for any $(\mathbf{x}_k,\mu_k)$ such that*

$$|\mu_k-\mu^*|+\operatorname{dist}(\mathbf{x}_k,\mathcal{S}^*)\le\eta, \quad \|\mathbf{x}_k\|=r,$$

and for any subspace $\mathcal{S}_k$ that contains the SQP *iterate $\mathbf{x}_{\rm SQP}$ associated with* (3.5)–(3.6)*, any solution $\mathbf{x}_{k+1}$ of* (3.1) *and associated multiplier μ_{k+1} given by* (3.7) *satisfy*

the estimate

$$\operatorname{dist}(\mathbf{x}_{k+1}, \mathcal{S}^*) + |\mu_{k+1} - \mu_*| \le C\big(\operatorname{dist}(\mathbf{x}_k, \mathcal{S}^*)^2 + |\mu_k - \mu^*|^2\big).$$

The eigenvalue problem (1.2), corresponding to $\mathbf{b} = \mathbf{0}$, is always degenerate (with multiple solution) and the error has the special form

$$\operatorname{dist}(\mathbf{x}_{k+1}, \mathcal{S}^*) \le C|\mu_k + \lambda_1|\operatorname{dist}(\mathbf{x}_k, \mathcal{S}^*).$$

When the multiplier is estimated using (3.7), it can be shown, when $\mathbf{b} = \mathbf{0}$, that the error in the multiplier is bounded by a constant times the error in the solution vector squared (see the remark at the end of section 3.1). It follows that for some constant C,

$$\operatorname{dist}(\mathbf{x}_{k+1}, \mathcal{S}^*) \le C\operatorname{dist}(\mathbf{x}_k, \mathcal{S}^*)^3 \quad \text{and} \quad |\mu_{k+1} + \lambda_1| \le C\operatorname{dist}(\mathbf{x}_k, \mathcal{S}^*)^6, \tag{3.8}$$

which is the same as the convergence result for the Rayleigh quotient iteration.

3.1. Nondegenerate problems. We begin the derivation of Theorem 3.1 with the nondegenerate case.

LEMMA 3.2. *If* (2.1) *has a solution* $\mathbf{x}^*$ *and an associated multiplier* μ^* *with* $\mu^* > -\lambda_1$, *then there exist a neighborhood* $\mathcal{N}$ *of* $(\mathbf{x}^*, \mu^*)$ *and a constant* C *with the property that for any* $(\mathbf{x}_k, \mu_k) \in \mathcal{N}$ *with* $\|\mathbf{x}_k\| = r$, *and for any subspace* $\mathcal{S}_k$ *that contains the* SQP *iterate* $\mathbf{x}_{\mathrm{SQP}}$ *associated with* (3.5)–(3.6), *any solution* $\mathbf{x}_{k+1}$ *of* (3.1) *and associated multiplier* μ_{k+1} *given by* (3.7) *satisfy the estimate*

$$\|\mathbf{x}_{k+1} - \mathbf{x}^*\| + |\mu_{k+1} - \mu^*| \le C(\|\mathbf{x}_k - \mathbf{x}^*\|^2 + |\mu_k - \mu^*|^2).$$

Proof. Since $\mu^* > -\lambda_1$, the matrix $\mathbf{A} + \mu^*\mathbf{I}$ is positive definite, and the Jacobian of the nonlinear system (3.4) is nonsingular at $(\mathbf{x}^*, \mu^*)$. By the standard convergence theorem for Newton's method applied to a smooth system of equations, there exist a neighborhood $\mathcal{N}$ of $(\mathbf{x}_k, \mu_k)$ and a constant c such that

$$\|\mathbf{x}_{\mathrm{SQP}} - \mathbf{x}^*\| + |\mu_{\mathrm{SQP}} - \mu^*| \le c(\|\mathbf{x}_k - \mathbf{x}^*\|^2 + |\mu_k - \mu^*|^2)$$

whenever $(\mathbf{x}_k, \mu_k) \in \mathcal{N}$.

Let α and β be positive scalars chosen so that

$$\alpha\|\mathbf{x}\|^2 \le \mathbf{x}^{\mathsf{T}}(\mathbf{A} + \mu^*\mathbf{I})\mathbf{x} \le \beta\|\mathbf{x}\|^2 \tag{3.9}$$

for all $\mathbf{x} \in \mathbf{R}^n$, let f be the cost function in (2.1), $f(\mathbf{x}) = \mathbf{x}^{\mathsf{T}}\mathbf{A}\mathbf{x} - 2\mathbf{b}^{\mathsf{T}}\mathbf{x}$, and let $\mathcal{L}$ be the Lagrangian defined by

$$\mathcal{L}(\mathbf{x}, \mu) = f(\mathbf{x}) + \mu(\mathbf{x}^{\mathsf{T}}\mathbf{x} - r^2).$$

A Taylor expansion around $\mathbf{x}^*$ yields the relation

$$f(\mathbf{x}) = \mathcal{L}(\mathbf{x}, \mu^*) = f(\mathbf{x}^*) + (\mathbf{x} - \mathbf{x}^*)^{\mathsf{T}}(\mathbf{A} + \mu^*\mathbf{I})(\mathbf{x} - \mathbf{x}^*)$$

for any $\mathbf{x} \in \mathcal{B}_r = \{\mathbf{x} \in \mathbf{R}^n : \|\mathbf{x}\| = r\}$. Combining this with (3.9) gives

$$\alpha\|\mathbf{x} - \mathbf{x}^*\|^2 \le f(\mathbf{x}) - f(\mathbf{x}^*) \le \beta\|\mathbf{x} - \mathbf{x}^*\|^2 \tag{3.10}$$

for any $\mathbf{x} \in \mathcal{B}_r$.

If $\mathbf{p}$ is the projection of $\mathbf{x}_{\rm SQP}$ onto $\mathcal{B}_r$, then

$$\|\mathbf{x}_{\rm SQP} - \mathbf{p}\| \le \|\mathbf{x}_{\rm SQP} - \mathbf{x}^*\| \le c(\|\mathbf{x}_k - \mathbf{x}^*\|^2 + |\mu_k - \mu^*|^2). \tag{3.11}$$

Hence, we have

$$\|\mathbf{p} - \mathbf{x}^*\| \le \|\mathbf{p} - \mathbf{x}_{\rm SQP}\| + \|\mathbf{x}_{\rm SQP} - \mathbf{x}^*\| \le 2c(\|\mathbf{x}_k - \mathbf{x}^*\|^2 + |\mu_k - \mu^*|^2).$$

Since $\mathbf{p} = \gamma \mathbf{x}_{\rm SQP}$ for some γ, it follows that $\mathbf{p} \in \mathcal{S}_k$ and $f(\mathbf{x}_{k+1}) \le f(\mathbf{p})$. Combining this inequality with (3.10) and (3.11) gives

$$\begin{aligned}\alpha\|\mathbf{x}_{k+1} - \mathbf{x}^*\|^2 &\le f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*)\\ &\le f(\mathbf{p}) - f(\mathbf{x}^*)\\ &\le \beta\|\mathbf{p} - \mathbf{x}^*\|^2\\ &\le 4c^2\beta(\|\mathbf{x}_k - \mathbf{x}^*\|^2 + |\mu_k - \mu^*|^2)^2,\end{aligned}$$

which implies that

$$\|\mathbf{x}_{k+1} - \mathbf{x}^*\| \le 2c\sqrt{\beta/\alpha}(\|\mathbf{x}_k - \mathbf{x}^*\|^2 + |\mu_k - \mu^*|^2). \tag{3.12}$$

Since $\mathbf{b} = (\mathbf{A} + \mu^*\mathbf{I})\mathbf{x}^*$, we have, for any $\mathbf{x} \in \mathcal{B}_r$,

$$\begin{aligned}r^2\rho(\mathbf{x}) &= (\mathbf{b} - \mathbf{A}\mathbf{x})^{\mathsf T}\mathbf{x} = ((\mathbf{A} + \mu^*\mathbf{I})\mathbf{x}^* - \mathbf{A}\mathbf{x})^{\mathsf T}\mathbf{x}\\ &= \mathbf{x}^{\mathsf T}(\mathbf{A} + \mu^*\mathbf{I})(\mathbf{x}^* - \mathbf{x}) + \mu^* r^2.\end{aligned} \tag{3.13}$$

Making this substitution gives

$$|\mu_{k+1} - \mu^*| = |\rho(\mathbf{x}_{k+1}) - \mu^*| \le \frac{\lambda_n + \mu^*}{r}\|\mathbf{x}_{k+1} - \mathbf{x}^*\|. \tag{3.14}$$

Combining (3.14) with (3.12), the proof is complete. ∎

Remark. For the eigenvalue problem (1.2), we have $\mathbf{x}^* = r\boldsymbol{\phi}_1$, $\mu^* = -\lambda_1$, and $\boldsymbol{\phi}_1^{\mathsf T}(\mathbf{A} - \lambda_1\mathbf{I}) = \mathbf{0}$. In this case, (3.13) yields

$$r^2\rho(\mathbf{x}) = (\mathbf{x} - \mathbf{x}^*)^{\mathsf T}(\mathbf{A} - \lambda_1\mathbf{I})(\mathbf{x}^* - \mathbf{x}) - \lambda_1 r^2,$$

and (3.14) becomes

$$|\mu_{k+1} - \mu^*| \le \frac{\lambda_n - \lambda_1}{r^2}\|\mathbf{x}_{k+1} - \mathbf{x}^*\|^2.$$

3.2. Degenerate problems. Now consider local convergence in the degenerate case where $\mu^* = -\lambda_1$. Referring to Lemma 2.2, the degenerate case can happen only when $\beta_i = \mathbf{b}^{\mathsf T}\boldsymbol{\phi}_i = 0$ for all $i \in \mathcal{E}_1$. Any solution to (2.1) in the degenerate case can be expressed as $\mathbf{x}^* = \boldsymbol{\Phi}_1 + \boldsymbol{\Phi}_+$, where

$$\boldsymbol{\Phi}_+ = \sum_{i \in \mathcal{E}_+} c_i\boldsymbol{\phi}_i, \quad c_i = \beta_i/(\lambda_i - \lambda_1), \tag{3.15}$$

and $\boldsymbol{\Phi}_1$ is any linear combination of the vectors $\boldsymbol{\phi}_i$, $i \in \mathcal{E}_1$, satisfying the relation

$$\|\boldsymbol{\Phi}_1\|^2 + \|\boldsymbol{\Phi}_+\|^2 = r^2.$$

Initially, we suppose that $\|\mathbf{\Phi}_1\| = \delta > 0$, in which case the projection of $\mathcal{S}^*$ on the eigenspace associated with $\mathcal{E}_1$ contains a sphere of radius δ. Our convergence result is the following.

LEMMA 3.3. *Suppose that the multiplier μ^* of Lemma 2.1 associated with the set of solutions $\mathcal{S}^*$ of (2.1) is given by $\mu^* = -\lambda_1$ and that $\|\mathbf{\Phi}_1\| = \delta > 0$, where $\mathbf{\Phi}_1$ is the component of an element of $\mathcal{S}^*$ in the eigenspace associated with $\mathcal{E}_1$. Then there exist positive constants η and C with the property that for any $(\mathbf{x}_k, \mu_k)$ such that*

$$|\mu_k + \lambda_1| + \operatorname{dist}(\mathbf{x}_k, \mathcal{S}^*) \leq \eta, \quad \|\mathbf{x}_k\| = r,$$

and for any subspace $\mathcal{S}_k$ that contains the SQP *iterate $\mathbf{x}_{\mathrm{SQP}}$ associated with (3.5)–(3.6), any solution $\mathbf{x}_{k+1}$ of (3.1) and associated multiplier μ_{k+1} given by (3.7) satisfy the estimate*

$$\operatorname{dist}(\mathbf{x}_{k+1}, \mathcal{S}^*) + |\mu_{k+1} + \lambda_1| \leq C\big(\operatorname{dist}(\mathbf{x}_k, \mathcal{S}^*)^2 + |\mu_k + \lambda_1|^2\big).$$

Proof. Initially, let us assume that μ_k is near $-\lambda_1$, but $\mu_k \neq -\lambda_1$. In this case, the linear system (3.5)–(3.6) is nonsingular, and there exists a unique solution $(\mathbf{z}, \nu)$. We expand $\mathbf{z}$ and $\mathbf{x}_k$ in terms of the eigenvectors of $\mathbf{A}$ writing $\mathbf{z} = \sum_{i=1}^n \zeta_i \boldsymbol{\phi}_i$ and $\mathbf{x}_k = \sum_{i=1}^n \chi_i \boldsymbol{\phi}_i$. Utilizing (3.5), we obtain

$$\zeta_i = \frac{-\chi_i \nu}{\lambda_i + \mu_k} + \frac{\beta_i - (\lambda_i + \mu_k)\chi_i}{\lambda_i + \mu_k}. \tag{3.16}$$

Substituting this in (3.6) gives

$$\nu = \frac{\sum_{i=1}^n \chi_i(\beta_i - (\lambda_i + \mu_k)\chi_i)/(\lambda_i + \mu_k)}{\sum_{i=1}^n \chi_i^2/(\lambda_i + \mu_k)}. \tag{3.17}$$

Let us define $\mathbf{R} = \mathbf{b} - (\mathbf{A} + \mu_k \mathbf{I})\mathbf{x}_k$ and $\rho_i = \mathbf{R}^{\mathsf{T}} \boldsymbol{\phi}_i$. For $i \in \mathcal{E}_1$, $\beta_i = 0$ and

$$\nu = \frac{-(\lambda_1 + \mu_k)\left(\sum_{i \in \mathcal{E}_1} \chi_i^2 + \sum_{i \in \mathcal{E}_+} \frac{\chi_i \rho_i}{\lambda_i + \mu_k}\right)}{\sum_{i \in \mathcal{E}_1} \chi_i^2 + (\lambda_1 + \mu_k) \sum_{i \in \mathcal{E}_+} \frac{\chi_i^2}{\lambda_i + \mu_k}}. \tag{3.18}$$

If $\mathbf{x}^* \in \mathcal{S}^*$, then since $\mathbf{b} = (\mathbf{A} - \lambda_1 \mathbf{I})\mathbf{x}^*$ and $\|\mathbf{x}_k\| = r$, we have

$$\begin{aligned} \|\mathbf{R}\| = \|\mathbf{b} - (\mathbf{A} + \mu_k \mathbf{I})\mathbf{x}_k\| &\leq r|\lambda_1 + \mu_k| + \|\mathbf{A} - \lambda_1 \mathbf{I}\|\|\mathbf{x}_k - \mathbf{x}^*\| \\ &\leq \max\{r, \|\mathbf{A} - \lambda_1 \mathbf{I}\|\}(|\lambda_1 + \mu_k| + \|\mathbf{x}_k - \mathbf{x}^*\|). \end{aligned} \tag{3.19}$$

Let ϵ_k be the error at step k defined by

$$\epsilon_k = |\lambda_1 + \mu_k| + \operatorname{dist}(\mathbf{x}_k, \mathcal{S}^*).$$

By (3.19), we have $\|\mathbf{R}\| = O(\epsilon_k)$, while (3.18) gives

$$\nu = -(\lambda_1 + \mu_k)(1 + O(\epsilon_k)) \tag{3.20}$$

$$= -(\lambda_1 + \mu_k) + O(\epsilon_k^2) \tag{3.21}$$

since $\sum_{i \in \mathcal{E}_1} \chi_i^2$ is near $\delta^2 > 0$ when $\mathbf{x}_k$ is near $\mathcal{S}^*$. From (3.16), we have

$$\zeta_i + \chi_i = \frac{\beta_i - \chi_i \nu}{\lambda_i + \mu_k}. \tag{3.22}$$

Since $\beta_i = 0$ and $\lambda_i = \lambda_1$ for $i \in \mathcal{E}_1$, (3.20) and (3.22) give

$$\zeta_i + \chi_i = \chi_i + O(\epsilon_k) \quad \text{for } i \in \mathcal{E}_1. \tag{3.23}$$

Let $\mathbf{x}^*$ be the closest element of $\mathcal{S}^*$ to $\mathbf{x}_k$ and define $\chi_i^* = \boldsymbol{\phi}^{\mathsf{T}}\mathbf{x}^*$. Then we have

$$|\chi_i - \chi_i^*| = |(\mathbf{x}_k - \mathbf{x}^*)^{\mathsf{T}}\boldsymbol{\phi}_i| \le \|\mathbf{x}_k - \mathbf{x}^*\| \le \epsilon_k. \tag{3.24}$$

By (3.23) the $\boldsymbol{\phi}_i$ component of $\mathbf{x}_{\text{SQP}} = \mathbf{z} + \mathbf{x}_k$ for $i \in \mathcal{E}_1$ is in error by $O(\epsilon_k)$ since χ_i, the $\boldsymbol{\phi}_i$ component of $\mathbf{x}_k$, is in error by $O(\epsilon_k)$ by (3.24).

Lemma 2.2 implies that $\beta_i = \chi_i^*(\lambda_i - \lambda_1)$ for $i \in \mathcal{E}_+$. Combining this with (3.21) and (3.22) gives

$$\begin{aligned}
\zeta_i + \chi_i &= \frac{\beta_i - \chi_i\nu}{\lambda_i + \mu_k} = \frac{\beta_i + \chi_i(\lambda_1 + \mu_k)}{\lambda_i + \mu_k} + O(\epsilon_k^2) \\
&= \frac{\chi_i^*(\lambda_i - \lambda_1) + \chi_i(\lambda_1 + \mu_k)}{\lambda_i + \mu_k} + O(\epsilon_k^2) \\
&= \frac{\chi_i^*(\lambda_i - \lambda_1) + \chi_i^*(\lambda_1 + \mu_k)}{\lambda_i + \mu_k} + O(\epsilon_k^2) = \chi_i^* + O(\epsilon_k^2). \qquad (3.25)
\end{aligned}$$

Hence, for $i \in \mathcal{E}_+$ the $\boldsymbol{\phi}_i$ component of $\mathbf{x}_{\text{SQP}}$ is in error by $O(\epsilon_k^2)$.

Let $\|\cdot\|_+$ be the seminorm associated with projection into the eigenspace associated with $\mathcal{E}_+$:

$$\|\mathbf{x}\|_+^2 = \sum_{i \in \mathcal{E}_+} (\mathbf{x}^{\mathsf{T}}\boldsymbol{\phi}_i)^2.$$

Then we have

$$(\lambda_+ - \lambda_1)\|\mathbf{x}\|_+^2 \le \mathbf{x}^{\mathsf{T}}(\mathbf{A} - \lambda_1\mathbf{I})\mathbf{x} \le (\lambda_n - \lambda_1)\|\mathbf{x}\|_+^2 \tag{3.26}$$

for all $\mathbf{x} \in \mathbf{R}^n$, where $\lambda_+ = \min\{\lambda_i : \lambda_i > \lambda_1, 1 \le i \le n\}$. Proceeding as we did earlier, but replacing norms with seminorms,

$$\begin{aligned}
\alpha\|\mathbf{x}_{k+1} - \mathbf{x}^*\|_+^2 &\le f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*) \\
&\le f(\mathbf{p}) - f(\mathbf{x}^*) \\
&\le \beta\|\mathbf{p} - \mathbf{x}^*\|_+^2, \qquad (3.27)
\end{aligned}$$

where $\mathbf{p}$ is the projection of $\mathbf{x}_{\text{SQP}}$ onto the ball $\mathcal{B}_r$, and $\mathbf{p} = \gamma\mathbf{x}_{\text{SQP}}$ for some $\gamma \ge 0$. Since $\|\mathbf{z}\| = O(\epsilon_k)$ by (3.23) and (3.25), and $\mathbf{z}$ is perpendicular to $\mathbf{x}_k$ by (3.6), we have

$$\|\mathbf{x}_{\text{SQP}}\|^2 = \|\mathbf{z} + \mathbf{x}_k\|^2 = \|\mathbf{z}\|^2 + \|\mathbf{x}_k\|^2 = r^2 + O(\epsilon_k^2).$$

This implies that $\|\mathbf{x}_{\text{SQP}}\| = r + O(\epsilon_k^2)$, and $\gamma = 1 + O(\epsilon_k^2)$. For $i \in \mathcal{E}_+$,

$$\mathbf{p}^{\mathsf{T}}\boldsymbol{\phi}_i = \gamma\mathbf{x}_{\text{SQP}}^{\mathsf{T}}\boldsymbol{\phi}_i = (1 + O(\epsilon_k^2))(\zeta_i + \chi_i) = (1 + O(\epsilon_k^2))(\chi_i^* + O(\epsilon_k^2)) = \chi_i^* + O(\epsilon_k^2).$$

Consequently, $\|\mathbf{p} - \mathbf{x}^*\|_+ = O(\epsilon_k^2)$, which combines with (3.27) to give

$$\|\mathbf{x}_{k+1} - \mathbf{x}^*\|_+ = O(\epsilon_k^2). \tag{3.28}$$

By the triangle inequality,

$$\|\mathbf{x}^*\|_+ - O(\epsilon_k^2) \le \|\mathbf{x}_{k+1}\|_+ \le \|\mathbf{x}^*\|_+ + O(\epsilon_k^2).$$

Let $\|\cdot\|_1$ be the seminorm defined by

$$\|\mathbf{x}\|_1^2 = \sum_{i\in\mathcal{E}_1} (\mathbf{x}^\mathsf{T}\boldsymbol{\phi}_i)^2, \tag{3.29}$$

and recall that $\|\mathbf{x}^*\|_1 = \delta$ for any $\mathbf{x}^* \in \mathcal{S}^*$. By the Pythagorean theorem and the fact that $\mathbf{x}_{k+1}$ has length r, we have

$$\|\mathbf{x}_{k+1}\|_1^2 = r^2 - \|\mathbf{x}_{k+1}\|_+^2 = r^2 - \|\mathbf{x}^*\|_+^2 + O(\epsilon_k^2) = \|\mathbf{x}^*\|_1^2 + O(\epsilon_k^2) = \delta^2 + O(\epsilon_k^2),$$

which implies that

$$\|\mathbf{x}_{k+1}\|_1 = \delta + O(\epsilon_k^2). \tag{3.30}$$

The distance from $\mathbf{x}_{k+1}$ to $\mathcal{S}^*$ is given by

$$\operatorname{dist}(\mathbf{x}_{k+1}, \mathcal{S}^*)^2 = \|\mathbf{x}_{k+1} - \mathbf{x}^*\|_+^2 + (\delta - \|\mathbf{x}_{k+1}\|_1)^2, \tag{3.31}$$

where $\mathbf{x}^*$ is any element of $\mathcal{S}^*$. Relations (3.28)–(3.31) yield $\operatorname{dist}(\mathbf{x}_{k+1}, \mathcal{S}^*) = O(\epsilon_k^2)$, while (3.14) gives $|\mu_{k+1} - \mu^*| = O(\epsilon_k^2)$. Combining these estimates, we have $\epsilon_{k+1} = O(\epsilon_k^2)$.

This analysis was given under the assumption that $\mu_k \neq -\lambda_1$. In the special case $\mu_k = -\lambda_1$, we now show how the analysis should be modified. With the change of variables $\mathbf{z} = \sum_{i=1}^n \zeta_i \boldsymbol{\phi}_i$ and the substitution $\mathbf{x}_k = \sum_{i=1}^n \chi_i \boldsymbol{\phi}_i$, the SQP system (3.5)–(3.6) is equivalent, by orthogonal transformation, to

$$\left[\begin{array}{c|c} \mathbf{D} & \boldsymbol{\chi} \\ \boldsymbol{\chi}^\mathsf{T} & \mathbf{0} \end{array}\right] \left[\begin{array}{c} \boldsymbol{\zeta} \\ \nu \end{array}\right] = \left[\begin{array}{c} \boldsymbol{\beta} - \mathbf{D}\boldsymbol{\chi} \\ 0 \end{array}\right], \tag{3.32}$$

where $\mathbf{D}$ is a diagonal matrix with diagonal elements $d_{ii} = \lambda_i - \lambda_1$. If $\mathcal{E}_1$ has s elements, then the first s diagonal elements of $\mathbf{D}$ and the first s components of $\boldsymbol{\beta} - \mathbf{D}\boldsymbol{\chi}$ vanish. Hence, the first s equations in (3.32) imply that $\nu = 0$. The next $n - s$ equations give

$$\zeta_i = -\chi_i + \beta_i/(\lambda_i - \lambda_1) = -\chi_i + \chi_i^*, \quad i \in \mathcal{E}_+, \tag{3.33}$$

while the last equation in (3.32) gives

$$\sum_{i\in\mathcal{E}_1} \chi_i\zeta_i = -\sum_{i\in\mathcal{E}_+} \chi_i\zeta_i.$$

The minimum norm solution to this last equation is

$$\zeta_i = -\left(\frac{\sum_{i\in\mathcal{E}_+} \zeta_i\chi_i}{\sum_{i\in\mathcal{E}_1} \chi_i^2}\right)\chi_i \quad \text{for } i \in \mathcal{E}_1. \tag{3.34}$$

By (3.33), $\zeta_i + \chi_i = \chi_i^*$ and $|\zeta_i| \le \epsilon_k$ for $i \in \mathcal{E}_+$. By (3.34), $|\zeta_i| = O(\epsilon_k)$ for $i \in \mathcal{E}_1$. Combining these bounds, we have $\|\mathbf{z}\| = O(\epsilon_k)$. With these relations, all the analysis from (3.26) onward can be applied, leading us to the estimate $\epsilon_{k+1} = O(\epsilon_k^2)$. ∎

Lemmas 3.2 and 3.3 yield Theorem 3.1.

3.3. Nondegenerate-degenerate problems. Finally, let us consider the nondegenerate-degenerate case, where $\mu^* = -\lambda_1$, $\mathbf{x}^* = \mathbf{\Phi}_1 + \mathbf{\Phi}_+$, and the $\mathbf{\Phi}_1$ component of $\mathbf{x}^*$ in the eigenspace associated with the smallest eigenvalue of $\mathbf{A}$ vanishes. Our convergence result is the following.

LEMMA 3.4. *If* (2.1) *has a solution* $\mathbf{x}^* = \mathbf{\Phi}_+$, *where* $\mathbf{\Phi}_+$ *is given by* (3.15), *then there exist a neighborhood* $\mathcal{N}$ *of* $(\mathbf{x}^*, -\lambda_1)$ *and a constant* C *with the property that for any* $(\mathbf{x}_k, \mu_k) \in \mathcal{N}$ *with*

$$\mu_k \geq -\lambda_1 + \|\mathbf{b} - (\mathbf{A} + \rho(\mathbf{x}_k)\mathbf{I})\mathbf{x}_k\|, \quad \|\mathbf{x}_k\| = r, \tag{3.35}$$

and for any subspace $\mathcal{S}_k$ *that contains the* SQP *iterate* $\mathbf{x}_{\rm SQP}$ *associated with* (3.5)–(3.6), *the solution* $\mathbf{x}_{k+1}$ *of* (3.1) *and associated multiplier* μ_{k+1} *given by* (3.7) *satisfy the estimate*

$$\|\mathbf{x}_{k+1} - \mathbf{x}^*\| + |\mu_{k+1} - \mu^*| \leq C(\|\mathbf{x}_k - \mathbf{x}^*\|^2 + |\mu_k - \mu^*|^2).$$

In the case that $\mu_k = -\lambda_1 + \|\mathbf{b} - (\mathbf{A} + \rho(\mathbf{x}_k)\mathbf{I})\mathbf{x}_k\|$, C *can be chosen so that*

$$\|\mathbf{x}_{k+1} - \mathbf{x}^*\| + |\mu_{k+1} - \mu^*| \leq C\|\mathbf{x}_k - \mathbf{x}^*\|^2.$$

Proof. Focusing on the numerator in (3.17), and substituting $\beta_i = (\lambda_i - \lambda_1)\chi_i^*$, we have

$$\begin{aligned}
&\sum_{i=1}^n \frac{\chi_i(\beta_i - (\lambda_i + \mu_k)\chi_i)}{\lambda_i + \mu_k} \\
&= \sum_{i=1}^n \frac{\chi_i\big((\lambda_i - \lambda_1)(\chi_i^* - \chi_i + \chi_i) - (\lambda_i + \mu_k)\chi_i\big)}{\lambda_i + \mu_k} \\
&= -(\lambda_1 + \mu_k)\sum_{i=1}^n \frac{\chi_i^2}{\lambda_i + \mu_k} + \sum_{i\in\mathcal{E}_+} \frac{\chi_i(\lambda_i - \lambda_1)(\chi_i^* - \chi_i)}{\lambda_i + \mu_k} \\
&= -(\lambda_1 + \mu_k)\sum_{i=1}^n \frac{\chi_i^2}{\lambda_i + \mu_k} + \sum_{i\in\mathcal{E}_+} \chi_i(\chi_i^* - \chi_i) - (\lambda_1 + \mu_k)\sum_{i\in\mathcal{E}_+} \frac{\chi_i(\chi_i^* - \chi_i)}{\lambda_i + \mu_k}.
\end{aligned}$$

With this substitution for the numerator of ν in (3.17), we obtain

$$\nu = -(\lambda_1 + \mu_k) + \frac{\sum_{i\in\mathcal{E}_+} \chi_i(\chi_i^* - \chi_i)}{\sum_{i=1}^n \chi_i^2/(\lambda_i + \mu_k)} - \frac{(\lambda_1 + \mu_k)\sum_{i\in\mathcal{E}_+} \frac{\chi_i(\chi_i^* - \chi_i)}{\lambda_i + \mu_k}}{\sum_{i=1}^n \chi_i^2/(\lambda_i + \mu_k)}. \tag{3.36}$$

Since $\mu_k > -\lambda_1$, the denominator terms in (3.36) have the lower bound

$$\sum_{i=1}^n \frac{\chi_i^2}{\lambda_i + \mu_k} \geq \sum_{i=1}^n \frac{\chi_i^2}{\lambda_n + \mu_k} = \frac{r^2}{\lambda_n + \mu_k}. \tag{3.37}$$

Another lower bound is gotten by neglecting terms corresponding to indices $i \in \mathcal{E}_+$:

$$\sum_{i=1}^n \frac{\chi_i^2}{\lambda_i + \mu_k} \geq \sum_{i\in\mathcal{E}_1} \frac{\chi_i^2}{\lambda_i + \mu_k} = \frac{\|\mathbf{x}_k\|_1^2}{\lambda_1 + \mu_k}, \tag{3.38}$$

where the seminorm $\|\cdot\|_1$ is defined in (3.29). Combining (3.36)–(3.38) yields

$$\nu = -(\lambda_1 + \mu_k)(1 + O(\|\mathbf{x}_k - \mathbf{x}^*\|_+)) + \frac{O(\|\mathbf{x}^* - \mathbf{x}_k\|_+)}{\max\{1, \|\mathbf{x}_k\|_1^2/(\lambda_1 + \mu_k)\}}. \tag{3.39}$$

Returning to our previous analysis of the degenerate case, it follows from (3.22) and (3.39) that for $i \in \mathcal{E}_1$, we have

$$\begin{aligned} \zeta_i + \chi_i &= \frac{\beta_i - \chi_i \nu}{\lambda_i + \mu_k} = \frac{-\chi_i \nu}{\lambda_1 + \mu_k} \\ &= \chi_i + O(\epsilon_k) \left(1 + \frac{\|\mathbf{x}_k - \mathbf{x}^*\|_+}{\max\{\lambda_1 + \mu_k, \|\mathbf{x}_k\|_1^2\}} \right). \end{aligned} \tag{3.40}$$

Here we exploit the fact that for $i \in \mathcal{E}_1$, $|\chi_i| \leq \|\mathbf{x}_k\|_1 \leq \epsilon_k$. In order to analyze (3.40), we consider two separate cases: (i) $\|\mathbf{x}_k\|_1^2 \geq \sigma \|\mathbf{x}_k - \mathbf{x}^*\|_+$ and (ii) $\|\mathbf{x}_k\|_1^2 < \sigma \|\mathbf{x}_k - \mathbf{x}^*\|_+$, where σ is any fixed constant satisfying

$$0 < \sigma < \frac{r(\lambda_+ - \lambda_1)}{\lambda_n - \lambda_1}, \quad \lambda_+ = \min\{\lambda_i : \lambda_i > \lambda_1,\ 1 \leq i \leq n\}. \tag{3.41}$$

In case (i),

$$\frac{\|\mathbf{x}_k - \mathbf{x}^*\|_+}{\max\{\lambda_1 + \mu_k, \|\mathbf{x}_k\|_1^2\}} \leq \frac{\|\mathbf{x}_k - \mathbf{x}^*\|_+}{\max\{\lambda_1 + \mu_k, \sigma \|\mathbf{x}_k - \mathbf{x}^*\|_+\}} \leq \frac{1}{\sigma}. \tag{3.42}$$

We now derive a similar bound for the left side of (3.42) in case (ii). In this case, it follows from (3.35) that

$$\frac{\|\mathbf{x}_k - \mathbf{x}^*\|_+}{\max\{\lambda_1 + \mu_k, \|\mathbf{x}_k\|_1^2\}} \leq \frac{\|\mathbf{x}^* - \mathbf{x}_k\|_+}{\|\mathbf{b} - (\mathbf{A} + \rho(\mathbf{x}_k)\mathbf{I})\mathbf{x}_k\|}.$$

Since $\mathbf{b} = (\mathbf{A} - \lambda_1 \mathbf{I})\mathbf{x}^*$, we have

$$\begin{aligned} \mathbf{b} - (\mathbf{A} + \rho(\mathbf{x})\mathbf{I})\mathbf{x} &= (\mathbf{A} - \lambda_1 \mathbf{I})\mathbf{x}^* - (\mathbf{A} + \rho(\mathbf{x})\mathbf{I})\mathbf{x} \\ &= (\mathbf{A} - \lambda_1 \mathbf{I})(\mathbf{x}^* - \mathbf{x}) - (\rho(\mathbf{x}) + \lambda_1)\mathbf{x} \\ &= (\mathbf{A} - \lambda_1 \mathbf{I})(\mathbf{x}^* - \mathbf{x})_+ - (\rho(\mathbf{x}) + \lambda_1)\mathbf{x} \end{aligned}$$

for any $\mathbf{x} \in \mathbf{R}^n$, where a + subscript on a vector is used to denote its projection on the eigenspace associated with $\mathcal{E}_+$. After substituting for ρ using (3.13), we obtain

$$\mathbf{b} - (\mathbf{A} + \rho(\mathbf{x})\mathbf{I})\mathbf{x} = (\mathbf{A} - \lambda_1 \mathbf{I})(\mathbf{x}^* - \mathbf{x}) - r^{-2}\big(\mathbf{x}^{\mathsf{T}}(\mathbf{A} - \lambda_1 \mathbf{I})(\mathbf{x}^* - \mathbf{x})_+\big)\mathbf{x} \tag{3.43}$$

for any $\mathbf{x} \in \mathcal{B}_r$. Assuming $\mathbf{x}_k \neq \mathbf{x}^*$, it follows that

$$\frac{\|\mathbf{x}^* - \mathbf{x}_k\|_+}{\|\mathbf{b} - (\mathbf{A} + \rho(\mathbf{x}_k)\mathbf{I})\mathbf{x}_k\|} = \frac{1}{\|(\mathbf{A} - \lambda_1 \mathbf{I})\mathbf{y} - r^{-2}\big(\mathbf{x}_k^{\mathsf{T}}(\mathbf{A} - \lambda_1 \mathbf{I})\mathbf{y}\big)\mathbf{x}_k\|}, \tag{3.44}$$

where $\mathbf{y} = (\mathbf{x}^* - \mathbf{x}_k)_+ / \|\mathbf{x}^* - \mathbf{x}_k\|_+$ is a unit vector (note that when $\|\mathbf{x}_k\| = r$, $\|\mathbf{x}^* - \mathbf{x}_k\|_+ = 0$ if and only if $\mathbf{x}_k = \mathbf{x}^*$ since $\|\mathbf{x}^*\|_1 = 0$).

We will establish a uniform bound for the expression (3.44) when $\mathbf{x}_k$ is near $\mathbf{x}^*$, $\|\mathbf{x}_k\|_1^2 \leq \sigma \|\mathbf{x}_k - \mathbf{x}^*\|_+$, and $\|\mathbf{x}_k\| = r$. To facilitate this analysis, we first consider whether the equation

$$(\mathbf{A} - \lambda_1 \mathbf{I})\mathbf{y} = r^{-2}\big(\mathbf{y}^{\mathsf{T}}(\mathbf{A} - \lambda_1 \mathbf{I})\mathbf{x}^*\big)\mathbf{x}^* \tag{3.45}$$

has a solution of the form $\mathbf{y} = (\mathbf{x}^* - \mathbf{x})_+ / \|\mathbf{x}^* - \mathbf{x}\|_+$ with $\mathbf{x}$ near $\mathbf{x}^*$, $\|\mathbf{x}\| = r$, and $\|\mathbf{x}\|_1^2 \leq \sigma \|\mathbf{x} - \mathbf{x}^*\|_+$. Since $\|\mathbf{y}\| = 1$ for $\mathbf{y}$ of this form, the Schwarz inequality gives

$$|\mathbf{y}^{\mathsf{T}}(\mathbf{A} - \lambda_1 \mathbf{I})\mathbf{x}^*| \leq (\lambda_n - \lambda_1)\|\mathbf{y}\|\|\mathbf{x}^*\| = r(\lambda_n - \lambda_1). \tag{3.46}$$

Since the unit vector $\mathbf{y}$ is orthogonal to the eigenspace associated with λ_1,

$$\mathbf{y}^{\mathsf{T}}(\mathbf{A}-\lambda_1\mathbf{I})\mathbf{y} \geq \lambda_+ - \lambda_1. \tag{3.47}$$

Multiplying (3.45) by $\mathbf{y}^{\mathsf{T}}$ and using both (3.46) and (3.47) gives

$$|\mathbf{y}^{\mathsf{T}}\mathbf{x}^*| \geq \frac{r(\lambda_+ - \lambda_1)}{\lambda_n - \lambda_1} > \sigma. \tag{3.48}$$

For any $\mathbf{x} \in \mathcal{B}_r$, we have

$$\begin{aligned} r^2 = \|\mathbf{x}\|^2 &= r^2 + 2(\mathbf{x}-\mathbf{x}^*)^{\mathsf{T}}\mathbf{x}^* + \|\mathbf{x}-\mathbf{x}^*\|^2 \\ &= r^2 - 2\|\mathbf{x}^*-\mathbf{x}\|_+\mathbf{y}^{\mathsf{T}}\mathbf{x}^* + \|\mathbf{x}-\mathbf{x}^*\|^2, \end{aligned}$$

which implies that

$$\begin{aligned} \mathbf{y}^{\mathsf{T}}\mathbf{x}^* &= \frac{\|\mathbf{x}-\mathbf{x}^*\|^2}{2\|\mathbf{x}-\mathbf{x}^*\|_+} = \frac{\|\mathbf{x}-\mathbf{x}^*\|_+^2 + \|\mathbf{x}-\mathbf{x}^*\|_1^2}{2\|\mathbf{x}-\mathbf{x}^*\|_+} \\ &= \frac{1}{2}\left(\|\mathbf{x}-\mathbf{x}^*\|_+ + \frac{\|\mathbf{x}\|_1^2}{\|\mathbf{x}-\mathbf{x}^*\|_+}\right) \end{aligned} \tag{3.49}$$

since $\|\mathbf{x}-\mathbf{x}^*\|_1 = \|\mathbf{x}\|_1$. If $\|\mathbf{x}\|_1^2 \leq \sigma\|\mathbf{x}-\mathbf{x}^*\|_+$, then (3.49) yields the relation

$$0 \leq \mathbf{y}^{\mathsf{T}}\mathbf{x}^* \leq \frac{1}{2}(\|\mathbf{x}-\mathbf{x}^*\|_+ + \sigma). \tag{3.50}$$

Referring to (3.48), we have a contradiction when $\|\mathbf{x}-\mathbf{x}^*\|_+ \leq \sigma$.

In summary, (3.45) has no solution over the set $\mathcal{Y}$ consisting of those $\mathbf{y}$ that satisfy the conditions $\mathbf{y} = (\mathbf{x}^*-\mathbf{x})_+/\|\mathbf{x}^*-\mathbf{x}\|_+$, $\mathbf{x} \neq \mathbf{x}^*$, $\|\mathbf{x}\|_1^2 \leq \sigma\|\mathbf{x}-\mathbf{x}^*\|_+$, $\|\mathbf{x}-\mathbf{x}^*\|_+ \leq \sigma$, and $\|\mathbf{x}\| = r$. If $\mathbf{y}$ lies in the closure of $\mathcal{Y}$, then by (3.50), $\mathbf{y}^{\mathsf{T}}\mathbf{x}^* \leq \sigma$; since any solution of (3.45) satisfies (3.48), $\mathbf{y}$ cannot be a solution of (3.45). Since (3.45) has no solution over the closure of $\mathcal{Y}$, the following constant δ is strictly positive:

$$\delta = \min_{\mathbf{y}\in\mathcal{Y}} \|(\mathbf{A}-\lambda_1\mathbf{I})\mathbf{y} - r^{-2}\big(\mathbf{y}^{\mathsf{T}}(\mathbf{A}-\lambda_1\mathbf{I})\mathbf{x}^*\big)\mathbf{x}^*\|.$$

Since

$$\lim_{\mathbf{x}_k\to\mathbf{x}^*} \min_{\mathbf{y}\in\mathcal{Y}} \|(\mathbf{A}-\lambda_1\mathbf{I})\mathbf{y} - r^{-2}\big(\mathbf{y}^{\mathsf{T}}(\mathbf{A}-\lambda_1\mathbf{I})\mathbf{x}_k\big)\mathbf{x}_k\| = \delta,$$

(3.44) is bounded uniformly over all $\mathbf{x}_k$ near $\mathbf{x}^*$ with $\|\mathbf{x}_k\| = r$ and $\|\mathbf{x}_k\|_1^2 < \sigma\|\mathbf{x}_k - \mathbf{x}^*\|_+$. Thus in either case (i) or (ii), the left side of (3.42) is bounded and, by (3.40), we have

$$\zeta_i + \chi_i = \chi_i + O(\epsilon_k) \quad \text{for } i \in \mathcal{E}_1,$$

which is the same as relation (3.23) in the degenerate case.

To establish the analogue of (3.25) for indices $i \in \mathcal{E}_+$, we need a different bound for the next to last term in (3.36). From the identity $\sum_{i=1}^n \chi_i^2 = \sum_{i=1}^n \chi_i^{*2} = r^2$, we obtain

$$\sum_{i=1}^n (\chi_i^* + \chi_i)(\chi_i^* - \chi_i) = 0. \tag{3.51}$$

Hence, we have

$$-\sum_{i=1}^{n}\chi_i(\chi_i^*-\chi_i) = -\sum_{i=1}^{n}\chi_i(\chi_i^*-\chi_i) + \frac{1}{2}(\chi_i^*+\chi_i)(\chi_i^*-\chi_i)$$

$$= \frac{1}{2}\sum_{i=1}^{n}(\chi_i^*-\chi_i)^2. \tag{3.52}$$

Since $\chi_i^* = 0$ for $i \in \mathcal{E}_1$, (3.52) implies that

$$-\sum_{i\in\mathcal{E}_+}\chi_i(\chi_i^*-\chi_i) = \frac{1}{2}\sum_{i\in\mathcal{E}_+}(\chi_i^*-\chi_i)^2 - \frac{1}{2}\sum_{i\in\mathcal{E}_1}\chi_i^2$$

$$= \frac{1}{2}\sum_{i\in\mathcal{E}_+}(\chi_i^*-\chi_i)^2 - \frac{1}{2}\sum_{i\in\mathcal{E}_1}(\chi_i^*-\chi_i)^2.$$

It follows that

$$\left|\sum_{i\in\mathcal{E}_+}\chi_i(\chi_i^*-\chi_i)\right| \le \|\mathbf{x}^*-\mathbf{x}_k\|^2.$$

This estimate, along with the lower bound (3.37) for the denominator in (3.36), yields the relation

$$\nu = -(\lambda_1+\mu_k) + O(\epsilon_k^2).$$

The remainder of the analysis is identical to that given for the degenerate case (Lemma 3.3), starting with (3.25). Since $\mathcal{S}^* = \{\mathbf{x}^*\}$, it follows from the analysis of Lemma 3.3 that

$$\|\mathbf{x}_{k+1}-\mathbf{x}^*\| + |\mu_{k+1}+\lambda_1| \le C(\|\mathbf{x}_k-\mathbf{x}^*\|^2 + |\mu_k+\lambda_1|^2). \tag{3.53}$$

In the special case $\mu_k = -\lambda_1 + \|\mathbf{b}-(\mathbf{A}+\rho(\mathbf{x}_k)\mathbf{I})\mathbf{x}_k\|$, (3.43) gives

$$|\mu_k+\lambda_1| = \|\mathbf{b}-(\mathbf{A}+\rho(\mathbf{x}_k)\mathbf{I})\mathbf{x}_k\| = O(\|\mathbf{x}_k-\mathbf{x}^*\|).$$

Hence, the $|\mu_k+\lambda_1|^2$ term in (3.53) can be absorbed in the $\|\mathbf{x}_k-\mathbf{x}^*\|^2$ term. This completes the proof. ∎

4. Implementation. In our experimentation with the SSM, we put the following four vectors in $\mathcal{S}_k$ in each iteration: $\mathbf{x}_{\mathrm{SQP}}$, $\mathbf{x}_k$, $\mathbf{b}-\mathbf{A}\mathbf{x}_k$, and an estimate for an eigenvector of $\mathbf{A}$ associated with the smallest eigenvalue. By including $\mathbf{x}_k$ in $\mathcal{S}_k$, the value of the cost function can only decrease in consecutive iterations. The multiple $\mathbf{b}-\mathbf{A}\mathbf{x}_k$ of the cost function gradient ensures descent if the current iterate does not satisfy the first-order optimality conditions. The eigenvector associated with the smallest eigenvalue will dislodge the iterates from a nonoptimal stationary point. We also use this vector in a "safe-guard" strategy designed to keep $\mathbf{A}+\mu_k\mathbf{I}$ positive definite.

4.1. The SQP system. Now consider the SQP system (3.5)–(3.6). According to (3.6), $\mathbf{z}$ is orthogonal to the prior iterate $\mathbf{x}_k$. Let $\mathbf{P}$ be the matrix that projects a vector into the space perpendicular to $\mathbf{x}_k$:

$$\mathbf{P} = \mathbf{I} - \frac{\mathbf{x}_k\mathbf{x}_k^{\mathsf{T}}}{\mathbf{x}_k^{\mathsf{T}}\mathbf{x}_k}.$$

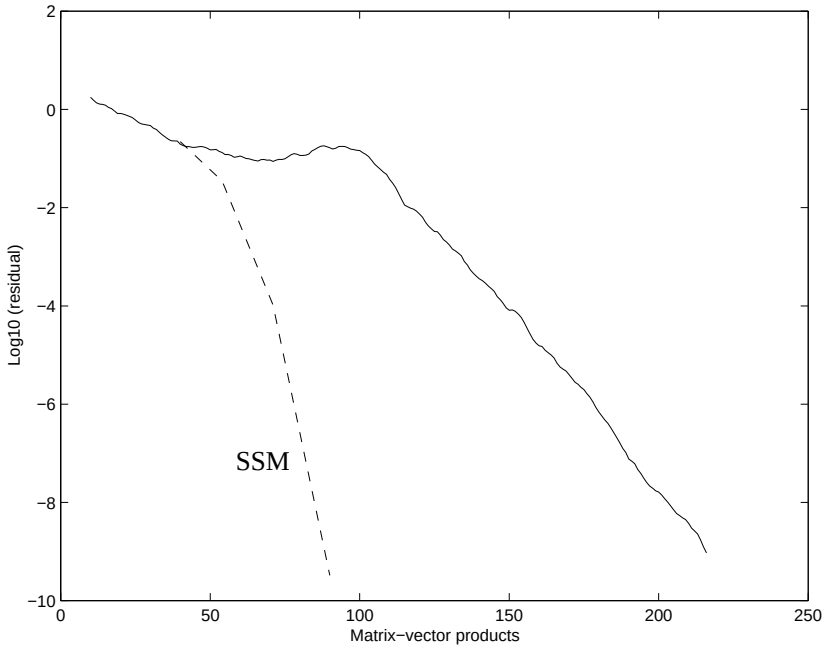


FIG. 4.1. *Convergence of the tridiagonalization approach* (*solid*) *and SSM* (*dashed*) *for the second test problem from* [24].

Multiplying (3.5) by $\mathbf{P}$ yields

$$\mathbf{P}(\mathbf{A}+\mu_k\mathbf{I})\mathbf{z} = \mathbf{P}(\mathbf{b}-\mathbf{A}\mathbf{x}_k).$$

Since $\mathbf{P}\mathbf{z} = \mathbf{z}$, according to (3.6), we have

$$\mathbf{P}(\mathbf{A}+\mu_k\mathbf{I})\mathbf{P}\mathbf{z} = \mathbf{P}(\mathbf{b}-\mathbf{A}\mathbf{x}_k). \tag{4.1}$$

We have found that preconditioned Krylov space methods, such as the Gauss–Seidel scheme in [9], converge very quickly when applied to (4.1). As a small illustration, let us consider the second test problem from [24] with $r = 100$ and $\mathbf{A} = \mathbf{Q}\boldsymbol{\Delta}\mathbf{Q}$, where $\boldsymbol{\Delta}$ is a 1000×1000 diagonal matrix with diagonal elements selected randomly from a uniform distribution on $(-.5, .5)$ and $\mathbf{Q} = \mathbf{I} - 2\mathbf{q}\mathbf{q}^{\mathsf{T}}$, where $\mathbf{q}$ is obtained by first generating random numbers on $(-.5, .5)$ and then scaling the resulting vector to have unit length. The vector $\mathbf{b}$ is generated in the same way as $\mathbf{q}$. The solid curve in Figure 4.1 gives the convergence when a Lanczos type process (Algorithm 1, with starting vector $\mathbf{v}_1 = \mathbf{P}\mathbf{b}$) is used to generate the matrix $\mathbf{V}$ used in (3.2). The Lanczos process was modified to ensure orthogonality of the columns of $\mathbf{V}$. For each value of l in Algorithm 1, we solve the $l \times l$ tridiagonal problem (3.3) to obtain an approximate solution $\mathbf{x}$ and associated multiplier $\mu = \rho(\mathbf{x})$ for the original problem (2.1). In the solid curve of Figure 4.1, we plot the base 10 logarithm of the norm of the residual $\|\mathbf{b} - (\mathbf{A} + \mu\mathbf{I})\mathbf{x}\|$. According to Lemma 2.1, the residual vanishes at an optimal solution.

The dashed curve of Figure 4.1, based on the SSM approach, is obtained in the following way: Taking $l = 40$ in Algorithm 1, we generate a $\mathbf{V}$ with 40 orthonormal columns. Solving (3.3), we obtain a starting guess of $\mathbf{x}_0$. In iteration k of the SSM

phase, we start with the vector $\mathbf{v}_1 = \mathbf{P}(\mathbf{b}-\mathbf{A}\mathbf{x}_k)$ and we use the Gauss–Seidel/Krylov space approach of [9] to generate a matrix $\mathbf{V}$, with orthonormal columns, that approximately contains a solution of (4.1) in its range. Using the $\mathbf{V}$ generated in this way, we solve (3.2) to obtain the next iterate $\mathbf{x}_{k+1}$. The associated multiplier is estimated using (3.7). Each kink in the dashed curve of Figure 4.1 corresponds to the number of iterations needed to obtain an approximate solution of (4.1). In this example, roughly 15 multiplications by the elements of the matrix $\mathbf{A}$ are used to solve (4.1). The quadratic convergence of SSM is reflected in the rapid decay of the residual norm.

This approach for generating $\mathbf{V}$, using a nonsymmetric Gauss–Seidel matrix, Krylov spaces, and orthogonalization, can become expensive when n is really large since each of the columns of $\mathbf{V}$ should be stored in memory. Hence, in the remainder of this paper, we focus on low-storage symmetric techniques for solving (4.1), which we compare to other approaches.

We solve (4.1) using a preconditioned version of Paige and Saunders' MINRES algorithm [17]. More precisely, we use Algorithms 3 and 3a in [9] and three different choices for the symmetrizing preconditioner $\mathbf{W}$ in that paper: (i) $\mathbf{W} = \mathbf{I}$, corresponding to unconditioned iterations; (ii) $\mathbf{W} = \mathbf{D}^{1/2}$, where $\mathbf{D}$ is the diagonal matrix whose diagonal matches that of $\mathbf{C} = \mathbf{P}(\mathbf{A} + \mu_k\mathbf{I})\mathbf{P}$ (Jacobi symmetrization); (iii) $\mathbf{W} = \mathbf{D}^{-1/2}(\mathbf{L} + \mathbf{D})$, where $\mathbf{L}$ is the strictly lower triangular matrix whose lower triangle matches that of $\mathbf{C}$ (SSOR symmetrization). The implementations of SSM associated with the latter two preconditioners are denoted SSM_d and SSM_l, respectively.

Typically, the $\mathbf{L}$ matrix associated with $\mathbf{C} = \mathbf{P}(\mathbf{A}+\mu_k\mathbf{I})\mathbf{P}$ is dense, even when $\mathbf{A}$ is sparse, since $\mathbf{P}$ is often dense. Nonetheless, linear systems of the form $(\mathbf{L}+\mathbf{D})\mathbf{y} = \mathbf{g}$ can be solved in time proportional to the number of nonzero elements in the lower triangle of $\mathbf{A}$, due to the special structure of $\mathbf{C}$. In terms of the vectors $\mathbf{w}$, $\mathbf{q}$, and $\mathbf{p}$ defined by

$$\mathbf{w} = \mathbf{x}_k/\|\mathbf{x}_k\|, \quad \mathbf{q} = (\mathbf{A} + \mu_k\mathbf{I})\mathbf{w}, \quad \text{and} \quad \mathbf{p} = \mathbf{q} - (\mathbf{q}^{\mathsf{T}}\mathbf{w})\mathbf{w},$$

the diagonal $\mathbf{d}$ of $\mathbf{C}$ can be expressed

$$d_i = a_{ii} + \mu_k - (p_i + q_i)w_i,$$

while the off-diagonal elements of $\mathbf{C}$ are

$$c_{ij} = a_{ij} - w_i q_j - p_i w_j, \quad i \neq j.$$

Exploiting this structure, it can be shown that the solution to $(\mathbf{L} + \mathbf{D})\mathbf{y} = \mathbf{g}$ can be computed in the following way.

ALGORITHM 2 ($\mathbf{y} = (\mathbf{L} + \mathbf{D})^{-1}\mathbf{g}$, $\mathbf{L} + \mathbf{D} + \mathbf{L}^{\mathsf{T}} = \mathbf{P}(\mathbf{A} + \mu_k\mathbf{I})\mathbf{P}$, $\mathbf{P} = \mathbf{I} - \mathbf{w}\mathbf{w}^{\mathsf{T}}$).

$\mathbf{y} = \mathbf{g}$, $s = 0$, $t = 0$
for $i = 1 : n - 1$
  $y_i = (y_i + sw_i + tp_i)/d_i$
  $s = s + q_i y_i$
  $t = t + w_i y_i$
  $y_{i+1:n} = y_{i+1:n} - y_i a_{i+1:n,i}$
end
$y_n = (y_n + sw_n + tp_n)/d_n$
END ALGORITHM 2

The statement $y_{i+1:n} = y_{i+1:n} - y_i a_{i+1:n,i}$ of Algorithm 2 requires only the nonzero elements in column i of $\mathbf{A}$ beneath the diagonal. Hence, the number of floating point operations for Algorithm 2 is $O(n)$ plus the number of nonzero elements in the lower triangle of $\mathbf{A}$.

The analogous procedure for the transposed system is the following.

ALGORITHM 3 ($\mathbf{y} = (\mathbf{L}+\mathbf{D})^{-\mathsf{T}}\mathbf{g}$, $\mathbf{L}+\mathbf{D}+\mathbf{L}^{\mathsf{T}} = \mathbf{P}(\mathbf{A}+\mu_k\mathbf{I})\mathbf{P}$, $\mathbf{P} = \mathbf{I} - \mathbf{w}\mathbf{w}^{\mathsf{T}}$).

$\mathbf{y} = \mathbf{g}$, $s = 0$, $t = 0$
for $i = n : -1 : 2$
  $y_i = (y_i + sw_i + tq_i)/d_i$
  $s = s + p_i y_i$
  $t = t + w_i y_i$
  $y_{1:i-1} = y_{1:i-1} - y_i a_{1:i-1,i}$
end
$y_1 = (y_1 + sw_1 + tq_1)/d_1$

END ALGORITHM 3

4.2. Positive definiteness. In theory, the MINRES algorithm we use to solve (4.1) can be applied to any symmetric matrix. In practice, convergence can be extremely slow when $\mathbf{C}$ is indefinite. For this reason, we try to choose μ_k so that $\mathbf{A}+\mu_k\mathbf{I}$ is positive definite. If $\mathbf{e}$ is an eigenvector of the matrix $\mathbf{B}$ in (3.3) associated with the smallest eigenvalue σ, then the pair $(\mathbf{v}, \sigma)$, where $\mathbf{v} = \mathbf{V}\mathbf{e}/\|\mathbf{V}\mathbf{e}\|$, approximates an eigenpair of $\mathbf{A}$ corresponding to the smallest eigenvalue. The error in σ can be estimated in the following way: If σ is closer to λ_1 than the other eigenvalues of $\mathbf{A}$, then after substituting

$$\mathbf{v} = \sum_{i=1}^{n} \nu_i \boldsymbol{\phi}_i, \quad \nu_i = \mathbf{v}^{\mathsf{T}}\boldsymbol{\phi}_i,$$

in the residual $\mathbf{r} = \mathbf{A}\mathbf{v} - \sigma\mathbf{v}$, we have

$$\|\mathbf{r}\|^2 = \sum_{i=1}^{n} |\sigma - \lambda_i|^2 \nu_i^2 \geq \sum_{i=1}^{n} |\sigma - \lambda_1|^2 \nu_i^2 = |\sigma - \lambda_1|^2,$$

since $\sum_{i=1}^{n} \nu_i^2 = 1$. Thus $|\sigma - \lambda_1| \leq \|\mathbf{r}\|$, which implies that

$$\lambda_1 \geq \sigma - \|\mathbf{r}\|.$$

With this insight, we replace the least squares estimate (3.7) by the safe-guarded estimate

$$\mu_k = \max\{\|\mathbf{r}\| - \sigma, \rho(\mathbf{x}_k)\}. \tag{4.2}$$

This choice for μ_k helps to ensure that $\mathbf{A} + \mu_k\mathbf{I}$ is positive definite, often leading to faster convergence of iterative methods applied to (4.1).

When the approximate eigenpair $(\mathbf{v}, \sigma)$ is not very accurate, then the safe-guarded step (4.2) is a safe, but poor, approximation to μ^*. Hence, whenever $\mu_k = \|\mathbf{r}\| - \sigma$, we apply one iteration of SSM to the quadratic eigenvalue problem (1.2) in order to compute a more accurate eigenpair. Due to the third- and sixth-order estimates in (3.8), simply one iteration of SSM for the eigenproblem often yields a highly accurate eigenpair.

4.3. The algorithm. We now collect our observations and present the algorithm that was used to generate the numerical results of the next section. To simplify the presentation, we introduce the following subroutines:

- $\mathbf{V} = \text{Lanczos}(\mathbf{A}, \mathbf{v}_1, l)$: This routine applies Algorithm 1 to the matrix $\mathbf{A}$, starting from the vector $\mathbf{v}_1$, to generate a matrix $\mathbf{V}$ with columns $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_l$.
- $(\mathbf{x}, \mu, \mathbf{v}, \sigma) = \text{SSM}(\mathbf{A}, \mathbf{b}, \mathcal{S}_k)$: This routine solves the problem (3.1), generating a solution denoted $\mathbf{x}$ and an associated multiplier $\mu = \rho(\mathbf{x})$. If $\mathbf{V}$ is a matrix whose columns are an orthonormal basis for $\mathcal{S}_k$, then an estimate $(\mathbf{v}, \sigma)$ for the smallest eigenvalue of $\mathbf{A}$ and an associated eigenvector is obtained by computing the smallest eigenvalue σ and an associated eigenvector $\mathbf{e}$ for $\mathbf{B} = \mathbf{V}^\mathsf{T}\mathbf{A}\mathbf{V}$ and setting $\mathbf{v} = \mathbf{V}\mathbf{e}$.
- $\mathbf{z} = \text{SQP}(\mathbf{A}, \mu, \mathbf{b}, \mathbf{x})$: This routine computes a (minimum residual, minimum norm) solution $(\mathbf{z}, \nu)$ of the linear system

$$\left[\begin{array}{c|c} \mathbf{A}+\mu\mathbf{I} & \mathbf{x} \\ \mathbf{x}^\mathsf{T} & \mathbf{0} \end{array}\right]\left[\begin{array}{c} \mathbf{z} \\ \nu \end{array}\right] = \left[\begin{array}{c} \mathbf{b}-(\mathbf{A}+\mu\mathbf{I})\mathbf{x} \\ 0 \end{array}\right].$$

Our implementation of the sequential subspace method combines these three routines and the safe-guarded step (4.2).

ALGORITHM 4 (SAFE-GUARDED SSM WITH LANCZOS STARTUP).

$it = \nu = \mu = 0,\ \mathbf{v} = \mathbf{x} = \mathbf{0},\ \mathbf{c} = \text{rand}(n,1) - .5$
$\mathbf{u} = \mathbf{c}/(100\|\mathbf{c}\|) + \mathbf{b}/(r\|\mathbf{A}\|)$
while ($\nu == \mu$ & $it = it + 1 \le \bar{it}$)
  $\mathbf{V} = \text{Lanczos}(\mathbf{A}, \mathbf{u}, l)$
  $(\mathbf{x}, \mu, \mathbf{v}, \sigma) = \text{SSM}(\mathbf{A}, \mathbf{b}, \text{span}(\mathbf{x}, \mathbf{v}, \mathbf{v}_1, \ldots, \mathbf{v}_l))$
  $\nu = \|(\mathbf{A} - \sigma\mathbf{I})\mathbf{v}\| - \sigma$
  if ($\nu > \mu$) $\mu = \nu$
  $\mathbf{u} = \mathbf{b} - (\mathbf{A} + \mu\mathbf{I})\mathbf{x}$
end
while ($\|\mathbf{b} - (\mathbf{A} + \mu\mathbf{I})\mathbf{x}\| > tol$)
  $\mathbf{z} = \text{SQP}(\mathbf{A}, \mu, \mathbf{b}, \mathbf{x})$
  $\mathcal{S} = \text{span}(\mathbf{x}, \mathbf{z}, \mathbf{v}, \mathbf{b} - \mathbf{A}\mathbf{x})$
  $(\mathbf{x}, \mu, \mathbf{v}, \sigma) = \text{SSM}(\mathbf{A}, \mathbf{b}, \mathcal{S})$
  $\nu = \|(\mathbf{A} - \sigma\mathbf{I})\mathbf{v}\| - \sigma$
  $\epsilon = \|\mathbf{b} - (\mathbf{A} + \mu\mathbf{I})\mathbf{x}\|$
  if ($\nu > \mu$ & $\nu + \sigma > \epsilon/r$)
    $\mathbf{z} = \text{SQP}(\mathbf{A}, \nu, \mathbf{0}, \mathbf{v})$
    $(\mathbf{x}, \mu, \mathbf{v}, \sigma) = \text{SSM}(\mathbf{A}, \mathbf{b}, \text{span}(\mathcal{S}, \mathbf{z}))$
    $\nu = \|(\mathbf{A} - \sigma\mathbf{I})\mathbf{v}\| - \sigma$
  end
  if ($\nu > \mu$) $\mu = \nu$
end

END ALGORITHM 4

For the computational results reported in the next section, we took $\bar{it} = 3$ and $l = \max\{10, .01n\}$. The "rand" function appearing at the start of Algorithm 4 generates a vector with components uniformly distributed on $[0, 1]$.

5. Computational results. In this section we compare the performance of SSM to the performance of the algorithms in [7, 20, 24], denoted GLRT, RW, and S, respectively, using the three test problems presented in [24]. The results that we

TABLE 5.1
Problem 1*, average number of matrix-vector products versus tolerance.*

Tolerance	S	RW	GLRT	SSM	SSM_d	SSM_l
10^{-4}	249.0 (04.2)	383.6 (3)	51.0	78.0	51.2	44.2
10^{-6}	824.0 (08.4)	460.7 (4)	65.7	107.1	65.5	54.3
10^{-8}	1633.4 (12.3)	465.7 (4)	86.7	124.3	86.7	70.7

TABLE 5.2
Problem 2*, average number of matrix-vector products versus radius.*

Radius	S	RW	GLRT	SSM	SSM_d	SSM_l
10	240 (08)	1437.9 (5.5)	27.0	88.3	42.3	54.1
100	579 (13)	2567.7 (7.8)	188.8	353.7	88.4	136.2

report for S were extracted from [24], while the results reported for GLRT and RW were obtained using codes provided by the authors. Each of these codes used different stopping criteria. GLRT stopped when $\|\mathbf{b}-(\mathbf{A}+\mu\mathbf{I})\mathbf{x}\|/\|\mathbf{b}\|$ was bounded by a given tolerance, while RW stopped when the gap between the value of the primal and dual problem, and hence the error in the primal cost function, was smaller than a given tolerance. In order to ensure that each code computed a solution with the same accuracy, we adjusted the error tolerance parameter of each code until the value of $\|\mathbf{b}-(\mathbf{A}+\mu\mathbf{I})\mathbf{x}\|$ for the computed solution was smaller than a given tolerance (specified below).

In the first test problem of [24], $\mathbf{A} = \mathbf{A}_0 - 5\mathbf{I}$, where $\mathbf{A}_0$ is the standard 2-D discrete Laplacian on the unit square based on a 5-point stencil with equally spaced mesh points. Taking $n = 32^2 = 1024$ and $r = 100$, a series of 20 problems was generated, where $\mathbf{b}$ was a vector with elements uniformly distributed on $[0, 1]$. Each of these problems was solved using three different tolerances, 10^{-4}, 10^{-6}, and 10^{-8}. In Table 5.1 we give the average number of matrix-vector products involving $\mathbf{A}$ for each algorithm. Each iteration of the preconditioned MINRES algorithm with lower triangular preconditioner involves roughly twice as many flops as an iteration of either the identity or the diagonal preconditioned schemes. Hence, in doing the bookkeeping, we charged for two matrix-vector products in each iteration of the triangular preconditioned scheme. As seen in Table 5.1, SSM_l converges more than twice as fast as the identity and diagonal preconditioned schemes and, overall, SSM_l uses the smallest number of matrix-vector products for this test problem. Since the parametric eigenvalue algorithms S and RW compute an extreme eigenvalue for a series of matrices, we also list in parentheses in Table 5.1 the number of these eigenproblems that are solved. Hence, RW is very economical in terms of the number of these eigenproblems that are solved.

The second suite of test problems in [24] utilizes the matrix described earlier in section 3. In these problems, the radius of the sphere is varied and the number of matrix-vector products is tabulated. For radii of one or smaller, solutions can be computed extremely quickly, so we focused on $r = 10$ and $r = 100$ and an error tolerance of 10^{-7}. In Table 5.2 we see that for $r = 100$, SSM_d had the fewest matrix-vector products, while GLRT had the fewest for $r = 10$.

The final problem of [24] again employed the discrete Laplacian matrix, but with $n = 16^2$ and $r = 100$. The vector $\mathbf{b}$ was designed to make the problem degenerate; first a random $\mathbf{b}$ was generated, then its $\boldsymbol{\phi}_1$ component was removed. Table 5.3 gives the results for the various algorithms.

TABLE 5.3
Problem 3, average number of matrix-vector products.

S	RW	GLRT	SSM	SSM_d	SSM_l
291 (11)	441.0 (5.4)	134.0*	179.3	179.2	161.5

We placed an asterisk by the result in Table 5.3 for GLRT since this routine reduced the error to 10^{-4}, not the 10^{-7} tolerance used by the other routines. Among the routines that achieved the error tolerance, SSM_l performed the best relative to the number of matrix-vector products. Note that the number of matrix-vector products given in Table 5.3 for S was taken from [24] while Rojas, in her recent thesis [21], developed a more efficient implementation of Sorensen's approach for degenerate problems.

In summary, a Lanczos type process seems to be very effective when the problem is very nondegenerate ($\mu^* >> -\lambda_1$). As the problem becomes more degenerate, preconditioned schemes such as SSM_d or SSM_l appear more effective. The number of times that RW computes an extreme eigenpair is often around 5. For the numerical experiments reported in this paper, Matlab's eigs routine was used to compute this extreme eigenpair. If this routine for computing an extreme eigenpair could be sped up, possibly using the Jacobi type methods of Sleijpen and Van der Vorst [22] or the truncated RQ iteration of Sorensen and Yang [25], the number of matrix-vector operations used in the parametric eigenvalue approach would be reduced.

Acknowledgments. The author gratefully acknowledges the comments and suggestions of the referees. He also thanks Henry Wolkowicz for pointing out the related paper [7] and for his comments and suggestions, and the authors of [24] for providing access to their codes.

REFERENCES

[1] R. H. BYRD, R. B. SCHNABEL, AND G. A. SCHULTZ, *A trust region algorithm for nonlinearly constrained optimization*, SIAM J. Numer. Anal., 24 (1987), pp. 1152–1170.
[2] M. R. CELIS, J. E. DENNIS, AND R. A. TAPIA, *A trust region strategy for nonlinear equality constrained optimization*, in Numerical Optimization 1984, P. T. Boggs, R. H. Byrd, and R. B. Schnabel, eds., SIAM, Philadelphia, PA, 1985, pp. 71–82.
[3] M. EL-ALEM, *A global convergence theory for the Celis–Dennis–Tapia trust-region algorithm for constrained optimization*, SIAM J. Numer. Anal., 28 (1991), pp. 266–290.
[4] W. GANDER, *Least squares with a quadratic constraint*, Numer. Math., 36 (1981), pp. 291–307.
[5] G. H. GOLUB AND U. VON MATT, *Quadratically constrained least squares and quadratic problems*, Numer. Math., 59 (1991), pp. 561–580.
[6] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 1989.
[7] N. I. M. GOULD, S. LUCIDI, M. ROMA, AND P. L. TOINT, *Solving the trust-region subproblem using the Lanczos method*, SIAM J. Optim., 9 (1999), pp. 504–525.
[8] W. W. HAGER, *Applied Numerical Linear Algebra*, Prentice-Hall, Englewood Cliffs, NJ, 1988.
[9] W. W. HAGER, *Iterative methods for nearly singular linear systems*, SIAM J. Sci. Comput., 22 (2000), pp. 747–766.
[10] W. W. HAGER AND Y. KRYLYUK, *Graph partitioning and continuous quadratic programming*, SIAM J. Discrete Math., 12 (1999), pp. 500–523.
[11] P. C. HANSEN, *Regularization tools: A MATLAB package for analysis and solution of discrete ill-posed problems*, Numer. Algorithms, 6 (1994), pp. 1–35.
[12] P. C. HANSEN, *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*, SIAM, Philadelphia, PA, 1998.
[13] R. B. LEHOUCQ, D. C. SORENSEN, AND C. YANG, *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicity Restarted Arnoldi Methods*, SIAM, Philadelphia, PA, 1998.

[14] W. Menke, *Geophysical Data Analysis: Discrete Inverse Theory*, Academic Press, San Diego, CA, 1989.
[15] J. J. Moré, *Recent developments in algorithms and software for trust region methods*, in Mathematical Programming: State of the Art, A. Bachem, M. Grotschel, and B. Korte, eds., Springer-Verlag, Berlin, 1983, pp. 258–287.
[16] J. J. Moré and D. C. Sorensen, *Computing a trust region step*, SIAM J. Sci. Stat. Comput., 4 (1983), pp. 553–572.
[17] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.
[18] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.
[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. Rendl and H. Wolkowicz, *A semidefinite framework for trust region subproblems with applications to large scale minimization*, Math. Programming, 77 (1997), pp. 273–299.
[21] M. Rojas, *A Large-Scale Trust-Region Approach to the Regularization of Discrete Ill-Posed Problems*, Ph.D. thesis, Computational and Applied Mathematics, Rice University, Houston, TX, 1998.
[22] G. L. G. Sleijpen and H. A. Van der Vorst, *A Jacobi–Davidson iteration method for linear eigenvalue problems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 401–425.
[23] D. C. Sorensen, *Newton's method with a model trust region modification*, SIAM J. Numer. Anal., 19 (1982), pp. 409–426.
[24] D. C. Sorensen, *Minimization of a large-scale quadratic function subject to a spherical constraint*, SIAM J. Optim., 7 (1997), pp. 141–161.
[25] D. C. Sorensen and C. Yang, *A truncated RQ iteration for large scale eigenvalue calculations*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 1045–1073.
[26] A. Tarantola, *Inverse Problem Theory*, Elsevier, Amsterdam, The Netherlands, 1987.

SIAM J. OPTIM.
Vol. 12, No. 1, pp. 209–237

REDUCED-HESSIAN QUASI-NEWTON METHODS FOR UNCONSTRAINED OPTIMIZATION*

PHILIP E. GILL† AND MICHAEL W. LEONARD‡

Abstract. Quasi-Newton methods are reliable and efficient on a wide range of problems, but they can require many iterations if the problem is ill-conditioned or if a poor initial estimate of the Hessian is used. In this paper, we discuss methods designed to be more efficient in these situations. All the methods to be considered exploit the fact that quasi-Newton methods accumulate approximate second-derivative information in a sequence of expanding subspaces. Associated with each of these subspaces is a certain *reduced* approximate Hessian that provides a complete but compact representation of the second derivative information approximated up to that point. Algorithms that compute an explicit reduced-Hessian approximation have two important advantages over conventional quasi-Newton methods. First, the amount of computation for each iteration is significantly less during the early stages. This advantage is increased by forcing the iterates to *linger* on a manifold whose dimension can be significantly smaller than the subspace in which curvature has been accumulated. Second, approximate curvature along directions that lie off the manifold can be reinitialized as the iterations proceed, thereby reducing the influence of a poor initial estimate of the Hessian. These advantages are illustrated by extensive numerical results from problems in the CUTE test set. Our experiments provide strong evidence that reduced-Hessian quasi-Newton methods are more efficient and robust than conventional BFGS methods and some recently proposed extensions.

Key words. unconstrained optimization, quasi-Newton methods, BFGS method, conjugate-direction methods

AMS subject classifications. 65K05, 90C30

PII. S1052623400307950

1. Introduction.

Quasi-Newton methods are arguably the most effective methods for finding a minimizer of a smooth nonlinear function $f : \mathbb{R}^n \to \mathbb{R}$ when second derivatives are either unavailable or too expensive to calculate. Quasi-Newton methods build up second-derivative information by estimating the curvature along a sequence of search directions. Each curvature estimate is installed in an approximate Hessian by applying a rank-one or rank-two update. One of the most successful updates is the Broyden–Fletcher–Goldfarb–Shanno (BFGS) formula, which is a member of the wider Broyden class of rank-two updates (see section 2 for details).

Despite the success of these methods on a wide range of problems, it is well known that conventional quasi-Newton methods can require a disproportionately large number of iterations and function evaluations on some problems. This inefficiency may be caused by a poor choice of initial approximate Hessian or may result from the search direction's being poorly defined when the Hessian is ill-conditioned. This paper is concerned with the formulation of methods that are less susceptible to these difficulties.

All the methods to be discussed are based on exploiting an important property of quasi-Newton methods in which second-derivative information is accumulated in a

*Received by the editors January 19, 2000; accepted for publication (in revised form) March 5, 2001; published electronically October 18, 2001.

http://www.siam.org/journals/siopt/12-1/30795.html

†Department of Mathematics, University of California, San Diego, La Jolla, CA 92093-0112 (pgill@ucsd.edu). This author's research was supported by National Science Foundation grant DMI-9424639 and Office of Naval Research grant N00014-96-1-0274.

‡Department of Mathematics, University of California, Los Angeles, CA 90095-1555 (na.mleonard@na-net.ornl.gov). This author's research was supported by National Science Foundation grant DMI-9424639.

sequence of expanding subspaces. At the kth iteration ($k < n$) curvature is known along vectors that lie in a certain *gradient subspace* whose dimension is no greater than $k+1$. This property is well documented in the context of solving positive-definite symmetric systems $Ax = b$. In particular, the iterates lie on an expanding sequence of manifolds characterized by the Krylov subspace associated with A (see, e.g., Freund, Golub, and Nachtigal [7] and Kelley [14, p. 12]). These manifolds are identical to those associated with the BFGS method applied to minimizing the quadratic $c - b^T x + \frac{1}{2} x^T A x$. (For further details of the equivalence of quasi-Newton methods and conjugate-gradient methods, see Nazareth [22].)

In the quasi-Newton context, the availability of an explicit basis for the gradient subspace makes it possible to represent the approximate curvature in terms of a *reduced* approximate Hessian matrix with order at most $k + 1$. Quasi-Newton algorithms that explicitly calculate a reduced Hessian have been proposed by Fenelon [4] and Nazareth [21], who also considered modified Newton methods in the same context. Siegel [27] has proposed methods that work with a reduced inverse approximate Hessian. In practical terms, the reduced-Hessian formulation can require significantly less work per iteration when k is small relative to n. This property can be exploited by forcing iterates to *linger* on a manifold while the objective function is minimized to greater accuracy. While iterates linger, the search direction is calculated from a system that is generally smaller than the reduced Hessian. In many practical situations convergence occurs before the dimension of the lingering subspace reaches n, resulting in substantial savings in computing time (see section 7).

More recently, Siegel [28] has proposed the conjugate-direction scaling algorithm, which is a quasi-Newton method based on a conjugate-direction factorization of the inverse approximate Hessian. Although no explicit reduced Hessian is updated, the method maintains a basis for the expanding subspaces and allows iterates to linger on a manifold. The method also has the benefit of finite termination on a quadratic (see Leonard [16, p. 77]). More importantly, Siegel's method includes a feature that can considerably enhance the benefits of lingering. Siegel notes that the search direction is the sum of two vectors: one with the scale of the estimated derivatives and the other with the scale of the initial approximate Hessian. Siegel suggests rescaling the second vector using newly computed approximate curvature. Algorithms that combine lingering and rescaling have the potential for giving significant improvements over conventional quasi-Newton methods. Lingering forces the iterates to remain on a manifold until the curvature has been sufficiently established; rescaling ensures that the initial curvature in the unexplored manifold is commensurate with curvature already found.

In this paper we propose several algorithms based on maintaining the triangular factors of an explicit reduced Hessian. We demonstrate how these factors can be used to force the iterates to linger while curvature information continues to be accumulated along directions lying off the manifold. With the BFGS method, it is shown that while lingering takes place, the new curvature is restricted to an upper-trapezoidal portion of the factor of the reduced Hessian and the remaining portion retains the diagonal structure of the initial approximate Hessian. It follows that conjugate-direction scaling is equivalent to simply *reinitializing* the diagonal part of the reduced Hessian with freshly computed curvature information.

Despite the similarities between reduced-Hessian reinitialization and conjugate-direction scaling, it must be emphasized that these methods are not the same, in the sense that they involve very different storage and computational overheads. More-

over, the reduced-Hessian factorization has both practical and theoretical advantages over Siegel's conjugate-direction factorization. On the practical side, the early search directions can be calculated with significantly less work. This can result in a significantly faster minimization when the dimension of the manifold grows relatively slowly, as it does on many problems (see sections 6 and 7). On the theoretical side, the simple structure exhibited by the reduced-Hessian factor allows the benefits of reinitialization to be extended to the large-scale case (see Gill and Leonard [9]).

A reduced-Hessian method allows *expansion* of the manifold on which curvature information is known. Thus, when implementing software, it is necessary either to allocate new memory dynamically as the reduced Hessian grows or to reserve sufficient storage space in advance. In practice, however, the order of the reduced Hessian often remains much less than n, i.e., the problem is solved without needing room for an $n \times n$ matrix. Notwithstanding this benefit, on very large problems it may be necessary to *explicitly* limit the amount of storage used, by placing a limit on the order of the reduced Hessian. Such *limited-memory* reduced-Hessian methods discard old curvature information whenever the addition of new information causes a predefined storage limit to be exceeded. Methods of this type have been considered by Fenelon [4] and Siegel [27]. Limited-memory methods directly related to the methods considered in this paper are discussed by Gill and Leonard [9].

The paper is organized as follows. Section 2 contains a discussion of various theoretical aspects of reduced-Hessian quasi-Newton methods, concluding with the statement of Algorithm RH, a reduced-Hessian formulation of a conventional quasi-Newton method. Algorithm RH is the starting point for the improved algorithms presented in sections 3 and 4. Section 3 is concerned with the effects of lingering on the form of the factorization of the reduced Hessian. In section 4, Siegel's conjugate-direction scaling algorithm is reformulated as an explicit reduced-Hessian method. In section 4.1 we present a reduced-Hessian method that combines lingering with reinitialization. The convergence properties of this algorithm are discussed in sections 4.2 and 4.3. To simplify the discussion, the algorithms of sections 2–4 are given with the assumption that all computations are performed in exact arithmetic. The effects of rounding error are discussed in section 5. Finally, sections 6 and 7 include some numerical results when various reduced-Hessian algorithms are applied to test problems taken from the CUTE test collection (see Bongartz et al. [1]). Section 6 also includes comparisons with Siegel's method and with Lalee and Nocedal's automatic column-scaling method [15], which is another extension of the BFGS method. Results from the package NPSOL [10] are provided to illustrate how the reduced-Hessian approach compares to a conventional quasi-Newton method. Our experiments demonstrate that reduced-Hessian methods can require substantially less computer time than these alternatives. Part of the reduction in computer time corresponds to the smaller number of iterations and function evaluations required when using the reinitialization strategy. However, much of this reduction comes from the fact that the average cost of an iteration is less than for competing methods.

Unless explicitly indicated otherwise, $\|\cdot\|$ denotes the vector two-norm or its subordinate matrix norm. The vector e_i is used to denote the ith unit vector of the appropriate dimension. A floating-point operation, or *flop*, refers to a calculation of the form $\alpha x + y$, i.e., a multiplication and an addition.

2. Background. Given a twice-continuously differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ with gradient vector ∇f and Hessian matrix $\nabla^2 f$, the kth iteration of a quasi-Newton

method is given by

$$H_k p_k = -\nabla f(x_k), \qquad x_{k+1} = x_k + \alpha_k p_k, \tag{2.1}$$

where H_k is a symmetric, positive-definite matrix, p_k is the search direction, and α_k is a scalar step length. If H_k is interpreted as an approximation to $\nabla^2 f(x_k)$, then $x_k + p_k$ can be viewed as minimizing a quadratic model of f with Hessian H_k. The matrix H_{k+1} is obtained from H_k by adding a low-rank matrix defined in terms of $\delta_k = x_{k+1} - x_k$ and $\gamma_k = g_{k+1} - g_k$, where $g_k = \nabla f(x_k)$. Updates from the Broyden class give a matrix H_{k+1} such that

$$H_{k+1} = H_k - \frac{1}{\delta_k^T H_k \delta_k} H_k \delta_k \delta_k^T H_k + \frac{1}{\gamma_k^T \delta_k} \gamma_k \gamma_k^T + \phi_k (\delta_k^T H_k \delta_k) w_k w_k^T, \tag{2.2}$$

where $w_k = \gamma_k / \gamma_k^T \delta_k - H_k \delta_k / \delta_k^T H_k \delta_k$, and ϕ_k is a scalar parameter. It is generally accepted that the most effective update corresponds to $\phi_k = 0$, which defines the well-known BFGS update

$$H_{k+1} = H_k - \frac{1}{\delta_k^T H_k \delta_k} H_k \delta_k \delta_k^T H_k + \frac{1}{\gamma_k^T \delta_k} \gamma_k \gamma_k^T. \tag{2.3}$$

For brevity, the term "Broyden's method" refers to a method based on iteration (2.1) when used with an update from the Broyden class. Similarly, the term "BFGS method" refers to iteration (2.1) with the BFGS update.

The scalar $\gamma_k^T \delta_k$, known as the *approximate curvature*, is a difference estimate of the (unknown) curvature $\delta_k^T \nabla^2 f(x_k) \delta_k$. Each Broyden update gives an approximate Hessian satisfying $\delta_k^T H_{k+1} \delta_k = \gamma_k^T \delta_k$, which implies that the approximate curvature $\gamma_k^T \delta_k$ is installed as the *exact* curvature of the new quadratic model in the direction δ_k. It follows that a positive value for the approximate curvature is a necessary condition for H_{k+1} to be positive definite.

We follow common practice and restrict our attention to Broyden updates with the property that if H_k is positive definite, then H_{k+1} is positive definite if and only if $\gamma_k^T \delta_k > 0$. This restriction allows H_{k+1} to be kept positive definite by using a step length algorithm that ensures a positive value of the approximate curvature. Practical step length algorithms also include a criterion for sufficient descent. Two criteria often used are the Wolfe conditions

$$f(x_k + \alpha_k p_k) \leq f(x_k) + \mu \alpha_k g_k^T p_k \quad \text{and} \quad g_{k+1}^T p_k \geq \eta g_k^T p_k, \tag{2.4}$$

where the constants μ and η are chosen so that $0 \leq \mu \leq \eta < 1$ and $\mu < \frac{1}{2}$.

If n is sufficiently small that an $n \times n$ dense matrix can be stored explicitly, two alternative methods have emerged for implementing quasi-Newton methods. The first is based on using the upper-triangular matrix C_k such that $H_k = C_k^T C_k$ (see Gill et al. [10]). The second uses a matrix V_k satisfying the conjugate-direction identity $V_k^T H_k V_k = I$ (see Powell [25], Siegel [28]). Neither of these methods store H_k (or its inverse) as an explicit matrix. Instead, C_k or V_k is updated directly by exploiting the fact that every update from the Broyden class defines a rank-one update to C_k or V_k (see Goldfarb [12] and Dennis and Schnabel [3]). The rank-one update to C_k generally destroys the upper-triangular form of C_k. However, the updated C_k can be restored to upper-triangular form in $\mathcal{O}(n^2)$ operations.

2.1. Reduced-Hessian quasi-Newton methods. In this section, we review the formulation of conventional quasi-Newton methods as reduced-Hessian methods. The next key result is proved by Siegel [27] (see Fletcher and Powell [6], and Fenelon [4] for similar results in terms of the DFP and BFGS updates). Let $\mathcal{G}_k$ denote the subspace $\mathcal{G}_k = \text{span}\{g_0, g_1, \ldots, g_k\}$, and let $\mathcal{G}_k^\perp$ denote the orthogonal complement of $\mathcal{G}_k$ in $\mathbb{R}^n$.

LEMMA 2.1. *Consider the Broyden method applied to a general nonlinear function. If $H_0 = \sigma I$ ($\sigma > 0$), then $p_k \in \mathcal{G}_k$ for all k. Moreover, if $z \in \mathcal{G}_k$ and $w \in \mathcal{G}_k^\perp$, then $H_k z \in \mathcal{G}_k$ and $H_k w = \sigma w$.* ∎

Let r_k denote $\dim(\mathcal{G}_k)$, and let B_k (B for "basis") denote an $n \times r_k$ matrix whose columns form a basis for $\mathcal{G}_k$. An orthonormal basis matrix Z_k can be defined from the QR decomposition $B_k = Z_k T_k$, where T_k is a nonsingular upper-triangular matrix.[1] Let the $n - r_k$ columns of W_k define an orthonormal basis for $\mathcal{G}_k^\perp$. If Q_k is the orthogonal matrix $Q_k = \begin{pmatrix} Z_k & W_k \end{pmatrix}$, then the transformation $x = Q_k x_Q$ defines a transformed approximate Hessian $Q_k^T H_k Q_k$ and a transformed gradient $Q_k^T g_k$. If $H_0 = \sigma I$ ($\sigma > 0$), it follows from (2.2) and Lemma 2.1 that the transformation induces a block-diagonal structure, with

$$Q_k^T H_k Q_k = \begin{pmatrix} Z_k^T H_k Z_k & 0 \\ 0 & \sigma I_{n-r_k} \end{pmatrix} \quad \text{and} \quad Q_k^T g_k = \begin{pmatrix} Z_k^T g_k \\ 0 \end{pmatrix}. \tag{2.5}$$

The positive-definite matrix $Z_k^T H_k Z_k$ is known as a *reduced* approximate Hessian (or just reduced Hessian). The vector $Z_k^T g_k$ is known as a reduced gradient.

If we write the equation for the search direction as $(Q_k^T H_k Q_k) Q_k^T p_k = -Q_k^T g_k$, it follows from (2.5) that

$$p_k = Z_k q_k, \quad \text{where } q_k \text{ satisfies } Z_k^T H_k Z_k q_k = -Z_k^T g_k. \tag{2.6}$$

If the Cholesky factorization $Z_k^T H_k Z_k = R_k^T R_k$ is known, q_k can be computed from the forward substitution $R_k^T d_k = -Z_k^T g_k$ and back-substitution $R_k q_k = d_k$. A benefit of this approach is that Z_k and R_k require less storage than H_k when $k \ll n$ (see Gill and Leonard [9]). In addition, the computation of p_k when $k \ll n$ requires less work than it does for methods that store C_k or V_k. A benefit of using an orthonormal Z_k is that $\text{cond}(Z_k^T H_k Z_k) \leq \text{cond}(H_k)$, where $\text{cond}(\cdot)$ denotes the spectral condition number (see, e.g., Gill, Murray, and Wright [11, p. 162]).

There are a number of alternative choices for the basis B_k. Both Fenelon and Siegel propose that B_k be formed from a linearly independent subset of $\{g_0, g_1, \ldots, g_k\}$. With this choice, the orthonormal basis can be accumulated columnwise as the iterations proceed using Gram–Schmidt orthogonalization (see, e.g., Golub and Van Loan [13, pp. 218–220]). During iteration k, the number of columns of Z_k either remains unchanged or increases by one, depending on the value of the scalar ρ_{k+1}, such that $\rho_{k+1} = \|(I - Z_k Z_k^T) g_{k+1}\|$. If $\rho_{k+1} = 0$, the new gradient has no component outside $\text{range}(Z_k)$ and g_{k+1} is said to be *rejected*. Thus, if $\rho_{k+1} = 0$, then Z_k already provides a basis for $\mathcal{G}_{k+1}$ with $r_{k+1} = r_k$ and $Z_{k+1} = Z_k$. Otherwise, $r_{k+1} = r_k + 1$ and the gradient g_{k+1} is said to be *accepted*. In this case, Z_k gains a new column z_{k+1} defined by the identity $\rho_{k+1} z_{k+1} = (I - Z_k Z_k^T) g_{k+1}$. The calculation of z_{k+1} also provides the r_k-vector $u_k = Z_k^T g_{k+1}$ and the scalar $z_{k+1}^T g_{k+1}$ ($= \rho_{k+1}$), which are the components of the reduced gradient $Z_{k+1}^T g_{k+1}$ for the next iteration. This orthogonalization procedure requires approximately $2nr_k$ flops.

[1] The matrix T_k appears only in the theoretical discussion—it is not needed for computation.

Definition (2.6) of each search direction implies that $p_j \in \mathcal{G}_k$ for all $0 \leq j \leq k$. This leads naturally to another basis for $\mathcal{G}_k$ based on orthogonalizing the search directions $p_0, p_1, \ldots, p_k$. The next lemma implies that Z_k can be defined not only by the accepted gradients, but also by the corresponding search directions.

LEMMA 2.2. *At the start of iteration k, let Z_k denote the matrix obtained by orthogonalizing the gradients $g_0, g_1, \ldots, g_k$ of Broyden's method. Let P_k and G_k denote the matrices of search directions and gradients associated with iterations at which a gradient is accepted. Then there are nonsingular upper-triangular matrices T_k and $\widehat{T}_k$ such that $G_k = Z_k T_k$ and $P_k = Z_k \widehat{T}_k$.*

Proof. Without loss of generality, it is assumed that every gradient is accepted. The proof is by induction on the iteration number k.

The result is true for $k = 0$ because the single column $g_0/\|g_0\|$ of Z_0 is identical to the normalized version of the search direction $p_0 = -g_0/\sigma$.

If the result is true at the start of iteration $k-1$, there exist nonsingular T_{k-1} and $\widehat{T}_{k-1}$ with $G_{k-1} = Z_{k-1}T_{k-1}$ and $P_{k-1} = Z_{k-1}\widehat{T}_{k-1}$. At the start of iteration k, the last column of Z_k satisfies $\rho_k z_k = g_k - Z_{k-1}Z_{k-1}^T g_k$, and

$$G_k = \begin{pmatrix} G_{k-1} & g_k \end{pmatrix} = \begin{pmatrix} Z_{k-1} & z_k \end{pmatrix} \begin{pmatrix} T_{k-1} & Z_{k-1}^T g_k \\ 0 & \rho_k \end{pmatrix} = Z_k T_k. \tag{2.7}$$

The last equality defines T_k, which is nonsingular since $\rho_k \neq 0$. Since $p_k = Z_k Z_k^T p_k$, we find

$$P_k = \begin{pmatrix} P_{k-1} & p_k \end{pmatrix} = \begin{pmatrix} Z_{k-1} & z_k \end{pmatrix} \begin{pmatrix} \widehat{T}_{k-1} & Z_{k-1}^T p_k \\ 0 & z_k^T p_k \end{pmatrix} = Z_k \widehat{T}_k,$$

where the last equality defines $\widehat{T}_k$. The scalar $z_k^T p_k$ is nonzero (see Leonard [16, pp. 94–99][2]), which implies that $\widehat{T}_k$ is nonsingular, and thus the induction is complete. ∎

Lemma 2.2 can be used to show that Z_k provides an orthonormal basis for the span $\mathcal{P}_k$ of *all* search directions $\{p_0, p_1, \ldots, p_k\}$.

THEOREM 2.3. *The subspaces $\mathcal{G}_k$ and $\mathcal{P}_k$ generated by the gradients and search directions of the conventional Broyden method are identical.*

Proof. The definition of each p_j $(0 \leq j \leq k)$ implies that $\mathcal{P}_k \subseteq \mathcal{G}_k$. Lemma 2.2 implies that $\mathcal{G}_k = \text{range}(P_k)$. Since $\text{range}(P_k) \subseteq \mathcal{P}_k$, it follows that $\mathcal{G}_k \subseteq \mathcal{P}_k$. ∎

Given Z_{k+1} and H_{k+1}, the calculation of the search direction for the next iteration requires the Cholesky factor of $Z_{k+1}^T H_{k+1} Z_{k+1}$. This factor can be obtained from R_k in a two-step process that does not require knowledge of H_k. The first step, which is not needed if g_{k+1} is rejected, is to compute the factor R_k'' of $Z_{k+1}^T H_k Z_{k+1}$ (the symbol R_k' is reserved for use in section 3). This step involves adding a row and column to R_k to account for the new last column of Z_{k+1}. It follows from Lemma 2.1 and (2.5) that

$$Z_{k+1}^T H_k Z_{k+1} = \begin{pmatrix} Z_k^T H_k Z_k & Z_k^T H_k z_{k+1} \\ z_{k+1}^T H_k Z_k & z_{k+1}^T H_k z_{k+1} \end{pmatrix} = \begin{pmatrix} Z_k^T H_k Z_k & 0 \\ 0 & \sigma \end{pmatrix},$$

[2]The proof is nontrivial and is omitted for brevity.

giving an expanded block-diagonal factor R_k'' defined by

$$R_k'' = \begin{cases} R_k, & \text{if } r_{k+1} = r_k, \\ \begin{pmatrix} R_k & 0 \\ 0 & \sigma^{1/2} \end{pmatrix}, & \text{if } r_{k+1} = r_k + 1. \end{cases} \tag{2.8}$$

This expansion procedure involves the vectors $v_k = Z_k^T g_k$, $u_k = Z_k^T g_{k+1}$, and $q_k = Z_k^T p_k$, which are stored and updated for efficiency. As both p_k and g_k lie in Range(Z_k), if g_{k+1} is accepted, the vectors $v_k'' = Z_{k+1}^T g_k$ and $q_k'' = Z_{k+1}^T p_k$ are trivially defined from v_k and q_k by appending a zero component (cf. (2.5)). Similarly, the vector $u_k'' = Z_{k+1}^T g_{k+1}$ is formed from u_k and ρ_{k+1}. If g_{k+1} is rejected, then $v_k'' = v_k$, $u_k'' = u_k$ and $q_k'' = q_k$. In either case, v_{k+1} is equal to u_k'' and need not be calculated at the start of iteration $k+1$ (see Algorithm 2.1 below).

The second step of the modification alters R_k'' to reflect the rank-two quasi-Newton update to H_k. This update gives a modified factor $R_{k+1} = \boldsymbol{Broyden}(R_k'', s_k, y_k)$, where $s_k = Z_{k+1}^T(x_{k+1} - x_k) = \alpha_k q_k''$ and $y_k = Z_{k+1}^T(g_{k+1} - g_k) = u_k'' - v_k''$. The work required to compute R_{k+1} depends on the choice of Broyden update and the numerical method used to calculate $\boldsymbol{Broyden}(R_k'', s_k, y_k)$. For the BFGS update, R_{k+1} is the triangular factor associated with the QR factorization of $R_k'' + w_1 w_2^T$, where w_1 and w_2 are given by

$$w_1 = \frac{1}{\|R_k'' s_k\|} R_k'' s_k \quad \text{and} \quad w_2 = \frac{1}{(y_k^T s_k)^{1/2}}\, y_k - \frac{1}{\|R_k'' s_k\|} R_k''^T R_k'' s_k \tag{2.9}$$

(see Goldfarb [12] and Dennis and Schnabel [3]). R_{k+1} can be computed from R_k'' in $4r_k^2 + \mathcal{O}(r_k)$ flops using conventional plane rotations, or in $3r_k^2 + \mathcal{O}(r_k)$ flops using a modified rotation[3] (see Gill et al. [8]). These estimates exclude the cost of forming w_1 and w_2. The vector w_1 is computed in $O(r_k)$ operations from the vector $d_k/\|d_k\|$, where $d_k = -R_k^{-T} v_k$ is the intermediate quantity used in the calculation of q_k (see section 2.1). Similarly, w_2 is obtained in $O(r_k)$ operations using the identity $R_k''^T R_k'' s_k = -\alpha_k v_k''$ implied by (2.6) and the definition of $Z_{k+1}^T g_k$.

2.2. A reduced-Hessian method. We conclude this section by giving a complete reduced-Hessian formulation of a quasi-Newton method from the Broyden class. This method involves two main procedures: an *expand*, which determines Z_{k+1} using the Gram–Schmidt QR process and possibly increases the order of the reduced Hessian by one; and an *update*, which applies a Broyden update directly to the reduced Hessian. For brevity, we use the expression $(Z_{k+1}, R_k'', u_k'', v_k'', q_k'', r_{k+1}) = \boldsymbol{expand}(Z_k, R_k, u_k, v_k, q_k, g_{k+1}, r_k, \sigma)$ to signify the input and output quantities associated with the expand procedure. This statement should be interpreted as the following: Given values of the quantities Z_k, R_k, u_k, v_k, q_k, g_{k+1}, r_k, and σ, the expand procedure computes values of Z_{k+1}, R_k'', u_k'', v_k'', q_k'', r_{k+1}. (Unfortunately, the need to associate quantities used in the algorithm with quantities used in its derivation leads to an algorithm that is more complicated than its computer implementation. In practice, almost all most quantities are updated *in situ*.)

[3] Certain special techniques can be used to reduce this flop count further; see Goldfarb [12].

ALGORITHM 2.1. REDUCED-HESSIAN QUASI-NEWTON METHOD (RH).

Choose x_0 and σ ($\sigma > 0$);
$k = 0$; $r_0 = 1$; $g_0 = \nabla f(x_0)$;
$Z_0 = g_0/\|g_0\|$; $R_0 = \sigma^{1/2}$; $v_0 = \|g_0\|$;
while not converged **do**
  Solve $R_k^T d_k = -v_k$; $R_k q_k = d_k$;
  $p_k = Z_k q_k$;
  Find α_k satisfying the Wolfe conditions (2.4);
  $x_{k+1} = x_k + \alpha_k p_k$; $g_{k+1} = \nabla f(x_k + \alpha_k p_k)$; $u_k = Z_k^T g_{k+1}$;
  $(Z_{k+1}, r_{k+1}, R_k'', u_k'', v_k'', q_k'') = \boldsymbol{expand}(Z_k, r_k, R_k, u_k, v_k, q_k, g_{k+1}, \sigma)$;
  $s_k = \alpha_k q_k''$; $y_k = u_k'' - v_k''$; $R_{k+1} = \boldsymbol{Broyden}(R_k'', s_k, y_k)$;
  $v_{k+1} = u_k''$; $k \leftarrow k + 1$;
end do

In exact arithmetic, Algorithm RH generates the same iterates as its conventional Broyden counterpart, and the methods differ only in the storage needed and the number of operations per iteration. Since v_k is defined as a by-product of the orthogonalization, the computation of p_k involves the solution of two triangular systems and a matrix-vector product, requiring a total of approximately $nr_k + r_k^2$ flops. For the BFGS update, $3r_k^2 + \mathcal{O}(r_k)$ flops are required to update R_k, with the result that Algorithm RH requires approximately $(3r_k + 1)n + 4r_k^2 + \mathcal{O}(r_k)$ flops for each BFGS iteration. As r_k increases, the flop count approaches $7n^2 + \mathcal{O}(n)$. When r_k reaches n, Z_k is full and no more gradients are accepted; only $Z_k^T g_{k+1}$ is computed during the orthogonalization, and the work drops to $6n^2 + \mathcal{O}(n)$. Although H_k is not stored explicitly, it is always implicitly defined by reversing (2.5), i.e.,

$$H_k = Q_k R_Q^T R_Q Q_k^T, \quad \text{where} \quad R_Q = \begin{pmatrix} R_k & 0 \\ 0 & \sigma^{1/2} I_{n-r_k} \end{pmatrix} \tag{2.10}$$

and $Q_k = \begin{pmatrix} Z_k & W_k \end{pmatrix}$.

2.3. Geometric considerations. Next we consider the application of Algorithm RH to the strictly convex quadratic

$$f(x) = c - b^T x + \tfrac{1}{2} x^T A x, \tag{2.11}$$

where c is a scalar, b is an n-vector, and A is an $n \times n$ constant symmetric positive-definite matrix. Suppose that the BFGS update is used, and that each α_k is computed from an exact line search (i.e., α_k minimizes $f(x_k + \alpha p_k)$ with respect to α). Under these circumstances, it can be shown that the $k+1$ columns of Z_k are the normalized gradients $\{g_i/\|g_i\|\}$, and that R_k is upper bidiagonal with nonzero components $r_{ii} = \|g_{i-1}\|/(y_{i-1}^T s_{i-1})^{1/2}$ and $r_{i,i+1} = -\|g_i\|/(y_{i-1}^T s_{i-1})^{1/2}$ for $1 \le i \le k$, and $r_{k+1,k+1} = \sigma^{1/2}$ (see Fenelon [4]). These relations imply that the search directions satisfy

$$p_0 = -\frac{1}{\sigma} g_0, \qquad p_k = -\frac{1}{\sigma} g_k + \beta_{k-1} p_{k-1}, \qquad k \ge 1,$$

with $\beta_{k-1} = \|g_k\|^2/\|g_{k-1}\|^2$. These vectors are parallel to the well-known conjugate-gradient search directions (cf. Corollary 4.2). When used with an exact line search, the search directions and gradients satisfy the relations (i) $p_i^T A p_j = 0$, $i \ne j$; (ii) $g_i^T g_j = 0$, $i \ne j$; and (iii) $g_i^T p_i = -\|g_i\|^2/\sigma$ (see, e.g., Fletcher [5, p. 81] for a proof for

the case $\sigma = 1$). The identities (i), (ii), and (iii) can be used to show that if the search directions are independent, then the local quadratic model $\varphi(p) = g_k^T p + \frac{1}{2} p^T H_k p$ is exact at the start of iteration $n+1$, i.e., $H_{n+1} = A$.

Since the columns of Z_k are the normalized gradients, the BFGS orthogonality relations (ii) imply that a new gradient g_{k+1} can be rejected only if $g_{k+1} = 0$, at which point the algorithm terminates. It follows that the reduced Hessian steadily expands as the iterations proceed. The curvature of the local quadratic model $\varphi(p)$ along any unit vector in Range(W_k) depends only on the choice of H_0 and has no effect on the definition of p_k. Only curvature along directions in Range(Z_k) affects the definition of p_k, and this curvature is completely determined by $Z_k^T H_k Z_k$.

The next lemma implies that $f(x)$ is minimized on a sequence of expanding linear manifolds and that, at the start of iteration k, the curvature of the quadratic model is exact on a certain subspace of dimension k. Let $\mathcal{M}(\mathcal{G}_k)$ denote the linear manifold $\mathcal{M}(\mathcal{G}_k) = \{x_0 + z \mid z \in \mathcal{G}_k\}$ determined by x_0 and $\mathcal{G}_k$.

LEMMA 2.4. *Suppose that the BFGS method with an exact line search is applied to the strictly convex quadratic $f(x)$ (2.11). If $H_0 = \sigma I$, then at the start of iteration k,* (a) *x_k minimizes $f(x)$ on the linear manifold $\mathcal{M}(\mathcal{G}_{k-1})$, and* (b) *the curvature of the quadratic model is exact on the k-dimensional subspace $\mathcal{G}_{k-1}$. Thus, $z^T H_k z = z^T A z$ for all $z \in \mathcal{G}_{k-1}$.*

Proof. Part (a) follows directly from the identity $Z_{k-1}^T g_k = 0$ implied by the orthogonality of the gradients and the special form of Z_{k-1}.

To verify part (b), we write the normalized gradients in terms of the search directions. With Fenelon's form of Z_k and R_k, we find that $Z_k = -P_k D_k R_k$, where $P_k = \begin{pmatrix} p_0 & p_1 & \cdots & p_k \end{pmatrix}$ and D_k is the nonnegative diagonal matrix such that

$$D_k^2 = \sigma^2 \operatorname{diag}\left(\frac{y_0^T s_0}{\|g_0\|^4},\ \frac{y_1^T s_1}{\|g_1\|^4},\ \ldots,\ \frac{y_{k-1}^T s_{k-1}}{\|g_{k-1}\|^4},\ \frac{1}{\sigma \|g_k\|^2} \right).$$

A simple computation using the conjugacy condition (i) above gives the reduced Hessian as $Z_k^T A Z_k = R_k^T D_k P_k^T A P_k D_k R_k = R_k^T \widehat{D} R_k$, where

$$\widehat{D}_k = \sigma^2 \operatorname{diag}\left(\frac{y_0^T s_0}{\|g_0\|^4} p_0^T A p_0,\ \frac{y_1^T s_1}{\|g_1\|^4} p_1^T A p_1,\ \ldots,\ \frac{y_{k-1}^T s_{k-1}}{\|g_{k-1}\|^4} p_{k-1}^T A p_{k-1},\ \frac{p_k^T A p_k}{\sigma \|g_k\|^2} \right).$$

The definition of α_i as the minimizer of $f(x_i + \alpha p_i)$ implies that $\alpha_i = -g_i^T p_i / (p_i^T A p_i)$ and $g_{i+1}^T p_i = 0$. Hence, for all i such that $0 \le i \le k-1$, it follows that

$$y_i^T s_i = \alpha_i y_i^T p_i = -\alpha_i g_i^T p_i = (g_i^T p_i)^2 / p_i^T A p_i.$$

Using these identities with $g_i^T p_i = -\|g_i\|^2/\sigma$ from (iii) above, the expression for $\widehat{D}_k$ simplifies, with $\widehat{D}_k = \operatorname{diag}(I_k,\ 1/\alpha_k)$.

Finally, if Z_k is partitioned so that $Z_k = \begin{pmatrix} Z_{k-1} & g_k/\|g_k\| \end{pmatrix}$, where Z_{k-1} has k columns, then comparison of the leading $k \times k$ principal minors of the matrices $R_k^T R_k$ $(= Z_k^T H_k Z_k)$ and $R_k^T \widehat{D}_k R_k$ $(= Z_k^T A Z_k)$ gives the required identity $Z_{k-1}^T H_k Z_{k-1} = Z_{k-1}^T A Z_{k-1}$. ∎

Part (a) of this result allows us to interpret each new iterate x_{k+1} as "stepping onto" a larger manifold $\mathcal{M}(\mathcal{G}_k)$ such that $\mathcal{M}(\mathcal{G}_{k-1}) \subset \mathcal{M}(\mathcal{G}_k)$. This interpretation also applies when minimizing a general nonlinear function, as long as g_k is accepted for the definition of $\mathcal{G}_k$. (Recall from the proof of Lemma 2.2 that $z_k^T p_k \neq 0$ in this case.)

We say that the curvature along z is *established* if $z^T H_k z = z^T \nabla^2 f(x_k) z$. In particular, under the conditions of Lemma 2.4, the curvature is established at iteration k on all of Range($\mathcal{G}_{k-1}$).

3. Lingering on a manifold. Up to this point we have considered reduced-Hessian methods that generate the same iterates as their Broyden counterparts. Now we expand our discussion to include methods that are not necessarily equivalent to a conventional quasi-Newton method. Our aim is to derive methods with better robustness and efficiency.

When f is a general nonlinear function, the step from x_k to x_{k+1} is unlikely to minimize f on the manifold $\mathcal{M}(\mathcal{G}_k)$. However, in a sequence of iterations in which the gradient is rejected, Z_k remains constant, and the algorithm proceeds to minimize f on the manifold $\mathcal{M}(\mathcal{G}_k)$. In this section, we propose an algorithm in which iterates can remain, or "linger," on a manifold even though new gradients are being accepted. The idea is to linger on a manifold as long as a good reduction in f is being achieved. Lingering has the advantage that the order of the relevant submatrix of the reduced Hessian can be significantly smaller than that of the reduced Hessian itself.

An algorithm that can linger uses one of two alternative search directions: an *RH direction* or a *lingering direction*. An RH direction is defined as in Algorithm RH, i.e., an RH direction lies in $\mathcal{G}_k$ and is computed using the reduced Hessian associated with Z_k. As discussed above, an RH direction defines an x_{k+1} on the manifold $\mathcal{M}(\mathcal{G}_k)$. By contrast, a lingering direction forces x_{k+1} to remain on a manifold $\mathcal{M}(\mathcal{U}_k)$, such that $\mathcal{U}_k \subset \mathcal{G}_k$. Given a point $x_k \in \mathcal{M}(\mathcal{U}_k)$, the next iterate x_{k+1} will also lie on $\mathcal{M}(\mathcal{U}_k)$ as long as $p_k \in \mathcal{U}_k$. Accordingly, an algorithm is said to "linger on $\mathcal{M}(\mathcal{U}_k)$" if the search direction satisfies $p_k \in$ Range(U_k), where the columns of U_k form a basis for $\mathcal{U}_k$. As long as $\mathcal{U}_k$ remains constant and p_k has the form $p_k = U_k p_U$ for some p_U, the iterates x_{k+1}, x_{k+2}, ... will continue to linger on $\mathcal{M}(\mathcal{U}_k)$.

The subspace $\mathcal{U}_k$ and an appropriate basis U_k are defined as follows. At the start of iteration k, an orthonormal basis for $\mathcal{G}_k$ is known such that

$$Z_k = \begin{pmatrix} U_k & Y_k \end{pmatrix}, \tag{3.1}$$

where U_k is an $n \times l_k$ matrix whose columns span the subspace $\mathcal{U}_k$ of all RH directions computed so far, and Y_k corresponds to a certain subset of the accepted gradients defined below. The integer l_k ($0 \le l_k \le r_k$) is known as the *partition parameter* for Z_k. It must be emphasized that the partition (3.1) is defined at *every* iteration, regardless of whether or not lingering occurs. The partition is necessary because quantities computed from U_k and Y_k are used to decide between an RH direction and a lingering direction.

The matrix Z_k is an orthonormal factor of a particular basis for $\mathcal{G}_k$ consisting of both gradients *and* search directions. Let P_k denote the $n \times l_k$ matrix of RH search directions computed so far, and let G_k denote an $n \times (r_k - l_k)$ matrix whose columns are a subset of the accepted gradients. (Note that the definitions of P_k and G_k are different from those of Lemma 2.2.) The matrix Z_k is the orthonormal factor corresponding to the QR factorization of the basis matrix $B_k = \begin{pmatrix} P_k & G_k \end{pmatrix}$, i.e., $\begin{pmatrix} P_k & G_k \end{pmatrix} = Z_k T_k$ for some nonsingular upper-triangular matrix T_k. If T_k is partitioned appropriately, we have

$$B_k = \begin{pmatrix} P_k & G_k \end{pmatrix} = Z_k T_k = \begin{pmatrix} U_k & Y_k \end{pmatrix} \begin{pmatrix} T_U & T_{UY} \\ 0 & T_Y \end{pmatrix}, \tag{3.2}$$

where T_U is an $l_k \times l_k$ upper-triangular matrix. Note that the definition of B_k implies that $\text{Range}(P_k) = \text{Range}(U_k T_U) = \text{Range}(U_k)$, as required.

Although the dimension of U_k remains fixed while iterates linger, the column dimension of Y_k increases as new gradients are accepted into the basis for $\mathcal{G}_k$. While iterates linger, the (as yet) unused approximate curvature along directions in $\text{Range}(Y_k)$ continues to be updated.

Lingering on a manifold ends when an RH direction is chosen and x_{k+1} steps "off" $\mathcal{M}(\mathcal{U}_k)$. Once an RH step is taken, the requirement that U_{k+1} be a basis for the subspace of all previously computed RH directions implies that U_{k+1} must be made to include the component of p_k in $\text{Range}(Y_k)$. This update necessitates a corresponding update to R_k. These updates are discussed in sections 3.3–3.4.

3.1. Definition of the basis. At the start of the first iteration, $r_0 = 1$ and Z_0 is just the normalized gradient $g_0/\|g_0\|$, as in Algorithm RH. The initial partition parameter l_0 is zero, which implies that U_0 is void and Y_0 $(= Z_0)$ is $g_0/\|g_0\|$. Since $\mathcal{U}_0$ is empty, it follows that $p_0 \notin \text{Range}(U_0)$, and an RH step is always taken on the first iteration. At the start of the second iteration, if g_1 is rejected, then $Z_1 = Z_0$, Y_1 is void, $U_1 = Z_1$, $r_1 = 1$, and $l_1 = 1$. On the other hand, if g_1 is accepted, then $Z_1 = (z_0 \;\; z_1)$, where $z_0 = g_0/\|g_0\|$ and z_1 is the normalized component of g_1 orthogonal to z_0. In this case $r_1 = 2$ and $l_1 = 1$, which implies that $U_1 = z_0$ and $Y_1 = z_2$. Using the definitions of z_0 and z_1 it can be verified that

$$B_1 = (p_0 \quad g_1) = (z_0 \quad z_1) \begin{pmatrix} \rho_0 & z_0^T g_1 \\ 0 & \rho_1 \end{pmatrix} = Z_1 T_1,$$

where $\rho_0 = \|p_0\|$.

At the start of the kth iteration, the composition of B_k depends on what has occurred in previous iterations. More precisely, we show that B_k is determined by $B_{k-1} = (P_{k-1} \;\; G_{k-1})$ and two decisions made during iteration $k-1$: (i) the choice of p_{k-1} (i.e., whether it defines an RH or a lingering step), and (ii) the result of the orthogonalization procedure (i.e., whether or not g_k is accepted at the end of iteration $k-1$).

Next, we consider the choice of search direction. Suppose p_{k-1} is an RH direction. Given B_{k-1}, the definition of the new basis B_k involves a *two-stage* procedure in which an intermediate basis B'_{k-1} is defined from matrices P'_{k-1} and G'_{k-1}. The matrix P'_{k-1} is defined by appending p_{k-1} to the right of P_{k-1} to give $P'_{k-1} = (P_{k-1} \;\; p_k)$. The RH direction p_{k-1} must, by definition, satisfy $p_{k-1} \in \text{Range}(P_{k-1} \;\; G_{k-1})$, and hence $(P_k \;\; G_{k-1}) = (P_{k-1} \;\; p_{k-1} \;\; G_{k-1})$ will always have dependent columns. To maintain a linearly independent set of basis vectors, it is therefore necessary to define G'_{k-1} as G_{k-1} with one of its columns removed. When a column is removed from G_{k-1}, the matrices Z_{k-1} and R_{k-1} must be updated. The work needed for this is least if the *last* column is deleted from G_{k-1} (see section 3.3). This procedure corresponds to discarding the most recently computed gradient remaining in G_{k-1}, say g_{k-j} $(k \geq j > 0)$. Note that this deletion procedure is always well defined since G_{k-1} cannot be void when p_{k-1} is an RH direction. Now assume that p_{k-1} is a lingering direction. In this case, we define $P'_{k-1} = P_{k-1}$ and $G'_{k-1} = G_{k-1}$.

The second stage in the calculation of B_k is the definition of P_k and G_k from P'_{k-1} and G'_{k-1} after the orthogonalization procedure of iteration $k-1$. Under the assumption that g_k is accepted, we define $P_k = P'_{k-1}$ and $G_k = (G'_{k-1} \;\; g_k)$. If g_k is not accepted, then $P_k = P'_{k-1}$ and $G_k = G'_{k-1}$.

TABLE 3.1
Example of the composition of P_k and G_k.

k	P_k	G_k	p_k	g_{k+1}
0	*void*	(g_0)	RH	accepted
1	(p_0)	(g_1)	RH	rejected
2	$(p_0\ p_1)$	*void*	lingering	accepted
3	$(p_0\ p_1)$	(g_3)	lingering	accepted
4	$(p_0\ p_1)$	$(g_3\ g_4)$	lingering	accepted
5	$(p_0\ p_1)$	$(g_3\ g_4\ g_5)$	lingering	accepted
6	$(p_0\ p_1)$	$(g_3\ g_4\ g_5\ g_6)$	RH	accepted
7	$(p_0\ p_1\ p_6)$	$(g_3\ g_4\ g_5\ g_7)$	RH	rejected
8	$(p_0\ p_1\ p_6\ p_7)$	$(g_3\ g_4\ g_5)$	RH	rejected
9	$(p_0\ p_1\ p_6\ p_7\ p_8)$	$(g_3\ g_4)$	lingering	rejected
10	$(p_0\ p_1\ p_6\ p_7\ p_8)$	$(g_3\ g_4)$	–	–

These updating rules provide the basis for an algorithm in which P_k can grow at a rate that is commensurate with the rate at which curvature is being established on the manifold $\mathcal{M}(\mathcal{U}_k)$. To illustrate how P_k can change from one iteration to the next, consider the composition of P_k and G_k for the first ten iterations for a function f with at least seven variables. The iterations are summarized in Table 3.1. Each row of the table depicts quantities computed during a given iteration. The first column gives the iteration number, the next two columns give the composition of P_k and G_k, the fourth column indicates the type of direction used, and the last column states whether or not g_{k+1} is accepted after the line search.

If G_k has one more column than G_{k-1}, then p_{k-1} must be a lingering direction and g_k must be accepted (as is the case for $k = 3$, 4, 5, and 6). Similarly, if G_k has one less column than G_{k-1}, then p_{k-1} must be an RH direction and g_k must be rejected ($k = 2$, 8, 9). The matrix G_k will have the same number of columns as G_{k-1} if p_{k-1} is a lingering direction and g_k is rejected ($k = 10$), or if p_{k-1} is an RH direction and g_k is accepted ($k = 1$, 7).

The column dimension of G_k is the number of accepted gradients with indices between 0 and k less the number of RH directions with indices between 0 and $k-1$. In our example, only g_3 and g_4 remain in G_{10}, although every other gradient lies in $\text{Range}\begin{pmatrix} P_{10} & G_{10} \end{pmatrix}$.

To simplify the notation for the remainder of this section, the index k is omitted and overbars indicate quantities associated with iteration $k+1$.

3.2. Definition of the search direction. Next we consider the definition of the lingering and RH search directions, and give a method for choosing between them.

If the rows and columns of R are partitioned to match the partition $Z = \begin{pmatrix} U & Y \end{pmatrix}$, we obtain

$$R = \begin{pmatrix} R_U & R_{UY} \\ 0 & R_Y \end{pmatrix}, \tag{3.3}$$

where R_U is an $l \times l$ upper-triangular matrix. In terms of this partition, the intermediate system $R^T d = -v$ of Algorithm RH is equivalent to two smaller systems

$$R_U^T d_U = -v_U \quad \text{and} \quad R_Y^T d_Y = -(R_{UY}^T d_U + v_Y),$$

where v_U, v_Y, d_U, and d_Y denote subvectors of v and d corresponding to the U- and Y-parts of Z. Note that the vector $v_U = U^T g$ is the reduced gradient associated with

the subspace Range(U). The RH direction minimizes the quadratic model $\varphi(p)$ in the r-dimensional subspace Range(Z), which includes Range(Y). The RH direction is denoted by p^r to distinguish it from the lingering direction p^l defined below.

If the new iterate $\bar{x}$ is to lie on $\mathcal{M}(\mathcal{U})$, the search direction must lie in Range(U). The obvious choice for p^l is the unique minimizer of the local quadratic model $\varphi(p) = g^T p + \frac{1}{2} p^T H p$ in Range(U). This minimizer is given by $-U(U^T H U)^{-1} U^T g$, from which it follows that p^l can be computed as $p^l = U R_U^{-1} d_U$.

The choice between p^r and p^l is based on comparing $\varphi(p^r)$ with $\varphi(p^l)$, where the quadratic model $\varphi(p)$ estimates $f(x+p) - f(x)$, the change in the objective. From the definitions of p^r and p^l, we have

$$\varphi(p^r) = -\tfrac{1}{2}\|d\|^2 \quad \text{and} \quad \varphi(p^l) = -\tfrac{1}{2}\|d_U\|^2.$$

These predictions are attained if f is a convex quadratic and an exact line search is used. In this case the gradients are mutually orthogonal (see section 2.3), both v_U and d_U are zero, and the only way to decrease f is to step off $\mathcal{M}(\mathcal{U})$ using p^r.

On the other hand, when minimizing a general nonlinear function with an inexact line search, it is possible that $\|d_U\| \approx \|d\|$, and nearly all of the reduction in the quadratic model is obtained on $\mathcal{M}(\mathcal{U})$. In this event, little is lost by forcing the iterates to remain on $\mathcal{M}(\mathcal{U})$. In addition, lingering can be used to ensure that the reduced gradient v_U is "sufficiently small," and may help to further establish the curvature on $\mathcal{U}$. In this sense, lingering is a way of forcing Broyden's method to perform on a general nonlinear function as it does on a quadratic.

As noted by Fenelon [4, p. 72], it can be inefficient to remain on $\mathcal{M}(\mathcal{U})$ until the reduced gradient v_U is zero. Instead, iterates are allowed to linger until the predicted reduction corresponding to a step moving off of $\mathcal{M}(\mathcal{U})$ is significantly better than the predicted reduction for a step that lingers. In particular, a step off of $\mathcal{M}(\mathcal{U})$ is taken if $\|d_U\|^2 \le \tau \|d\|^2$, where τ is a preassigned constant such that $\frac{1}{2} < \tau < 1$.

The following simple argument shows that if $p = p^r$ is selected when the condition $\|d_U\|^2 \le \tau\|d\|^2$ is satisfied, then the next iterate steps off of $\mathcal{M}(\mathcal{U})$. If the U- and Y-parts of q are denoted by q_U and q_Y, respectively, the partitioned form of $Rq = d$ is given by

$$R_Y q_Y = d_Y \quad \text{and} \quad R_U q_U = d_U - R_{UY} q_Y. \tag{3.4}$$

Written in terms of p_U and p_Y, the search direction satisfies $p = U q_U + Y q_Y$. The inequality $(1-\tau)\|d_U\|^2 \le \tau \|d_Y\|^2$ implies that both d and d_Y are nonzero, and it follows from (3.4) and the nonsingularity of R_Y that q_Y is nonzero. Hence, $Y q_Y$ is also nonzero and $\bar{x} = x + \alpha p^r$ steps off of $\mathcal{M}(\mathcal{U})$.

3.3. Updating *Z*. Let P, G, T, and Z denote matrices associated with the orthogonal factorization (3.2) at the start of an iteration. In section 3.1 it was shown that the basis undergoes two (possibly trivial) changes during an iteration, i.e., $B = (\,P \;\; G\,) \to B' = (\,P' \;\; G'\,) \to \bar{B} = (\,\bar{P} \;\; \bar{G}\,)$.

The first change to Z involves updating the orthogonal factorization $(\,P \;\; G\,) = ZT = (\,U \;\; Y\,)T$ to obtain $(\,P' \;\; G'\,) = Z'T' = (\,U' \;\; Y'\,)T'$, associated with the intermediate basis B'. The update depends on the choice of p. If p is the lingering direction p^l, we have the trivial case $T' = T$, $U' = U$, and $Y' = Y$. If p is the RH direction p^r, then $P' = (\,P \;\; p\,)$ and the resulting effect on U and Y must be calculated.

Introducing p on both sides of the decomposition $\begin{pmatrix} P & G \end{pmatrix} = ZT$ yields

$$\begin{pmatrix} P & p & G \end{pmatrix} = \begin{pmatrix} U & Y \end{pmatrix} \begin{pmatrix} T_U & q_U & T_{UY} \\ 0 & q_Y & T_Y \end{pmatrix},$$

where $q = Z^T p$ and q_U and q_Y denote the components of q corresponding to the U- and Y-parts of Z. The left-hand side can be repartitioned as $(\, P' \;\; G' \;\; \underline{g}\,)$, where $P' = \begin{pmatrix} P & p \end{pmatrix}$ and $\underline{g}$ is the most recently accepted gradient remaining in the basis. Let S denote an $r \times r$ orthogonal upper-Hessenberg matrix constructed such that

$$S = \begin{pmatrix} I_l & 0 \\ 0 & S_Y \end{pmatrix} \quad \text{and} \quad Sq = \begin{pmatrix} q_U \\ \|q_Y\| e_1 \end{pmatrix}.$$

It follows that

$$\begin{pmatrix} P' & G' & \underline{g} \end{pmatrix} = \begin{pmatrix} U & Y S_Y^T \end{pmatrix} T_S, \quad \text{where} \quad T_S = \begin{pmatrix} T_U & q_U & T_{UY} \\ 0 & S_Y q_Y & S_Y T_Y \end{pmatrix}. \tag{3.5}$$

The shape of S implies that the $(r-l) \times (r-l+1)$ matrix $\begin{pmatrix} S_Y q_Y & S_Y T_Y \end{pmatrix}$ is upper-Hessenberg, and the $r \times (r+1)$ matrix T_S is upper-trapezoidal. Deleting the last column from each side of the identity (3.5) gives the required factorization. In particular, $U' = (\, U \;\; Y S_Y^T e_1\,)$, $Y' = (\, Y S_Y^T e_2 \;\; Y S_Y^T e_3 \;\; \cdots \;\; Y S_Y^T e_{r-l}\,)$, $Z' = (\, U' \;\; Y'\,)$, and $T' = T_S E_r$, where E_r denotes the matrix of first r columns of I_{r+1}.

The matrix S is defined as a product of plane rotations and need not be stored explicitly. One choice of S that uses symmetric Givens matrices instead of plane rotations is given by Daniel et al. [2] in the context of inserting a column into a QR factorization. As S can be generated entirely from q_Y, the matrix T need not be stored.

If $l < r-1$, then the modification of U and Y requires approximately $3(r-l-1)n$ flops (see Daniel et al. [2]). If $l = r-1$, then no work is required since the columns of $Z = \begin{pmatrix} U & Y \end{pmatrix}$ are already an orthonormal basis for $\begin{pmatrix} P & p \end{pmatrix}$. (The argument is similar to that given in Lemma 2.2, although here q_Y is nonzero according to the reasoning given at the end of section 3.2.)

The second stage in updating Z is to compensate for the change from B' to the final basis $\bar{B}$. After the line search, if the new gradient $\bar{g}$ is rejected, then we have the trivial case $\bar{T} = T'$, $\bar{U} = U'$, and $\bar{Y} = Y'$. If $\bar{g}$ is accepted, then $\bar{\rho}\bar{z} = \bar{g} - Z'Z'^T\bar{g}$ defines the normalized component of $\bar{g}$ outside Range(Z') (see section 2.1). In this case, the identity

$$\begin{pmatrix} P' & G' & \bar{g} \end{pmatrix} = \begin{pmatrix} Z' & \bar{z} \end{pmatrix} \begin{pmatrix} T' & Z'^T\bar{g} \\ 0 & \bar{\rho} \end{pmatrix}$$

implies that $\bar{P} = P'$, $\bar{G} = \begin{pmatrix} G' & \bar{g} \end{pmatrix}$, $\bar{U} = U'$, $\bar{Y} = (\, Y' \;\; \bar{z}\,)$, and $\bar{T}$ is T' augmented by the column $\bar{Z}^T\bar{g}$. In both cases, we define $\bar{Z} = (\, \bar{U} \;\; \bar{Y}\,)$.

3.4. Updating *R*. When Z is updated to include the new RH direction, the new reduced Hessian is $Z'^T H Z' = S Z^T H Z S^T = S R^T R S^T$, where H is given by (2.10). The $(2,2)$ block of RS^T is $R_Y S_Y^T$, which can be restored to upper-triangular form using a suitable sequence of plane rotations S' applied on the left of R. This

results in $R_Y S_Y^T$ being premultiplied by an orthogonal matrix S'_Y such that $S'_Y R_Y S_Y^T$ is upper-triangular. The Cholesky factor of $Z'^T H Z'$ is then

$$R' = S' R S^T = \begin{pmatrix} R_U & R_{UY} S_Y^T \\ 0 & S'_Y R_Y S_Y^T \end{pmatrix} = \begin{pmatrix} R'_{U'} & R'_{U'Y'} \\ 0 & R'_{Y'} \end{pmatrix},$$

where $R'_{U'}$ and $R'_{Y'}$ are upper-triangular matrices of order $l+1$ and $r-l-1$. For more details, see Leonard [16, p. 40]. The calculation of R' simplifies considerably if the BFGS update is used (see section 3.7).

It remains to update R' to reflect the second stage of the basis change: $B' = (\, P' \;\; G' \,) \rightarrow \bar{B} = (\, \bar{P} \;\; \bar{G} \,)$, which corresponds to the orthogonalization of the new gradient. If R'' denotes the updated factor, then R'' is obtained from R' by adding a scaled unit row and column, as in (2.8).

3.5. Updating related quantities. After the new gradient has been orthogonalized, the vectors $u'' = \bar{Z}^T \bar{g}$, $v'' = \bar{Z}^T g$, and $q'' = \bar{Z}^T p$ are used to define the quasi-Newton update $\bar{R} = \textbf{\textit{Broyden}}(R'', s, y)$ with $s = \alpha q''$ and $y = u'' - v''$. The vector u'' is computed as a by-product of the orthogonalization, as in Algorithm RH. The vectors v'' and q'' can be computed from v and q using intermediate vectors $v' = Z'^T g$ and $q' = Z'^T p$ in conjunction with the two-stage update to B. If p is a lingering direction, then $v' = v$ and $q' = q$. Otherwise, the definition of Z' implies that

$$v' = Z'^T g = S Z^T g = S v = \begin{pmatrix} v_U \\ S_Y v_Y \end{pmatrix},$$

which can be computed efficiently by applying the plane rotations of S as they are generated. Similarly, the U'- and Y'-portions of q' are $q'_{U'} = (q_U, \|q_Y\|)^T$ and $q'_{Y'} = 0$, since $S_Y q_Y = \|q_Y\| e_1$. These expressions provide a cheaper alternative to computing the RH search direction as $p = U q_U + Y q_Y$. With this alternative, U is modified as soon as q_U and q_Y are known, and p is computed from the expression $p = U' q'_{U'}$.

Once v' and q' are known, v'' and q'' are found from v' and q' during the orthogonalization procedure as in Algorithm RH.

3.6. A reduced-Hessian method with lingering. We summarize the results of this section by describing a complete reduced-Hessian method with lingering. As in Algorithm RH of section 2.2, certain calculations are represented schematically as functions with input and output arguments. The first stage of the basis update can be viewed as swapping the new RH direction with the most recently accepted gradient remaining in B_k. Accordingly, the modification of Z_k (and hence U_k and Y_k) and related quantities is represented by $(Z'_k, R'_k, q'_k, v'_k) = \textbf{\textit{swap}}(Z_k, R_k, q_k, v_k)$.

ALGORITHM 3.1. REDUCED-HESSIAN METHOD WITH LINGERING (RHL).

Choose x_0 and σ ($\sigma > 0$);
$k = 0$; $r_0 = 1$; $l_0 = 0$; $g_0 = \nabla f(x_0)$;
$Z_0 = g_0/\|g_0\|$; $R_0 = \sigma^{1/2}$; $v_0 = \|g_0\|$;
while not converged **do**
  Partition R_k as R_U, R_Y, and R_{UY}; Partition v_k as v_U and v_Y;
  Solve $R_U^T d_U = -v_U$; $R_Y^T d_Y = -(R_{UY}^T d_U + v_Y)$;
  if $\|d_U\|^2 > \tau \|d\|^2$ **then**
    Solve $R_U q_U = d_U$; $q_Y = 0$;

$Z'_k = Z_k$; $R'_k = R_k$; $q'_k = q_k$; $v'_k = v_k$;
$l'_k = l_k$;
else
Solve $R_Y q_Y = d_Y$; $R_U q_U = d_U - R_{UY} q_Y$;
$(Z'_k, R'_k, q'_k, v'_k) = \boldsymbol{swap}(Z_k, R_k, q_k, v_k)$;
$l'_k = l_k + 1$;
end if
$p_k = U'_k q'_{U'}$; $l_{k+1} = l'_k$;
Find α_k satisfying the Wolfe conditions (2.4);
$x_{k+1} = x_k + \alpha_k p_k$; $g_{k+1} = \nabla f(x_k + \alpha_k p_k)$; $u_k = Z_k^T g_{k+1}$;
$(Z_{k+1}, r_{k+1}, R''_k, u''_k, v''_k, q''_k) = \boldsymbol{expand}(Z'_k, r_k, R'_k, u_k, v'_k, q'_k, g_{k+1}, \sigma)$;
$s_k = \alpha_k q''_k$; $y_k = u''_k - v''_k$; $R_{k+1} = \boldsymbol{Broyden}(R''_k, s_k, y_k)$;
$v_{k+1} = u''_k$; $k \leftarrow k + 1$;
end do

As in Algorithm RH, no new gradients are accepted once r_k reaches n. If the test $\|d_U\|^2 \leq \tau \|d\|^2$ is satisfied every iteration, Algorithm RHL generates the same sequence of iterates as Algorithm RH. In this case, every iteration starts with $l_k = r_k$ or $l_k = r_k - 1$. If $l_k = r_k$, the previous gradient g_k was rejected and both algorithms compute a lingering direction. Otherwise, if $l_k = r_k - 1$, then g_k was accepted and Y_k must have just one column. Once the RH direction is computed, the swap procedure amounts to moving the partition of Z_k so that the Y-part becomes void. It follows that if only RH directions are chosen, the partition of Z_k is used only to decide if p_k is an RH or lingering direction.

3.7. The BFGS update. If the BFGS update is used, the block structure (3.3) of R simplifies to the extent that R_Y is always $\sigma^{1/2} I_{r-l}$. This can be shown using two results. The first describes the effect of the BFGS update on R when $s \in \text{Range}(U)$.

LEMMA 3.1. *Let R denote an $r \times r$ nonsingular upper-triangular matrix partitioned as in* (3.3). *Let y and s be r-vectors such that $y^T s > 0$. If the Y-components of s are zero, then the update $\bar{R} = \boldsymbol{BFGS}(R, s, y)$ does not alter the* $(2,2)$ *block of R (i.e., $\bar{R}_Y = R_Y$). Moreover, $\bar{R}_U$ and $\bar{R}_{UY}$ are independent of R_Y.*

Proof. The result follows from the definition (2.9) of the rank-one BFGS update to R (see Leonard [16, pp. 13–15] for the first part). ∎

The next lemma considers the cumulative effect of Algorithm RHL on the block structure of R.

LEMMA 3.2. *Assume that Algorithm RHRL is used with the BFGS update, and that Z is partitioned as $Z = \begin{pmatrix} U & Y \end{pmatrix}$. Then there exist orthogonal matrices S and S' for the basis update such that, at the start of every iteration, R has the form*

$$\begin{pmatrix} R_U & R_{UY} \\ 0 & R_Y \end{pmatrix} \quad \text{with} \quad R_Y = \sigma^{1/2} I_{r-l}. \tag{3.6}$$

Proof. The proof is by induction. The result holds at the start of the first iteration since $r_0 = 1$, $l_0 = 0$, and $R_Y = \sigma^{1/2}$. Assume that the result holds at the start of iteration k.

If the partition parameter is increased, the columns of Y are modified and the $(2,2)$ block of R' satisfies $R'_Y = S'_Y R_Y S_Y^T = \sigma S'_Y S_Y^T$. If $S' = S$, then $R'_Y = \sigma I_{r-l}$. If the partition parameter does not change, then $R' = R$ and $R'_Y = \sigma I_{r-l}$ trivially.

The repartition resulting from the change in l gives $\sigma^{1/2} I_{r-\bar{l}}$ in the $(2,2)$ block, and it follows that $R'_{Y'}$ has the required form prior to the line search. Note that $R'_{Y'}$ is void if either R_Y is void (i.e., $l = r$) or l was increased to r (giving $\bar{l} = r$).

The expansion procedure may add a scaled unit row and column to R'. In either case, R'' can be partitioned to match $\bar{U}$ and $\bar{Y}$ as

$$R'' = \begin{pmatrix} R''_{\bar{U}} & R''_{\bar{U}\bar{Y}} \\ 0 & R''_{\bar{Y}} \end{pmatrix}.$$

It follows that $R''_{\bar{Y}} = \sigma^{1/2} I_{\bar{r}-\bar{l}}$.

Whatever the choice of search direction, q'' is of the form $q'' = (q''_{\bar{U}}, 0)^T$, where $q''_{\bar{U}}$ is an $\bar{l}$-vector. Thus, R'' and s satisfy the conditions of Lemma 3.1, and $\bar{R} = \boldsymbol{BFGS}(R'', s, y)$ has the required structure. □

If, in the BFGS case, instead of defining $S' = S$, we update R_Y according to the procedure of section 3.3, then the updated matrix will be of the form $R_Y = \sigma^{1/2} \tilde{I}_{r-l}$, where $\tilde{I}_{r-l}$ is a diagonal matrix of plus or minus ones. The purpose of Lemma 3.2 is to show that it is *unnecessary* to apply S^T and S' to R_Y when RHL is used with the BFGS update. Instead, S_Y^T need only be applied to R_{UY}, at a cost of $3(r-l-1)l$ flops.

3.8. Operation count for RHL with the BFGS update. The number of operations for an iteration of the BFGS version of Algorithm RHL will depend on the type of search direction selected. If a lingering direction is used, the vector $R^T R q$ will be different from $-v$, and the vector v cannot be substituted for the matrix-vector product in (2.9). However, in this case we have

$$R^T R q = \begin{pmatrix} R_U^T R_U q_U \\ R_{UY}^T R_U q_U \end{pmatrix} = \begin{pmatrix} -v_U \\ R_{UY}^T R_U q_U \end{pmatrix},$$

which requires only $R_{UY}^T R_U q_U$ to be computed explicitly.

Whichever search direction is used, the vector s has $\bar{r} - \bar{l}$ trailing zeros (see (2.9)), and the cost of applying the BFGS update drops to $6\bar{r}\bar{l} - 3\bar{l}^2$ flops. It follows that iterations involving a lingering direction require $(2r+l+1)n + \frac{1}{2}r^2 + 7rl - \frac{7}{2}l^2 + \mathcal{O}(r) + \mathcal{O}(l)$ flops. If $l = r$, the work is commensurate with that of Algorithm RH. If an RH step is taken, an additional n flops are required because p is a linear combination of $l+1$ n-vectors instead of l n-vectors. In this case, $3(r-l-1)(l+n)$ flops are required to update Z and R using the method of sections 3.3–3.4.

4. Modifying approximate curvature. The choice of H_0 can greatly influence the practical performance of quasi-Newton methods. The usual choice $H_0 = \sigma I$ ($\sigma > 0$) can result in many iterations and function evaluations—especially if $\nabla^2 f(x^*)$ is ill-conditioned (see, e.g., Powell [25] and Siegel [28]). This is sometimes associated with "stalling" of the iterates, a phenomenon that can greatly increase the overall cpu time for solution (or termination).

To date, the principal modifications of conventional quasi-Newton methods have involved *scaling* all or part of the approximate Hessian. Several scaling methods have appeared in the literature. In the *self-scaling variable metric (SSVM)* method of Oren and Luenberger [24], H_k is multiplied by a positive scalar prior to application of the Broyden update. The *conjugate-direction scaling* method of Siegel [28] scales columns of a certain conjugate-direction factorization of H_k^{-1}. This scheme, which

is a refinement of a method of Powell [25], has been shown to be globally and q-superlinearly convergent. In what follows, Siegel's method will be referred to as Algorithm CDS. Finally, Lalee and Nocedal [15] have proposed an algorithm that scales columns of a lower-Hessenberg factor of H_k. This method will be referred to as Algorithm ACS, which stands for *automatic column scaling*.

Here, scaling takes the form of resetting certain diagonal elements of the Cholesky factor of the reduced-Hessian. The structure of the transformed Hessian $Q_k^T H_k Q_k$ (2.5) reveals the influence of H_0 on the approximate Hessian. For example, the initial Hessian scale factor σ represents the approximate curvature along all unit directions in $\mathcal{G}_k^{\perp}$ (see Lemma 2.1). Inefficiencies resulting from poor choices of H_0 may be alleviated by maintaining a current estimate σ_k of the approximate curvature in $\mathcal{G}_k^{\perp}$. At the end of each iteration, the new estimate σ_{k+1} replaces σ_k in the transformed Hessian wherever this can be done without endangering its positive definiteness. This replacement has the effect of *reinitializing* approximate curvature along all directions in $\mathcal{G}_k^{\perp}$, and along certain directions in $\mathcal{G}_k$. In the next section, an algorithm of this type is introduced as a generalization of Algorithm RHL.

4.1. Reinitialization combined with lingering. In this section we extend the BFGS version of Algorithm RHL so that approximate curvature is modified in a subspace of dimension $n - \bar{l}$ immediately following the BFGS update. We choose to emphasize the BFGS method because the diagonal structure $\bar{R}_{\bar{Y}} = \sigma^{1/2} I_{\bar{r}-\bar{l}}$ of the $(2,2)$ block of the BFGS Cholesky factor reveals the main influence of H_0 on the approximate Hessian. In this case, the initial approximate curvature along unit directions in Range$(\bar{Y})$ is explicit and easily reinitialized. The approximate curvature along directions in Range$(\bar{U})$ is considered to be sufficiently established (in the sense of Lemma 2.4) and hence the corresponding part of the reduced Hessian is unaltered.

Reinitialization is not hard to achieve in comparison to some scaling procedures previously proposed. Reinitialization simply involves replacing the factor

$$R''' = \begin{pmatrix} R'''_{\bar{U}} & R'''_{\bar{U}\bar{Y}} \\ 0 & \sigma^{1/2} I_{\bar{r}-\bar{l}} \end{pmatrix} \quad \text{by} \quad \bar{R} = \begin{pmatrix} R'''_{\bar{U}} & R'''_{\bar{U}\bar{Y}} \\ 0 & \bar{\sigma}^{1/2} I_{\bar{r}-\bar{l}} \end{pmatrix},$$

where the matrix R''' is the final factor obtained in an iteration of Algorithm RHL. The corresponding effect on the $(2,2)$ block of the reduced Hessian is to replace the term $\sigma I_{\bar{r}-\bar{l}}$ by $\bar{\sigma} I_{\bar{r}-\bar{l}}$.

This reinitialization exploits the special structure of R''' resulting from the lingering scheme. The resulting method may be compared to Liu and Nocedal's limited-memory L-BFGS method [17]. In this case, the BFGS *inverse* Hessian is defined as the last of a sequence of auxiliary inverse Hessians generated implicitly from σI and a set of vector pairs (δ_k, γ_k) (see (2.3)). This form allows σI to be reinitialized at every iteration (in which case, every auxiliary inverse Hessian is changed). The fact that the rank-two terms are not summed explicitly is crucial. If the inverse Hessian were to be stored elementwise, then any reinitialization that adds a (possibly negative-definite) diagonal $(\bar{\sigma} - \sigma)I$ would leave all the auxiliary approximations unchanged except the first, and thereby define a potentially indefinite approximation. In the reduced-Hessian formulation, it is possible to maintain an elementwise approximation *and* reinitialize unestablished curvature without risk of indefiniteness. The diagonal form of $\bar{R}_{\bar{Y}}$ means that σ occurs as an explicit modifiable term in the expression for the curvature along directions in Range$(\bar{Y})$. This term can be safely reset to any positive number $\bar{\sigma}$.

It remains to define an appropriate value for $\bar{\sigma}$. We consider four alternatives that have been effective in practice. The first two are the simple choices:

$$\sigma_{k+1}^{R0} = 1 \quad \text{and} \quad \sigma_{k+1}^{R1} = \frac{y_0^T y_0}{y_0^T s_0} \tag{4.1}$$

(see Shanno and Phua [26] for a discussion of σ_{k+1}^{R1}). The third alternative is related to the scaling parameters used in Algorithm CDS (see Siegel [28]). It is defined in terms of a scalar γ_i and satisfies

$$\sigma_{k+1}^{R2} = \min_{0 \le i \le k} \{\gamma_i\}, \quad \text{where} \quad \gamma_i = \frac{y_i^T s_i}{\|s_i\|^2}. \tag{4.2}$$

The fourth alternative is the inverse of the value suggested by Liu and Nocedal [17] for use in the limited-memory BFGS method (see Nocedal [23]). For this option, we define

$$\sigma_{k+1}^{R3} = \frac{y_k^T y_k}{y_k^T s_k}. \tag{4.3}$$

Reinitialization is represented schematically as $\bar{R} = \boldsymbol{reinitialize}(R''', \bar{\sigma})$ in the algorithm below.

ALGORITHM 4.1. REDUCED-HESSIAN METHOD WITH REINITIALIZATION AND LINGERING (RHRL).

Choose x_0 and σ_0 ($\sigma_0 > 0$);
$k = 0$; $r_0 = 1$; $l_0 = 0$; $g_0 = \nabla f(x_0)$;
$Z_0 = g_0/\|g_0\|$; $R_0 = \sigma_0^{1/2}$; $v_0 = \|g_0\|$;
while not converged **do**
  Partition R_k as R_U, R_Y and R_{UY}; Partition v_k as v_U and v_Y;
  Solve $R_U^T d_U = -v_U$; $R_Y^T d_Y = -(R_{UY}^T d_U + v_Y)$;
  if $\|d_U\|^2 > \tau \|d\|^2$ **then**
    Solve $R_U q_U = d_U$; $q_Y = 0$;
    $Z_k' = Z_k$; $R_k' = R_k$; $q_k' = q_k$; $v_k' = v_k$;
    $l_k' = l_k$;
  else
    Solve $R_Y q_Y = d_Y$; $R_U q_U = d_U - R_{UY} q_Y$;
    $(Z_k', R_k', q_k', v_k') = \boldsymbol{swap}(Z_k, R_k, q_k, v_k)$;
    $l_k' = l_k + 1$;
  end if
  $p_k = U_k' q_{U'}'$; $l_{k+1} = l_k'$;
  Find α_k satisfying the Wolfe conditions (2.4);
  $x_{k+1} = x_k + \alpha_k p_k$; $g_{k+1} = \nabla f(x_k + \alpha_k p_k)$; $u_k = Z_k^T g_{k+1}$;
  $(Z_{k+1}, r_{k+1}, R_k'', u_k'', v_k'', q_k'') = \boldsymbol{expand}(Z_k', r_k, R_k', u_k, v_k', q_k', g_{k+1}, \sigma_k)$;
  $s_k = \alpha_k q_k''$; $y_k = u_k'' - v_k''$; $R_k''' = \boldsymbol{BFGS}(R_k'', s_k, y_k)$;
  Compute σ_{k+1}; $R_{k+1} = \boldsymbol{reinitialize}(R_k''', \sigma_{k+1})$;
  $v_{k+1} = u_k''$; $k \leftarrow k + 1$;
end do

Other than the addition of the ***reinitialize*** procedure, Algorithm RHRL differs from RHL only in the specific use of the BFGS update.

Reinitialization can be applied directly to Algorithm RH by redefining σ *before* the ***expand*** and ***Broyden*** procedures. When the BFGS update is used and R_k expands, the last diagonal of R_k'' is unaltered and is independent of the remainder of $\bar{R}$ (see section 3.7). In this case, the last diagonal can be redefined either before or after the update. This option is also available for RHRL, but reinitialization is done after the BFGS update to simplify the proof of Theorem 4.3. (The trailing columns of the conjugate-direction matrix are scaled *after* the BFGS update in Algorithm CDS [28].)

4.2. Algorithm RHRL applied to a quadratic. Consider the strictly convex quadratic function $f(x) = c - b^T x + \frac{1}{2}x^T A x$ of (2.11). The next theorem extends Fenelon's quadratic termination results for Algorithm RH to Algorithm RHRL (see section 2.3). In the statement of the theorem, r_{ij} denotes the (i,j)th component of R_k. For details of the proof, see Leonard [16, pp. 58–61].

THEOREM 4.1. *Consider Algorithm RHRL applied with an exact line search and $\sigma_0 = 1$ to minimize the quadratic $f(x)$ of* (2.11)*. Then, at the start of iteration k, the rank of R_k is $r_k = k+1$, the partition parameter is $l_k = k$, and Z_k satisfies $Z_k = (\,U_k \;\; Y_k\,)$, where the columns of U_k are the normalized gradients $\{g_i/\|g_i\|\}$, $1 \le i \le k-1$, and $Y_k = g_k/\|g_k\|$. Moreover, the upper-triangular matrix R_k is upper bi-diagonal with $R^k_{UY} = -\|g_k\|/(y^T_{k-1}s_{k-1})e_k$ and $R^k_Y = \sigma_k^{1/2}$. The nonzero components of R_k in R^k_U satisfy $r_{ii} = \|g_{i-1}\|/(y^T_{i-1}s_{i-1})^{1/2}$ and $r_{i,i+1} = -\|g_i\|/(y^T_{i-1}s_{i-1})^{1/2}$ for $1 \le i \le k$. Furthermore, the search directions satisfy*

$$p_0 = -g_0; \qquad p_k = -\frac{1}{\sigma_k}g_k + \beta_{k-1}p_{k-1}, \quad \beta_{k-1} = \frac{\sigma_{k-1}}{\sigma_k}\frac{\|g_k\|^2}{\|g_{k-1}\|^2}, \quad k \ge 1. \qquad \square$$

COROLLARY 4.2. *If Algorithm RHRL is applied with an exact line search to minimize the quadratic $f(x)$ of* (2.11)*, and $\sigma_0 = 1$, then the minimizer will be found in at most n iterations.*

Proof. We show by induction that the search directions are parallel to the conjugate-gradient directions $\{d_k\}$. Specifically, $\sigma_k p_k = d_k$ for all k. This is true for $k = 0$ since $\sigma_0 p_0 = -g_0 = d_0$. Assume that $\sigma_{k-1}p_{k-1} = d_{k-1}$. Using Theorem 4.1 and the inductive hypothesis, we find

$$\sigma_k p_k = -g_k + \sigma_{k-1}\frac{\|g_k\|^2}{\|g_{k-1}\|^2}p_{k-1} = -g_k + \frac{\|g_k\|^2}{\|g_{k-1}\|^2}d_{k-1} = d_k,$$

which completes the induction. Since the conjugate-gradient method has quadratic termination under the assumptions of the theorem, Algorithm RHRL must also have this property. $\square$

4.3. An equivalence with conjugate-direction scaling. The next theorem, proved by Leonard [16, pp. 62–77], states that under certain conditions, Algorithm RHRL generates the same iterates as the CDS algorithm of Siegel [28].

THEOREM 4.3. *Suppose that Algorithm* RHRL *and Algorithm* CDS *are used to find a local minimizer of a twice-continuously differentiable function $f : \mathbb{R}^n \to \mathbb{R}$. If both algorithms start from the same point and use the same line search, and if* RHRL *uses $\sigma_0 = 1$, $\sigma_k = \sigma_k^{R2}$, $\tau = \frac{10}{11}$, then the algorithms generate the same sequence of iterates.* $\square$

Despite this equivalence, we emphasize that RHRL and CDS are *not the same method*. First, the stated equivalence concerns CDS and one instance of RHRL, so

RHRL may be considered as a generalization of CDS. Second, the dimensions of the matrices required and the computation times differ substantially for the two methods. CDS has a 33% advantage with respect to storage, since RHRL requires $\frac{3}{2}n^2$ elements for Z_k and R_k, assuming that they grow to full size. However, RHRL requires substantially lower cpu times in practice—principally because of the more efficient calculation of p_k when k is small relative to n (see section 6.5).

The last result of this section gives convergence properties of Algorithm RHRL when applied to strictly convex functions.

COROLLARY 4.4. *Let $f : \mathbb{R}^n \to \mathbb{R}$ denote a strictly convex, twice-continuously differentiable function. Moreover, assume that $\nabla^2 f(x)$ is Lipschitz continuous with $\|\nabla^2 f(x)^{-1}\|$ bounded above for all x in the level set of $f(x_0)$. If Algorithm RHRL with $\sigma_0 = 1$, $\sigma_k = \sigma_k^{R2}$, $\tau = \frac{10}{11}$, and a Wolfe line search is used to minimize f, then convergence is global and q-superlinear.*

Proof. Since the conjugate-direction scaling algorithm has these convergence properties (see Siegel [28]), the proof is immediate from Theorem 4.3. ∎

5. Implementation details. In this section, we discuss the implementation of Algorithm RHRL. Numerical results are given in section 6.

5.1. Reorthogonalization. In exact arithmetic, a gradient is accepted into the basis $B_k' = \begin{pmatrix} P_k' & G_k' \end{pmatrix}$ if $\rho_{k+1} > 0$, where ρ_{k+1} is the two-norm of $(I - Z_k' Z_k'^T) g_{k+1}$. This ensures that the basis vectors are linearly independent, and that the implicitly defined matrix T_k' (3.2) is nonsingular. When ρ_{k+1} is computed in finite precision, gradients with small (but nonzero) ρ_{k+1} are rejected to discourage $\{T_k\}$ from becoming too ill-conditioned. In practice, an accepted gradient must satisfy $\rho_{k+1} \geq \epsilon \|g_{k+1}\|$, where ϵ is a preassigned positive constant. In the numerical results presented in section 6, ϵ was set at 10^{-4}.

Even when ϵ is large relative to the machine precision, Gram–Schmidt orthogonalization is unstable (see Golub and Van Loan [13, p. 218]). Two of the best known algorithms for stabilizing the process are modified Gram–Schmidt and Gram–Schmidt with reorthogonalization (see Golub and Van Loan [13, p. 218] and Daniel et al. [2]). We have used Gram–Schmidt with reorthogonalization in our implementation. Each reorthogonalization requires an additional $2nr_k$ flops.

5.2. The line search, BFGS update, and termination criterion. The line search for the reduced-Hessian methods is a slightly modified version of that used in the package NPSOL [10]. It is designed to ensure that α_k satisfies the strong Wolfe conditions:

$$f(x_k + \alpha_k p_k) \leq f(x_k) + \mu \alpha_k g_k^T p_k \quad \text{and} \quad |g_{k+1}^T p_k| \leq \eta |g_k^T p_k|$$

with $0 \leq \mu \leq \eta < 1$ and $\mu < \frac{1}{2}$. (For more details concerning algorithms designed to meet these criteria, see, e.g., Gill, Murray, and Wright [11], Fletcher [5], and Moré and Thuente [20].) The step length parameters are $\mu = 10^{-4}$ and $\eta = 0.9$. The line search is based on using a safeguarded polynomial interpolation to find an approximate minimizer of the univariate function $\psi(\alpha) = f(x_k + \alpha p_k) - f(x_k) - \mu \alpha g_k^T p_k$ (see Moré and Sorensen [19]). The step α_k is the first member of a minimizing sequence $\{\alpha_k^i\}$ satisfying the Wolfe conditions. The sequence is always started with $\alpha_k^0 = 1$.

If α_k satisfies the Wolfe conditions, it holds that $y_k^T s_k \geq -(1-\eta)\alpha_k g_k^T p_k > 0$, and hence the BFGS update can be applied without difficulty. On very difficult problems, however, the combination of a poor search direction and a rounding error in f may prevent the line search from satisfying the line search conditions within 20 function

TABLE 6.1
Comparison of RHRL with four reinitialization values on 64 *CUTE problems.*

Option	Itn	Fcn	Cpu
R0	26553	45476	22:26
R1	34815	41327	21:50
R2	25808	39856	20:56
R3	23356	30684	18:01

evaluations. In this case, the search terminates with α_k corresponding to the best value of f found so far. If this α_k defines a strict decrease in f, the minimization continues and the BFGS update is skipped unless $y_k^T s_k \geq \epsilon_M \alpha_k |g_k^T p_k|$, where ϵ_M is the machine precision. If a strict decrease in f is not obtained after 20 function evaluations, then the algorithm is terminated (no restarts are allowed).

Every run was terminated when $\|g_k\| < 10^{-6}$ or $\|g_k\| < \epsilon_M^{0.8}(1 + |f(x_k)|)$. Our intent is to compare methods *when they succeed*, and identify the cases where methods fail.

6. Numerical results. The methods are implemented in double precision Fortran 77 on an SGI O2 with R5000 processor and 64MB of RAM. The test problems are taken from the CUTE collection (see Bongartz et al. [1]).

The test set was constructed using the CUTE interactive `select` tool, which allows identification of groups of problems with certain features:

```
Objective function type         : *
Constraints type                : U
Regularity                      : R
Degree of available derivatives : *
Problem interest                : *
Explicit internal variables     : *
Number of variables             : v
Number of constraints           : 0.
```

Of the 73 problems selected with this specification, *indef* was omitted from the trials because the iterates became unbounded for all the methods. For the remaining problems, the smallest allowable value of n satisfying $n \geq 300$ was chosen, with the following exceptions: Smaller values of n were used for *penalty*3, *mancino*, and *sensors* because they otherwise took too much memory to "decode" using the SIF decoder (compiled with the option "tobig"); a smaller n was used for *penalty*2 because the initial steepest-descent direction for $n = 300$ was unusable by the optimizers; $n = 50$ was used for *chnrosnb* and *errinros* since this was the largest value admitted; and the value $n = 31$ was used for *watson* for the same reason.

Four more problems were identified using the `select` tool with input:

```
Number of variables             : in [    50,   300 ].
```

This resulted in problems *tointgor*, *tointpsp*, *tointqor*, and *hydc*20*ls* being added to the test set. All of these problems have 50 variables except *hydc*20*ls*, which has 99 variables.

We begin our discussion by identifying the "best" implementation of the various

TABLE 6.2
Final nonoptimal gradients for RHRL reinitialization schemes on 5 CUTE problems.

Problem	Reinitialization option			
	R0	R1	R2	R3
bdqrtic	–	1.0E-4	–	1.3E-5
cragglvy	–	9.5E-6	–	–
engval1	2.0E-6	3.0E-6	–	–
fletcbv3	3.8E-1	1.0E-1	–	–
vardim	–	2.1E-5	2.1E-5	2.1E-5

TABLE 6.3
RHRL vs. RHR on 62 CUTE problems.

Method	Itn	Fcn	Cpu
RHRL (R3)	19453	20949	16:35
RHR (R0)	25898	43676	22:19
RHR (R1)	31609	35722	19:30
RHR (R2)	25575	35994	21:00
RHR (R3)	25445	27411	18:30

reduced-Hessian methods presented earlier. There follows a numerical comparison between this method and several leading optimization codes, including NPSOL [10], the CDS method [28], and the ACS method [15].

6.1. The benefits of reinitializing curvature. First, we compare an implementation of RHRL using four alternative values of σ_{k+1} (see (4.1)–(4.3)), labeled R0–R3. Table 6.1 gives the total number function evaluations and total cpu time (in minutes and seconds) required for a subset of 64 of the 76 problems. The subset contains the 64 problems for which RHRL succeeded with every choice of σ_{k+1}.

The results clearly indicate that *some form* of approximate curvature reinitialization is beneficial in terms of the overall number of function evaluations. This point is reinforced when RHRL is compared with NPSOL, which has no provision for altering the initial approximate curvature. However, on the CUTE problems, the decrease in function evaluations does not necessarily translate into a large advantage in terms of cpu time. The reason for this is that on the problems where a large difference in function evaluations occurs, the required cpu time is small. For example, on the problem *extrosnb*, the function evaluations/cpu seconds required using R0–R3 are, respectively, 5398/39.6, 4914/20.6, 6764/27.1, and 3418/14.6. Although R3 (i.e., RHRL implemented with reinitialization option R3) offers a large advantage in terms of function evaluations, it gains little advantage in cpu time relative to the *overall* cpu time required for all 64 problems. This is partly because the CUTE problems tend to have objective functions that are cheap to evaluate. (On problem *extrosnb*, RHRL with R0 takes longer than R2 because the final r_k is roughly twice the R2 value. With R0, r_k reaches 67 at iteration 81 and remains at that value until convergence at iteration 3862. With R2, however, r_k reaches only 17 by iteration 81 and is never greater than 35, converging after 4976 iterations.)

None of R0–R3 succeed on problems *arglinb*, *arglinc*, *freuroth*, *hydc*20*ls*, *mancino*, *nonmsqrt*, and *penalty*3. The problems for which at least *one* of R0–R3 fail are *bdqrtic*, *cragglvy*, *engval1*, *fletcbv*3, and *vardim*. Table 6.2 shows which of R0–R3 failed on these five problems by giving the corresponding values for $\|g_k\|$ at the final iterate. It

TABLE 6.4
Final nonoptimal gradients for RHRL and RHR on 7 CUTE problems.

Problem	RHRL (R3)	RHR (R0)	RHR (R1)	RHR (R2)	RHR (R3)
bdqrtic	1.3E-5	–	8.3E-5	–	–
cragglvy	–	–	1.2E-5	–	–
*engval*1	–	–	1.9E-6	–	–
*fletcbv*3	–	3.3E-01	1.4E-1	1.3E-1	6.7E-2
fletchbv	–	–	6.4E+3	–	1.4E+6
*penalty*2	–	1.1E+11	–	–	–
vardim	2.1E-5	–	2.1E-5	2.1E-5	2.1E-5

TABLE 6.5
RHRL vs. NPSOL on 64 CUTE problems.

Method	Itn	Fcn	Cpu
RHRL (R3)	22362	27458	17:05
NPSOL	29204	49420	23:55

should be noted that R2 has no real advantage over R3 in this table because R3 nearly meets the termination criteria on *bdqrtic* (the final objective value is 1.20×10^{-3} for both methods) and because 74 function evaluations are required by R2, compared to 53 for R3. The cpu seconds required by R2 and R3 on *bdqrtic* are 0.38 and 0.28.

6.2. The benefits of lingering. Now we illustrate the benefits of lingering by comparing RHRL with an algorithm, designated RHR, that reinitializes the curvature when a gradient is accepted, but does not linger. Five algorithms were tested: RHR with all four resetting options R0–R3, and RHRL with option R3. The termination criteria were satisfied on 62 of the 76 problems. Table 6.3 gives the total number of iterations, function evaluations, and cpu time required. All five algorithms failed on problems *arglinb*, *arglinc*, *freuroth*, *hydc*20*ls*, *mancino*, *nonmsqrt* and *penalty*3. This leaves seven other problems on which at least one of the five methods failed. The two-norms of the final nonoptimal gradients for these problems are given in Table 6.4.

6.3. RHRL compared with NPSOL. Here we make a numerical comparison between RHRL and the general-purpose constrained solver NPSOL (see Gill et al. [10]). NPSOL uses a Cholesky factor of the approximate Hessian. The code requires approximately $n^2 + \mathcal{O}(n)$ storage locations for unconstrained optimization. The flop count for the method is $4n^2 + \mathcal{O}(n)$ per iteration, with approximately $3n^2$ operations being required for the BFGS update to the Cholesky factor.

In our comparison, both methods meet the termination criteria on 64 of the 76 problems. Table 6.5 gives the total number of iterations, function evaluations and cpu time for RHRL with R3 and for NPSOL. Both methods failed on problems *arglinb*, *arglinc*, *hydc*20*ls*, *mancino*, *nonmsqrt*, and *penalty*3. This leaves six other problems on which at least one of the methods failed (see Table 6.6).

6.4. RHRL compared with automatic column scaling. Next we compare RHRL and Algorithm ACS proposed by Lalee and Nocedal [15]. ACS requires storage for an $n \times n$ lower-Hessenberg matrix plus $\mathcal{O}(n)$ additional locations; however, the implementation uses $n^2 + \mathcal{O}(n)$ elements, as does NPSOL. The flop count for ACS is not given by Lalee and Nocedal, but we estimate it to be $4n^2 + \mathcal{O}(n)$. This number is obtained as follows. A total of $\frac{3}{2}n^2 + \mathcal{O}(n)$ flops are required to restore the lower-

TABLE 6.6
Final nonoptimal gradients for RHRL and NPSOL on 6 CUTE problems.

Problem	RHRL (R3)	NPSOL
bdqrtic	1.3e-5	–
*engval*1	–	1.8e-6
fletchbv	–	1.1E+6
freuroth	3.6e-6	–
*penalty*2	–	2.6E+4
vardim	2.1e-5	–

TABLE 6.7
RHRL vs. ACS on 57 CUTE problems.

Method	Itn	Fcn	Cpu
RHRL (R3)	24667	34947	20:19
ACS (23)	32828	39725	29:38

Hessenberg matrix to a lower-triangular matrix L_k prior to solving for the search direction. Another n^2 flops are required to compute the search direction p_k. After some additional $\mathcal{O}(n)$ operations, $\frac{3}{2}n^2+\mathcal{O}(n)$ flops are required for the BFGS update, assuming that $L_k^T p_k$ is saved while computing p_k. Note that the work is essentially the same as that needed for NPSOL because both methods require two sweeps of rotations to maintain a triangular factor of the approximate Hessian. We have neglected any computations required for scaling since the version of ACS we tested scales very conservatively. In particular, the ACS code has six built-in rescaling strategies numbered 21–26. The last two only rescale during the first iteration. Option 23 appears to be the one preferred by Lalee and Nocedal since it performs the best on the problems of Moré, Garbow, and Hillstrom [18] (see Lalee and Nocedal [15, p. 20]). This is the option used in the tests below.

In our comparison, both RHRL and ACS meet the termination criteria on 57 of the 76 problems. In Table 6.7, we show the total numbers of iterations, function evaluations and cpu time for RHRL with R3 and for ACS with scaling option 23. Both methods fail on problems *arglinb*, *arglinc*, *bdqrtic*, *freuroth*, *hydc*20*ls*, *mancino*, *nonmsqrt*, and *penalty*3. This leaves nine other problems on which at least one of the methods fails (see Table 6.8).

6.5. RHRL compared with conjugate-direction scaling. In this section, we provide a comparison between RHRL and Algorithm CDS proposed by Siegel [28]. CDS requires $n^2+\mathcal{O}(n)$ storage locations, making it comparable with the implemented versions of both NPSOL and ACS. An iteration of the algorithm presented by Siegel [28, p. 9] requires $7n^2+n(n-l_c)+\mathcal{O}(n)$ flops when a "full" step is taken, where the parameter l_c is analogous to the partition parameter of RHRL. Otherwise the count is $4n^2+3nl_c+\mathcal{O}(n)$ flops (see [28, p. 23]). However, Siegel gives a more complicated formulation that requires only $3n^2+\mathcal{O}(n)$ flops per iteration (see [28, pp. 23–26]). For our comparison, the faster version of CDS was emulated by using the simpler formulation while counting the cpu time for only $3n^2+\mathcal{O}(n)$ flops per iteration. This was done as follows. In order to isolate the $3n^2$ flops, the flop count for the simpler CDS method was divided into five parts. The first part is the calculation of $V_k^T g_k$, which requires n^2 flops for both the full and partial step. The second part is the start of the Goldfarb–Powell BFGS update to V_k and the calculation of the search

TABLE 6.8
Final nonoptimal gradients for RHRL and ACS on 11 *CUTE problems.*

Problem	RHRL(R3)	ACS(23)
chainwoo	–	1.8e-6
cragglvy	–	2.7e-5
edensch	–	3.1e-6
*engval*1	–	2.7e-6
errinros	–	5.2e-6
*ncb*20	–	2.2e-6
*ncb*20*b*	–	4.5e-6
noncvxun	–	3.1e-6
*penalty*2	–	3.3e+1
tointgor	–	2.7e-6
vardim	2.1e-5	–

TABLE 6.9
RHRL (R2) vs. CDS on 68 *CUTE problems.*

Method	Itn	Fcn	Cpu
RHRL (R2)	27190	43577	22:20
CDS	26974	44003	27:10

direction. This part involves postmultiplying V_k by an orthogonal lower-Hessenberg matrix, Ω_k say, and requires $3n^2$ flops for the full step. (Powell [25, p. 42] suggests a way to reduce this cost.) In the case of the partial step, $3nl_c$ flops are required. In both cases, the search direction can be provided as a by-product at the same cost (see Powell [25, pp. 41–42]), but Siegel prefers to list this calculation separately. Hence, the third part of CDS is the calculation of the search direction, which requires n^2 and nl_c additional flops for the full and partial steps, respectively. The fourth part of CDS is the completion of the BFGS update, which requires an additional $2n^2$ flops for both steps (see Powell [25, p. 33]). The last part of CDS scales trailing columns of V_k and requires $n(n - l_c)$ flops (multiplications). Hence, in order to count only $3n^2$ flops per iteration for *both* types of step, we omit the cpu time for the three tasks of calculating $V_k\Omega_k$, computing the search direction, and scaling V_k.

The CDS code was implemented with the same line search used for RHR and RHRL. This allows a fair comparison of CDS with RHRL (R2), which is the reduced-Hessian variant satisfying the conditions of Theorem 4.3.

Table 6.9 illustrates the *connection* between RHRL and CDS (see Theorem 4.3) as well as the *advantage* of using the reduced-Hessian method. A direct comparison can be made because both methods meet the termination criteria on the same 68 problems. The problems on which both methods fail are *arglinb*, *arglinc*, *freuroth*, *hydc*20*ls*, *mancino*, *nonmsqrt*, *penalty*3, and *vardim*. Note that despite the similarity in the number of iterations and function evaluations, RHRL is roughly 21% faster than CDS. The improvement in cpu time is gained primarily because the reduced-Hessian approach allows the search direction to be computed more cheaply during iterations when r is much less than n.

To further illustrate the connection between RHRL (R2) and CDS, Table 6.10 compares data obtained for the two methods at particular iterations. This comparison is only for illustration and no statistical argument is being made. The three problems were chosen because the iterates match quite closely. Table 6.9 illustrates that the

TABLE 6.10
Iteration data for RHRL (R2) and CDS on 3 CUTE problems.

Problem	Method	k	α_k	$f(x_k)$	$\|g_k\|$	$\|g_k^T p_k\|$
broydn7d	RHRL	144	0.12E+00	0.12069659E+03	0.35E-05	0.19E-11
	CDS	144	0.12E+00	0.12069659E+03	0.35E-05	0.19E-11
dixmaanl	RHRL	322	0.10E+01	0.10000001E+01	0.57E-04	0.12E-06
	CDS	322	0.10E+01	0.10000001E+01	0.57E-04	0.12E-06
morebv	RHRL	300	0.10E+01	0.15708889E-07	0.71E-05	0.72E-08
	CDS	300	0.10E+01	0.15708889E-07	0.71E-05	0.72E-08

TABLE 6.11
RHRL (R3) vs. CDS on 67 CUTE problems.

Method	Itn	Fcn	Cpu
RHRL (R3)	26255	37082	21:10
CDS	26921	43900	27:07

iterates are not always identical.

When RHRL is used with R3, a further improvement in cpu time is gained relative to CDS. In this case, RHRL fails on one additional problem, *bdqrtic*, with final gradient norm 1.3×10^{-5}. Table 6.11 compares the iterations, function evaluations, and cpu time for the two methods on the set of 67/76 mutually successful test problems. Here, RHRL has a 28% advantage in cpu time.

7. Conclusions. Algorithms that compute an explicit reduced-Hessian approximation have two important advantages over conventional quasi-Newton methods. First, the amount of computation for each iteration is significantly less during the early stages. Second, approximate curvature along directions that lie off the manifold can be reinitialized as the iterations proceed, thereby reducing the influence of a poor initial estimate of the Hessian.

The results of section 6 indicate that reduced-Hessian methods can require substantially less computer time than a conventional BFGS method and some recently proposed extensions. Part of the reduction in computer time corresponds to the smaller number of iterations and function evaluations required when using the reinitialization strategy (see Tables 6.5, 6.7, 6.9, and 6.11). However, much of the reduction in computer time is the result of the average cost of an iteration being less than for competing methods. This result may seem surprising when it is considered that a reduced-Hessian iteration generally requires more work as the number of iterations approaches n. For example, if an RH direction is always used on a problem with dimension $n = 300$, an iteration of RHRL is more expensive than an iteration of CDS when $r_k \geq 170$. However, on 83% of the problems tested with dimension 300, the average value of r_k remains below this value. In most cases, the *maximum* value of r_k remained small relative to 170. Table 7.1 gives the average and maximum values of r_k for 54 CUTE problems with $n = 300$. The maximum value of r_k exceeds 170 on only 20 of the 54 problems listed, while the average value exceeds 170 on only 9 problems. (Remarkably, there are several cases where r_k does not exceed 50.) It is this feature that gives RHRL a significant advantage over the other algorithms tested in terms of the cost of the linear algebra per iteration.

TABLE 7.1
Final and average values of r_k on 54 CUTE problems with dimension $n = 300$.

Problem	Mean r	Final r	Problem	Mean r	Final r
arglina	2	2	*fletchbv*	288	300
arwhead	2	2	*fletchcr*	26	50
brownal	3	3	*genrose*	225	300
broydn7d	78	154	*hilberta*	9	12
brybnd	27	52	*hilbertb*	4	6
chainwoo	101	193	*liarwhd*	2	2
cosine	6	11	*morebv*	159	296
cragglvy	54	106	*ncb*20	87	173
dixmaana	3	3	*ncb*20b	166	300
dixmaanb	6	11	*noncvxu*2	164	298
dixmaanc	7	13	*noncvxun*	150	290
dixmaand	8	14	*nondia*	2	2
dixmaane	48	93	*nondquar*	217	300
dixmaanf	47	90	*penalty*1	2	2
dixmaang	46	91	*powellsg*	4	4
dixmaanh	45	88	*power*	86	98
dixmaani	170	300	*quartc*	274	298
dixmaank	173	300	*schmvett*	22	43
dixmaanl	169	300	*sinquad*	3	3
dixon3dq	158	300	*sparsine*	216	300
dqdrtic	5	5	*sparsqur*	174	300
dqrtic	274	298	*srosenbr*	2	2
edensch	14	29	*testquad*	104	161
*engval*1	12	23	*tointgss*	2	2
extrosnb	28	32	*tridia*	85	166
*fletcbv*2	149	297	*vareigvl*	104	204
*fletcbv*3	284	300	*woods*	4	4

Acknowledgments. We thank Marucha Lalee and Jorge Nocedal for graciously providing a copy of their ACS code.

REFERENCES

[1] I. BONGARTZ, A. R. CONN, N. I. M. GOULD, AND P. L. TOINT, CUTE*: Constrained and unconstrained testing environment*, ACM Trans. Math. Software, 21 (1995), pp. 123–160.

[2] J. W. DANIEL, W. B. GRAGG, L. KAUFMAN, AND G. W. STEWART, *Reorthogonalization and stable algorithms for updating the Gram–Schmidt QR factorization*, Math. Comput., 30 (1976), pp. 772–795.

[3] 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, London, New York, 1981, pp. 167–199.

[4] M. C. FENELON, *Preconditioned Conjugate-Gradient-Type Methods for Large-Scale Unconstrained Optimization*, Ph.D. thesis, Department of Operations Research, Stanford University, Stanford, CA, 1981.

[5] R. FLETCHER, *Practical Methods of Optimization*, 2nd ed., John Wiley, Chichester, New York, Brisbane, Toronto, Singapore, 1987.

[6] R. FLETCHER AND M. J. D. POWELL, *A rapidly convergent descent method for minimization*, Computer Journal, 6 (1963), pp. 163–168.

[7] R. W. FREUND, G. H. GOLUB, AND N. M. NACHTIGAL, *Iterative solution of linear systems*, Acta Numer. 1992, Cambridge University Press, Cambridge, UK, 1992, pp. 57–100.

[8] P. E. GILL, G. H. GOLUB, W. MURRAY, AND M. A. SAUNDERS, *Methods for modifying matrix factorizations*, Math. Comput., 28 (1974), pp. 505–535.

[9] P. E. GILL AND M. W. LEONARD, *Limited-Memory Reduced-Hessian Methods for Unconstrained Optimization*, Numerical Analysis Report NA 97-1, University of California, San Diego, CA, 1997.

[10] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *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, 1986.

[11] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, London, New York, 1981.

[12] D. Goldfarb, *Factorized variable metric methods for unconstrained optimization*, Math. Comput., 30 (1976), pp. 796–811.

[13] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.

[14] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, Frontiers in Appl. Math. 16, SIAM, Philadelphia, PA, 1995.

[15] M. Lalee and J. Nocedal, *Automatic column scaling strategies for quasi-Newton methods*, SIAM J. Optim., 3 (1993), pp. 637–653.

[16] M. W. Leonard, *Reduced Hessian Quasi-Newton Methods for Optimization*, Ph.D. thesis, Department of Mathematics, University of California, San Diego, CA, 1995.

[17] D. C. Liu and J. Nocedal, *On the limited memory BFGS method for large scale optimization*, Math. Programming, 45 (1989), pp. 503–528.

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

[19] J. J. Moré and D. C. Sorensen, *Newton's method*, in Studies in Numerical Analysis, MAA Stud. Math. 24, G. H. Golub, ed., The Mathematical Association of America, Washington, DC, 1984, pp. 29–82.

[20] J. J. Moré and D. J. Thuente, *Line search algorithms with guaranteed sufficient decrease*, ACM Trans. Math. Software, 20 (1994), pp. 286–307.

[21] J. L. Nazareth, *The method of successive affine reduction for nonlinear minimization*, Math. Programming, 35 (1986), pp. 97–109.

[22] L. Nazareth, *A relationship between the BFGS and conjugate gradient algorithms and its implications for new algorithms*, SIAM J. Numer. Anal., 16 (1979), pp. 794–800.

[23] J. Nocedal, *Updating quasi-Newton matrices with limited storage*, Math. Comput., 35 (1980), pp. 773–782.

[24] S. S. Oren and D. G. Luenberger, *Self-scaling variable metric (SSVM) algorithms, Part* I*: Criteria and sufficient conditions for scaling a class of algorithms*, Management Science, 20 (1974), pp. 845–862.

[25] M. J. D. Powell, *Updating conjugate directions by the BFGS formula*, Math. Programming, 38 (1987), pp. 693–726.

[26] D. F. Shanno and K. Phua, *Matrix conditioning and nonlinear optimization*, Math. Programming, 14 (1978), pp. 149–160.

[27] D. Siegel, *Implementing and Modifying Broyden Class Updates for Large Scale Optimization*, Report DAMTP/1992/NA12, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, UK, 1992.

[28] D. Siegel, *Modifying the BFGS update by a new column scaling technique*, Math. Programming, 66 (1994), pp. 45–78.

SIAM J. OPTIM.
Vol. 12, No. 1, pp. 238–261

RESCALING AND STEPSIZE SELECTION IN PROXIMAL METHODS USING SEPARABLE GENERALIZED DISTANCES*

PAULO JOSÉ DA SILVA E SILVA†, JONATHAN ECKSTEIN‡, AND CARLOS HUMES, JR.†

Abstract. This paper presents a convergence proof technique for a broad class of proximal algorithms in which the perturbation term is separable and may contain barriers enforcing interval constraints. There are two key ingredients in the analysis: a mild regularity condition on the differential behavior of the barrier as one approaches an interval boundary and a lower stepsize limit that takes into account the curvature of the proximal term. We give two applications of our approach. First, we prove subsequential convergence of a very broad class of proximal minimization algorithms for convex optimization, where different stepsizes can be used for each coordinate. Applying these methods to the dual of a convex program, we obtain a wide class of multiplier methods with subsequential convergence of both primal and dual iterates and independent adjustment of the penalty parameter for each constraint. The adjustment rules for the penalty parameters generalize a well-established scheme for the exponential method of multipliers. The results may also be viewed as a generalization of recent work by Ben-Tal and Zibulevsky [*SIAM J. Optim*, 7 (1997), pp. 347–366] and Auslender, Teboulle, and Ben-Tiba [*Comput. Optim. Appl.*, 12 (1999), pp. 31–40; *Math. Oper. Res.*, 24 (1999), pp. 645–668] on methods derived from φ-divergences. The second application established full convergence, under a novel stepsize condition, of Bregman-function-based proximal methods for general monotone operator problems over a box. Prior results in this area required strong restrictive assumptions on the monotone operator.

Key words. proximal algorithms, Bregman distances, φ-divergence, convex programming, variational inequalities

AMS subject classifications. 90C25, 90C33

PII. S1052623499365784

1. Introduction.

Let $B \subseteq \mathbb{R}^n$ denote the possibly unbounded n-dimensional "box" $([a_1, b_1] \times \cdots \times [a_n, b_n]) \cap \mathbb{R}^n$, where $-\infty \leq a_i < b_i \leq +\infty$ for all $i = 1, \ldots, n$. This paper considers two closely related problems: the minimization problem

$$\min_{x \in B} f(x), \tag{1.1}$$

where $f : \mathbb{R}^n \to (-\infty, +\infty]$ is a closed proper convex function, and the variational inequality

$$0 \in T(x) + N_B(x), \tag{1.2}$$

where T is a (possibly set-valued) maximal monotone operator and $N_B(x)$ denotes the cone of vectors normal to the set B at x. It is well known that, under mild regularity conditions, (1.1) is the special case of (1.2) for which $T = \partial f$, the subgradient mapping of f.

*Received by the editors December 7, 1999; accepted for publication (in revised form) February 23, 2001; published electronically October 18, 2001. This paper was partially supported by FAPESP, processo 96/09939-0, the Rutgers University Board of Trustees Research Fellowship for Scholarly Excellence, and PRONEX, convênio 76.97.1008.00.

http://www.siam.org/journals/siopt/12-1/36578.html

†Department of Computer Science, Instituto de Matemática e Estatística—University of São Paulo, São Paulo, Brazil (rsilva@ime.usp.br, humes@ime.usp.br). The first author was a visiting student at RUTCOR from July 1, 1999 through December 31, 1999.

‡Faculty of Management and RUTCOR, 640 Bartholomew Road, Busch Campus, Rutgers University, Piscataway, NJ 08854 (jeckstei@rutcor.rutgers.edu).

The last decade has seen considerable progress in the theory of proximal point methods based on generalized distances [11, 13, 19, 5, 21, 31, 14, 2, 3, 17]. Such methods use a scalar-valued regularization function to derive better-behaved versions of problems (1.1) and (1.2). In this article, we consider separable regularization terms of the form

$$D(x,y) = \sum_{i=1}^{n} d_i(x_i, y_i),$$

where $d_1, \ldots, d_n$ are scalar functions conforming to very general assumptions (see Assumption 2.1 below). In particular, we assume that as $x \in \operatorname{int} B$ approaches the boundary of B, $\|\nabla_1 D(x,y)\| \to \infty$, where ∇_1 denotes the gradient with respect to the first vector argument. The distance-like measure D can be, for example, the squared Euclidean distance, a Bregman distance [8], or a φ-divergence [19] (see section 2.2 below).

Using these regularization terms, proximal methods for (1.1) take the form

$$x^{k+1} = \arg\min_x \left\{ f(x) + \sum_{i=1}^{n} \frac{1}{\alpha_i^k} d_i(x_i, x_i^k) \right\}, \tag{1.3}$$

where α^k is a positive n-dimensional vector whose elements are called *stepsizes*. Note that we allow different stepsizes for each coordinate, as suggested by a variety of computational and theoretical studies [32, 5, 2, 3]. Moreover, since $\|\nabla_1 D(x, x^k)\| \to \infty$ as x approaches the boundary of B, the regularization acts not only as a stabilizing proximal term but also as a kind of barrier function keeping the iterates within $\operatorname{int} B$.

In the case of the variational inequality (1.2), (1.3) generalizes to finding x^{k+1} satisfying the recursion

$$0 \in T(x^{k+1}) + \operatorname{diag}(\alpha^k)^{-1} \nabla_1 D(x^{k+1}, x^k). \tag{1.4}$$

We derive some general results for these types of algorithms in section 2, assuming that the stepsizes conform to a special rule that takes into account the curvature of the proximal term. This rule, although restrictive, appears to cover cases of the greatest practical interest; as we shall see, it covers the stepsize/penalty selection rules proposed in [32, 5, 2, 3].

Section 3 uses the results of section 2 to obtain subsequential convergence results for the generalized proximal minimization algorithm (1.3).

A critical application of (1.3), considered in section 3.2, is when f is minus the dual function of a convex program such as

$$\begin{array}{ll} \min & g_0(y) \\ \text{such that (s.t.)} & g_i(y) \le 0, \quad i = 1, \ldots, n, \end{array} \tag{1.5}$$

where $g_0, \ldots, g_n : \mathbb{R}^m \to \mathbb{R}$ are differentiable convex functions.[1] We also assume that this problem is feasible; i.e., there is a $\bar{y} \in \mathbb{R}^m$ such that $g_i(\bar{y}) \le 0, i = 1, \ldots, n$. Choosing B to be any box containing the nonnegative orthant and f to be the negative of the dual function of (1.5), we may implement (1.3) via a multiplier method in

[1] Actually, the results of section 3.2 continue to hold [28] if one supposes only that $g_0, \ldots, g_n : \mathbb{R}^m \to (-\infty, \infty]$ are closed proper convex and assumes appropriate conditions on the effective domains of the objective and constraints, as in [24, Chapter 28]. However, this further generality makes the proofs more convoluted and is dropped for the sake of simplicity in the exposition.

which a sequence of unconstrained penalized versions of (1.5) must be solved. This construction leads to a class of multiplier methods that is extremely broad, subsuming both the classical quadratic augmented Lagrangian and the exponential method of multipliers [32, 6].

For these multiplier methods, our stepsize choice ensures that for indices i with $x_i^k \to 0$ the corresponding penalty term is augmented so that it does not become so "flat" as to permit infeasibility of primal limit points. Empirically, the technique speeds convergence, and it also appears in a convergence *rate* analysis in [32] for the exponential method of multipliers case. Ben-Tal and Zibulevsky [5] have proved the optimality of the accumulation points of the exponential method, together with a class of proximal terms closely related to φ-divergences, and their results are extended in [3]. Section 3 places such results in a broader context that includes Bregman distances.

In section 4, we restrict our attention to Bregman distances. It has been known for the better part of a decade that, when $D(\cdot,\cdot)$ is any Bregman distance and the stepsizes do not vary by coordinate, the recursion (1.4) converges to a solution of the variational inequality (1.2) in various special cases: when $T = \partial f$, the subdifferential of a closed proper convex function f, or when $\operatorname{dom} T \subseteq \operatorname{int} B$, meaning that all constraints must already be embedded in the operator T. In [9], these results were extended to "paramonotone" operators T, a category which includes $T = \partial f$ as a special case. Unfortunately, many interesting practical cases, such as the subdifferential maps of saddle functions, are not paramonotone. More recently, Auslender, Teboulle, and Ben-Tiba [2] have obtained strong results for general maximal monotone T, but only for a specific φ-divergence choice of $D(\cdot,\cdot)$. As noted in [4], these results can be extended to the (generally non-Bregman) case in which $D(\cdot,\cdot)$ is obtained by adding a quadratic to any member of the class Φ_2 of [3].

Section 4 shows convergence, for general maximal monotone T, of the proximal method (1.4), where $D(\cdot,\cdot)$ is a Bregman distance, to a solution of (1.2). We do impose some additional assumptions, derived from those of section 2. First, we assume that the Bregman function used to construct the distance is twice-differentiable, which is not part of the standard Bregman function setup. Second, in addition to our general stepsize rule, we also require that the stepsizes do not vary by coordinate, that is, $\alpha_1^k = \cdots = \alpha_n^k$ for all k. The resulting condition is stronger than the usual requirement that the stepsize is simply bounded away from zero, but is crucial to the analysis, which blends the techniques of section 2 with traditional Fejér monotonicity arguments. Still, we have managed to substitute conditions on $D(\cdot,\cdot)$ and α^k, which are parts of the algorithm, for conditions on T, which is part of the problem to be solved.

Finally, we allow the calculations required for the recursions (1.3) and (1.4) to be performed approximately, as is likely to be necessary in practice. For the rescaling minimization case of section 3, we adopt a constructive approximation criterion inspired by [17] and [29]. However, our criterion, which is tailored to the proximal minimization case, appears to be new. In the variational inequality analysis of section 4, we use the simple, verifiable criterion of [14], although extension to the more sophisticated criterion of [29] may well be possible.

In summary, the primary contributions of this paper are

- a novel convergence proof framework for a broad class of proximal algorithms;
- using this framework to establish subsequential convergence of a wide range of proximal minimization algorithms (1.3) with differing stepsize parameters for each coordinate—this result in turn leads to subsequential convergence

of a broad class of multiplier methods with differing penalty parameters for each constraint;
- using the framework to show convergence of "interior" Bregman proximal point algorithms for maximal monotone operators, with a novel stepsize condition, but without the usual restrictive assumptions on the operator T.

The new proximal minimization approximation criterion of section 3 constitutes an additional contribution.

2. Fundamental analysis. This section develops the fundamental analysis necessary for our results. We concentrate our attention on the variational problem (1.2), since it subsumes the minimization problem (1.1) under mild assumptions.

In order to simplify the notation, we denote, for $i = 1, \ldots, n$,

$$d_i'(x_i, y_i) \stackrel{\text{def}}{=} \frac{\partial d_i}{\partial x_i}(x_i, y_i),$$

$$d_i''(x_i, y_i) \stackrel{\text{def}}{=} \frac{\partial^2 d_i}{\partial x_i^2}(x_i, y_i).$$

We are now able to present the necessary assumptions on the functions d_i.

ASSUMPTION 2.1. *For $i = 1, \ldots, n$, the function $d_i : \mathbb{R} \times (a_i, b_i) \to (-\infty, \infty]$ has the following properties:*

2.1.1. *For all $y_i \in (a_i, b_i)$, $d_i(\cdot, y_i)$ is closed and strictly convex, with its minimum at y_i. Moreover,* $\operatorname{int} \operatorname{dom} d_i(\cdot, y_i) = (a_i, b_i)$.

2.1.2. *d_i is continuously differentiable over $(a_i, b_i) \times (a_i, b_i)$, and, for all $y_i \in (a_i, b_i)$, $d_i''(y_i, y_i)$ exists and is strictly positive.*

2.1.3. *For all $y_i \in (a_i, b_i)$, $d_i(\cdot, y_i)$ is essentially smooth* [24, Chapter 26].

2.1.4. *There exist $\rho, \epsilon > 0$ such that if either $-\infty < a_i < y_i \leq x_i < a_i + \epsilon$ or $b_i - \epsilon < x_i \leq y_i < b_i < +\infty$, then $\rho\, |d_i'(x_i, y_i)| \leq d_i''(y_i, y_i)\, |x_i - y_i|$.*

The assumption of strict convexity is standard in generalized proximal methods. The assumption of twice-differentiability is also quite common, although many existing results require only a once-differentiable d_i. The essential smoothness assumption makes the distance D act like a barrier function, forcing the iterates defined by the recursion (1.4), and hence its approximate version (2.1) below, to remain in the interior of the box B. In section 2.2, we specialize these assumptions to the case of Bregman distances and φ-divergences, where similar comments can be made.

Finally, the fourth part of the assumption is new to the theory of generalized proximal methods, but is not very restrictive in practice. In particular, we show in section 2.2 that, for Bregman distances and φ-divergences, this condition can be written in terms of the kernels used to obtain the regularizations, and that it holds for most of the examples of which we are aware.

In addition, we make the following standard regularity assumption which, in view of the barrier function properties of d_i, is required for any sensible application of (1.4).

ASSUMPTION 2.2. $\operatorname{dom} T \cap \operatorname{int} B \neq \emptyset$.

We are now able to present the proximal minimization algorithm.

RESCALING PROXIMAL METHOD FOR VARIATIONAL INEQUALITY (RPMVI).

1. **Initialization:** *Let $k = 0$. Choose a scalar $c > 0$ and an initial iterate $x^0 \in \operatorname{int} B$.*
2. **Iteration:**
   (a) *Choose $\alpha^k \in \mathbb{R}^n_{++}$ such that $\alpha_i^k \geq c \max\left\{1, d_i''(x_i^k, x_i^k)\right\}$ for $i = 1, \ldots, n$.*
   (b) *Find x^{k+1} and e^{k+1} such that*

$$e^{k+1} \in T(x^{k+1}) + \operatorname{diag}(\alpha^k)^{-1} \nabla_1 D(x^{k+1}, x^k). \tag{2.1}$$

(c) *Let $k = k + 1$, and repeat the iteration.* ∎

To guarantee the convergence of the RPMVI, we need additional assumptions on the stepsizes $\{\alpha_i^k\}$ and the error sequence $\{e^k\}$; see Assumption 2.3 below.

We define

$$\gamma^k \stackrel{\text{def}}{=} e^k - \operatorname{diag}(\alpha^{k-1})^{-1} \nabla_1 D(x^k, x^{k-1}), \tag{2.2}$$

whence it is clear from (2.1) that $\gamma^k \in T(x^k)$ for all $k \geq 1$.

ASSUMPTION 2.3. *Let $\{\beta_k\}$ be a real sequence converging to zero. The error sequence $\{e^k\}$, the regularization functions $d_1, \ldots, d_n$, and the stepsizes $\{\alpha_i^k\}$, $i = 1, \ldots, n$, must be chosen in order to guarantee the following:*

2.3.1. $\left|e_i^k\right| \leq \frac{1}{\alpha_i^{k-1}} \left|d_i'(x_i^k, x_i^{k-1})\right| + \beta_k$.

2.3.2. *If $\bar{x}$ is an accumulation point of $\{x^k\}$, i.e., there is an infinite set $\mathcal{K} \subseteq \mathbb{N}$ such that $x^k \to_{\mathcal{K}} \bar{x}$, then, for each $i = 1, \ldots, n$, either $\gamma_i^k \to_{\mathcal{K}} 0$ or there is an infinite set $\mathcal{K}' \subseteq \mathcal{K}$ such that $x_i^{k-1} \to_{\mathcal{K}'} \bar{x}_i$.*

Assumption 2.3 may seem artificial at this point, but sections 3 and 4 will describe settings in which it is easily verifiable.

2.1. Convergence analysis. We assume throughout this section that Assumptions 2.1 and 2.2 hold and that sequences $\{\alpha_k\}$, $\{x^k\}$, and $\{e^k\}$ conforming to the recursions of the RPMVI algorithm and Assumption 2.3 exist. In sections 3 and 4 we will present conditions which, in more specific settings, guarantee the existence of such sequences.

LEMMA 2.4. *Let $\bar{x} \in \mathbb{R}^n$ be a limit point of $\{x^k\}$, i.e., $x^k \to_{\mathcal{K}} \bar{x}$ for some infinite set $\mathcal{K} \subseteq \mathbb{N}$. Then for $i = 1, \ldots, n$,*

$$\begin{array}{rcll} \lim\limits_{k \to_{\mathcal{K}} \infty} \gamma_i^k & = & 0 & \text{if } \bar{x}_i \in (a_i, b_i), \\ \liminf\limits_{k \to_{\mathcal{K}} \infty} \gamma_i^k & \geq & 0 & \text{if } \bar{x}_i = a_i, \\ \limsup\limits_{k \to_{\mathcal{K}} \infty} \gamma_i^k & \leq & 0 & \text{if } \bar{x}_i = b_i. \end{array} \tag{2.3}$$

Proof. For each i, we consider the three possible cases.

First, suppose i is such that $\bar{x}_i \in (a_i, b_i)$. For the sake of a contradiction, assume that $\gamma_i^k \not\to_{\mathcal{K}} 0$. Then, using Assumption 2.3.2, there is an infinite set $\mathcal{K}' \subseteq \mathcal{K}$ and a $\zeta > 0$ such that for all $k \in \mathcal{K}'$, $|\gamma_i^k| \geq \zeta$ and $x_i^{k-1} \to_{\mathcal{K}'} \bar{x}_i$. Therefore

$$\begin{array}{rcll} \left|\gamma_i^k\right| & = & \left| e_i^k - \dfrac{1}{\alpha_i^{k-1}} d_i'(x_i^k, x_i^{k-1}) \right| & \\ & \leq & \dfrac{1}{\alpha_i^{k-1}} \left|d_i'(x_i^k, x_i^{k-1})\right| + \left|e_i^k\right| & \\ & \leq & \dfrac{2}{\alpha_i^{k-1}} \left|d_i'(x_i^k, x_i^{k-1})\right| + \beta_k & \text{(Assumption 2.3.1)} \\ & \leq & (2/c) \left|d_i'(x_i^k, x_i^{k-1})\right| + \beta_k & \text{(choice of } \alpha_i^k) \\ & \underset{\mathcal{K}'}{\to} & (2/c) \left|d_i'(\bar{x}_i, \bar{x}_i)\right| + 0 & \\ & = & 0 & \text{(the minimum of } d_i(\cdot, \bar{x}_i) \text{ is } \bar{x}_i). \end{array}$$

This result contradicts $\left|\gamma_i^k\right| > \zeta,\ k \in \mathcal{K}'$.

Next, consider the case $\bar{x}_i = a_i$, and suppose that $\liminf_{k\to_{\mathcal{K}}\infty} \gamma_i^k < 0$. Then, using Assumption 2.3.2, there must be a $\zeta > 0$ and an infinite set $\mathcal{K}' \subseteq \mathcal{K}$ such that for all $k \in \mathcal{K}'$, $\gamma_i^k \le -\zeta$ and $x_i^{k-1} \to_{\mathcal{K}'} \bar{x}_i$. Then

$$\begin{aligned}
\zeta &\le \left|\gamma_i^k\right| \\
&= \left| e_i^k - \frac{1}{\alpha_i^{k-1}} d_i'(x_i^k, x_i^{k-1}) \right| \\
&\le \frac{2}{\alpha_i^{k-1}} \left| d_i'(x_i^k, x_i^{k-1}) \right| + \beta_k \\
&\le \frac{2}{c d_i''(x_i^{k-1}, x_i^{k-1})} \left| d_i'(x_i^k, x_i^{k-1}) \right| + \beta_k.
\end{aligned}$$

Let ϵ be as in Assumption 2.1.4. If there is an infinite set $\mathcal{K}'' \subseteq \mathcal{K}'$ such that $x_i^{k-1} \le x_i^k \le a_i + \epsilon$ for all $k \in \mathcal{K}''$, we can conclude from the assumption that

$$\begin{aligned}
\zeta &\le \frac{2}{c d_i''(x_i^{k-1}, x_i^{k-1})} \left| d_i'(x_i^k, x_i^{k-1}) \right| + \beta_k \\
&\le \frac{2 d_i''(x_i^{k-1}, x_i^{k-1})}{\rho c d_i''(x_i^{k-1}, x_i^{k-1})} \left| x_i^k - x_i^{k-1} \right| + \beta_k \\
&= \frac{2}{\rho c} \left| x_i^k - x_i^{k-1} \right| + \beta_k \\
&\underset{\mathcal{K}''}{\to} 0,
\end{aligned}$$

since $x^{k-1} \to_{\mathcal{K}'} \bar{x}_i$ and $\beta_k \to 0$; but this conclusion contradicts $\zeta > 0$. Therefore, $x_i^k \le x_i^{k-1}$ for sufficiently large $k \in \mathcal{K}'$.

As $d_i(\cdot, x_i^{k-1})$ achieves its minimum at x_i^{k-1}, having $x_i^k \le x_i^{k-1}$ implies that $d_i'(x_i^k, x_i^{k-1}) \le 0$. Hence

$$\begin{aligned}
\gamma_i^k &= e_i^k - \frac{1}{\alpha_i^{k-1}} d_i'(x_i^k, x_i^{k-1}) \\
&\ge \frac{1}{\alpha_i^{k-1}} \left| d_i'(x_i^k, x_i^{k-1}) \right| - \left| e_i^k \right| \\
&\ge -\beta_k \\
&> -\zeta
\end{aligned}$$

for sufficiently large $k \in \mathcal{K}'$, a contradiction with $\gamma_i^k \le -\zeta < 0$, $k \in \mathcal{K}'$.

Finally, the case of $\bar{x}_i = b_i$ is analogous to the case of $\bar{x}_i = a_i$. ∎

LEMMA 2.5. *Let $\bar{x}$ be a limit point of $\{x^k\}$, i.e., $x^k \to_{\mathcal{K}} \bar{x}$ for some infinite set $\mathcal{K} \subseteq \mathbb{N}$. Then, $\{\gamma^k\}_{\mathcal{K}}$ is bounded.*

Proof. By Assumption 2.2, there must exist some $\widetilde{x} \in \operatorname{dom} T \cap \operatorname{int} B$. Let $\widetilde{\gamma} \in T(\widetilde{x})$. The monotonicity of T implies that, for all $k \ge 0$,

$$0 \le \langle x^k - \widetilde{x}, \gamma^k - \widetilde{\gamma} \rangle = \sum_{i=1}^{n} (x_i^k - \widetilde{x}_i)(\gamma_i^k - \widetilde{\gamma}_i). \tag{2.4}$$

We will show that unboundedness of $\{\gamma^k\}_{\mathcal{K}}$ would contradict this inequality for some sufficiently large k.

If $\{\gamma^k\}_{\mathcal{K}}$ is unbounded, there must exist an infinite $\mathcal{K}' \subseteq \mathcal{K}$ such that $\{\gamma^k\}_{\mathcal{K}'}$ converges in $[-\infty, \infty]^n$, with at least one $\{\gamma_i^k\}_{\mathcal{K}'}$ unbounded. Lemma 2.4 implies that for each unbounded coordinate i, either

$$\gamma_i^k \to_{\mathcal{K}'} +\infty \text{ and } \bar{x}_i = a_i$$
$$\text{or}$$
$$\gamma_i^k \to_{\mathcal{K}'} -\infty \text{ and } \bar{x}_i = b_i.$$

Therefore, for each unbounded coordinate of $\{\gamma^k\}_{\mathcal{K}'}$, we have

$$(x_i^k - \widetilde{x}_i)(\gamma_i^k - \widetilde{\gamma}_i) \to_{\mathcal{K}'} (a_i - \widetilde{x}_i)(+\infty) = -\infty$$
$$\text{or}$$
$$(x_i^k - \widetilde{x}_i)(\gamma_i^k - \widetilde{\gamma}_i) \to_{\mathcal{K}'} (b_i - \widetilde{x}_i)(-\infty) = -\infty.$$

On the other hand, for coordinates such that $\{\gamma_i^k\}_{\mathcal{K}'}$ is bounded, $(x_i^k - \widetilde{x}_i)(\gamma_i^k - \widetilde{\gamma}_i)$ is also bounded. Thus, for sufficiently large $k \in \mathcal{K}' \subseteq \mathcal{K}$, $\langle x^k - \widetilde{x}, \gamma^k - \widetilde{\gamma} \rangle$ must be negative, contradicting (2.4). □

Finally, the main convergence theorem for the RPMVI follows.

THEOREM 2.6. *If $\{x^k\}$ is a sequence generated by the RPMVI algorithm with Assumptions* 2.1, 2.2, *and* 2.3 *holding, then all the limit points of $\{x^k\}$ are solutions to the variational inequality problem* (1.2).

Proof. Let $\bar{x}$ be any limit point of $\{x^k\}$, i.e., $x^k \to_{\mathcal{K}} \bar{x}$, for some infinite set $\mathcal{K} \subseteq \mathbb{N}$. From Lemma 2.5, we know that the corresponding sequence $\gamma^k \in T(x^k)$ is bounded. Then, there must exist some $\mathcal{K}' \subseteq \mathcal{K}$ with $\gamma^k \to_{\mathcal{K}'} \bar{\gamma} \in \mathbb{R}^n$. Since T must be outer semicontinuous [27, Exercise 12.8(b)], it follows that $\bar{\gamma} \in T(\bar{x})$. Lemma 2.4 implies that

$$\begin{array}{ll} \bar{\gamma}_i = 0 & \text{if } \bar{x}_i \in (a_i, b_i), \\ \bar{\gamma}_i \geq 0 & \text{if } \bar{x}_i = a_i, \\ \bar{\gamma}_i \leq 0 & \text{if } \bar{x}_i = b_i, \end{array}$$

and these conditions are equivalent to $0 \in T(\bar{x}) + N_B(\bar{x})$. □

Incidentally, it is possible to eliminate the requirement of twice-differentiability of $d_i(\cdot, y_i)$, at the cost of some additional complexity in the description of the method. Specifically, consider replacing Assumption 2.1.4 with the condition that there exist $\delta, \epsilon > 0$ and functions $L_i : (a_i, b_i) \to (\delta, +\infty)$ such that if either $-\infty < a_i < y_i \leq x_i < a_i + \epsilon$ or $b_i - \epsilon < x_i \leq y_i < b_i < +\infty$, then

$$|d_i'(x_i, y_i)| \leq L_i(y_i) \, |x_i - y_i| \, .$$

If the stepsizes are now selected so that for some scalar $c > 0$, we have for all $i = 1, \ldots, n$ and $k \geq 0$ that $\alpha_i^k \geq c L_i(x_i^k)$, then the conclusions of Theorem 2.6 continue to hold. We may examine this variation of the analysis in subsequent research. The present approach is equivalent to taking $L_i(y_i) = (1/\rho) d''(y_i, y_i)$, a natural choice since $d''(y_i, y_i)$ measures the rate of change of $d'(\cdot, y_i)$ around y_i.

2.2. Some examples of d_i functions. We present some examples of d_i functions that conform with Assumption 2.1. In particular, we show that two classes of regularizations widely studied in the literature, Bregman distances [11, 13] and φ-divergences [19], conform to the assumption under very mild restrictions.

2.2.1. Bregman distances. Bregman distances were introduced in [8] and have been studied in the context of proximal methods in [11, 12, 13], as well as many subsequent works. To construct each regularization $d_i(\cdot,\cdot)$, one uses an auxiliary convex function h_i and defines $d_i(x_i, y_i) = h_i(x_i) - h_i(y_i) - h_i'(y_i)(x_i - y_i)$. Nonseparable distances can also be constructed in a similar way, but the separable case is the most common.

The following properties guarantee that Assumption 2.1 holds for such d_i.

ASSUMPTION 2.7. *For $i = 1, \ldots, n$, the function $h_i : \mathbb{R} \to (-\infty, \infty]$ has the following properties:*

2.7.1. *h_i is closed,* $\operatorname{int}\operatorname{dom} h = (a_i, b_i)$*, and h_i is twice continuously differentiable, with a strictly positive second derivative throughout (a_i, b_i).*

2.7.2. *h_i is essentially smooth.*

2.7.3. *There exist $\rho > 0$ and $\epsilon > 0$ such that if either $-\infty < a_i < y_i \le x_i < a_i + \epsilon$ or $b_i - \epsilon < x_i \le y_i < b_i < +\infty$, then $\rho\,|h_i'(x_i) - h_i'(y_i)| \le h_i''(y_i)\,|x_i - y_i|$.*

Note that Assumption 2.7.1 implies that each h_i is strictly convex. Assumption 2.7.3 corresponds to Assumption 2.1.4, since $d_i''(x_i, y_i) = h_i''(x_i)$. Fortunately, it is not very restrictive. Consider the case of finite a_i. Since $\lim_{x_i \searrow a_i} h_i'(x_i) = -\infty$, we know that $h_i''(x_i)$ must be unbounded above as $x_i \searrow a_i$. To violate the assumption, $h_i''(x_i)$ would have to oscillate unboundedly as $x_i \searrow a_i$. As far as we are aware, every separable Bregman function proposed so far conforms not only to Assumption 2.7.3 but to a more stringent, easier-to-verify condition, as follows.

LEMMA 2.8. *If there is an $\epsilon > 0$ such that for all $x_i \in (a_i, a_i + \epsilon) \cap \mathbb{R}$, h_i'' is nonincreasing, and for all $x \in (b_i - \epsilon, b_i) \cap \mathbb{R}$, h_i'' is nondecreasing, then Assumption* 2.7.3 *holds.*

Proof. Suppose that $a_i > -\infty$ and let $x_i, y_i \in (a_i, a_i + \epsilon)$ and $y_i < x_i$. Then

$$|h_i'(x_i) - h_i'(y_i)| \;=\; \int_{y_i}^{x_i} h_i''(z)\,dz \;\le\; h''(y_i)\,|x_i - y_i|\,.$$

Therefore, Assumption 2.7.3 holds with $\rho = 1$. The case $b_i < \infty$ is analogous. ∎

Examples of functions h_i for which all of these assumptions hold are

- $h_i(x) = \frac{1}{2}x^2$, with $a_i = -\infty$, $b_i = +\infty$,
- $h_i(x) = -\log x$, with $a_i = 0$, $b_i = +\infty$,
- $h_i(x) = x \log x$, with $a_i = 0$, $b_i = +\infty$,
- $h_i(x) = x \log(e^x - 1)$, with $a_i = 0$, $b_i = +\infty$,
- $h_i(x) = x^\alpha - x^\beta$, for $\alpha \in [1, 2]$ and $\beta \in (0, 1)$, with $a_i = 0$, $b_i = +\infty$.

Finally, we note that for finite a_i we do *not* yet assume that $h_i(x_i)$ must approach a finite limit as $x_i \searrow a_i$, nor similarly for $x_i \nearrow b_i < +\infty$. Such an assumption is quite common in the theory of Bregman distances [11, 13, 9, 29], but, similarly to [21], it is not needed for the results of section 3 below. We will use it, however, in the variational inequality analysis of section 4.

2.2.2. φ-divergences. The φ-divergence regularizations have been studied in the context of proximal methods, for example, in [19], and more recently in [5, 3]. In these works, the box considered is the positive orthant, i.e., $B = \mathbb{R}^n_+$. An auxiliary strictly convex scalar function φ is used to define the distance d_i, but this time by

$$d_i(x_i, y_i) = y_i \varphi\left(\frac{x_i}{y_i}\right). \tag{2.5}$$

The following hypotheses can be used to guarantee Assumption 2.1 when $B = \mathbb{R}^n_+$.

ASSUMPTION 2.9. *The function $\varphi : \mathbb{R} \to (-\infty, +\infty]$ is such that*

2.9.1. φ *is closed and convex, with* $\operatorname{int}\operatorname{dom}\varphi = (0, +\infty)$*;*
2.9.2. φ *is twice differentiable on* $(0, +\infty)$*, with* $\varphi''(t) > 0$ *for all* $t > 0$*;*
2.9.3. $\varphi(1) = \varphi'(1) = 0$*;*
2.9.4. φ *is essentially smooth;*
2.9.5. *There exists a* $\rho > 0$ *such that* $\rho\varphi'(t) \leq \varphi''(1)(t-1)$ *for all* $t \geq 1$.

Slight variations on these assumptions appear, for example, in [5, 3], together with the following examples:

- $\varphi(t) = t\log t - t + 1$;
- $\varphi(t) = -\log t + t - 1$;
- $\varphi(t) = 2(\sqrt{t} - 1)^2$.

The next lemma states that Assumption 2.9.5 above implies Assumption 2.1.4.

LEMMA 2.10. *Let* $(a_i, b_i) = (0, +\infty)$ *and* d_i *be defined as in* (2.5)*. Then Assumption* 2.1.4 *is equivalent to the existence of a* $\rho > 0$ *such that* $\rho\varphi'(t) \leq \varphi''(1)(t-1)$ *for all* $t \geq 1$.

Proof. First we observe that

$$d_i'(x_i, y_i) = \varphi'\left(\frac{x_i}{y_i}\right),$$
$$d_i''(x_i, y_i) = \frac{1}{y_i}\varphi''\left(\frac{x_i}{y_i}\right),$$

and thus

$$d_i''(y_i, y_i) = \frac{1}{y_i}\varphi''(1).$$

Therefore, Assumption 2.1.4 reduces to

$$\exists \rho, \epsilon > 0: \quad 0 < y_i \leq x_i < \epsilon \quad \Rightarrow \quad \rho\varphi'\left(\frac{x_i}{y_i}\right) \leq \frac{1}{y_i}\varphi''(1)(x_i - y_i). \tag{2.6}$$

Taking $x_i \in (0, \epsilon)$, letting y_i range over $(0, x_i]$, and setting $t = x_i/y_i$, we obtain

$$\exists \rho > 0: \quad \rho\varphi'(t) \leq \varphi''(1)(t-1)\ \forall t \geq 1. \tag{2.7}$$

Conversely, if (2.7) is true, then (2.6) holds for an arbitrary choice of $\epsilon > 0$. ∎

We note that in [5], one assumes that the iterations are of the form

$$0 \in \partial f(x^{k+1}) + \operatorname{diag}(\alpha^k)^{-1}\nabla_1 D(x^{k+1}, x^k),$$

for which each α_i^k is greater than c/x_i^k, c being a positive constant. In [2, 3], this property is guaranteed by redefining the distance measure to be

$$\tilde{d}_i(x_i, y_i) = y_i d_i(x_i, y_i) = y_i^2\varphi\left(\frac{x_i}{y_i}\right), \qquad \tilde{D}(x, y) = \sum_{i=1}^n \tilde{d}_i(x_i, y_i)$$

and assuming stepsizes bounded away from zero. In this case, the iteration is

$$0 \in \partial f(x^{k+1}) + \operatorname{diag}(\widetilde{\alpha}^k)^{-1}\nabla_1 \tilde{D}(x^{k+1}, x^k),$$

with $\liminf_{k\to\infty} \widetilde{\alpha}_i^k > 0$ for all i. Defining $\alpha_i^k = \widetilde{\alpha}_i^k/x_i^k$ and rewriting the iteration with respect to D instead of $\tilde{D}$, we recover the rule from [5].

It turns out that these techniques are a special case of our stepsize choice rule, which gives in the case of a φ-divergence that

$$\alpha_i^k \geq c d_i''(x_i^k, x_i^k) = \frac{c\varphi''(1)}{x_i^k},$$

which is identical if one redefines the constant factor c.

Thus, the reader should note that the class of φ-divergences described by Assumption 2.9 encompasses the regularizations studied in [5, 2, 3]. In particular, it includes the classes Φ_1 and Φ_2 described in [3].

However, the stepsize rule in the RPMVI is more stringent than the one in [5, 2, 3], as it also assumes that the stepsize is bounded away from zero. To overcome this slight restriction, we point out that the assumption $\alpha_i^k > c$ is used here only in the first part of the proof of Lemma 2.4, and it can be replaced by the assumption that $d_i''(y_i, y_i)$ is continuous and strictly positive over (a_i, b_i). This condition holds for φ-divergences, since $d_i''(y_i, y_i) = (1/y_i)\varphi''(1) > 0$ for all $y_i > 0$.

In this sense, the results here can be seen as extensions of those in [5, 2, 3].

3. Proximal minimization methods with rescaling. This section applies the analysis of the RPMVI method to the minimization problem (1.1). We leave Assumption 2.1 as a standing assumption; we also make the following standard regularity assumption, which in view of the barrier function properties of D, is required for any sensible application of (1.3).

ASSUMPTION 3.1. $\operatorname{dom} f \cap \operatorname{int} B \neq \emptyset$.

Note that, since $\operatorname{int} B$ is open, this assumption implies that $\operatorname{ri} \operatorname{dom} f \cap \operatorname{int} B \neq \emptyset$, which implies that $\operatorname{dom} \partial f \cap \operatorname{int} B \neq \emptyset$. Then, using [24, Theorem 23.8], one can show that the minimization problem (1.1) is equivalent to the variational inequality problem (1.2) with $T = \partial f$. Moreover, Assumption 2.2 holds.

Then, we specialize the RPMVI to the following algorithm.

RESCALING PROXIMAL MINIMIZATION METHOD (RPMM).

1. **Initialization:** *Choose $c > 0$ and $\sigma \in [0, 1]$. Choose nonnegative scalar sequences $\{s_k\}$ and $\{z_k\}$ with $\sum_{k=1}^{\infty} s_k < \infty$ and $z_k \to 0$. Let $k = 0$ and $x^0 \in \operatorname{int} B$.*
2. **Iteration:**
   (a) *Choose $\alpha^k \in \mathbb{R}^n_{++}$ such that $\alpha_i^k \geq c \max\left\{1, d_i''(x_i^k, x_i^k)\right\}$ for $i = 1, \ldots, n$.*
   (b) *Find $x^{k+1}, e^{k+1} \in \mathbb{R}^n$ such that*

$$e^{k+1} \in \partial f(x^{k+1}) + \operatorname{diag}(\alpha^k)^{-1} \nabla_1 D(x^{k+1}, x^k), \tag{3.1}$$

$$\left|e_i^{k+1}\right| \leq \frac{\sigma}{\alpha_i^k} \left|d_i'(x_i^{k+1}, x_i^k)\right| + \min\left\{\frac{s_{k+1}}{\|x^{k+1} - x^k\|}, z_{k+1}\right\}, \quad i = 1, \ldots, n, \tag{3.2}$$

   with the standing convention that $\min\left\{s_{k+1}/\|x^{k+1} - x^k\|, z_{k+1}\right\}$ is z_{k+1} whenever $x^{k+1} = x^k$.
   (c) *Let $k = k + 1$, and repeat the iteration.* ∎

Note that if one chooses $s_k, z_k = 0$ for all k, then (3.2) reduces to the "constructive" criterion

$$\left|e_i^{k+1}\right| \leq \frac{\sigma}{\alpha_i^k} \left|d_i'(x_i^{k+1}, x_i^k)\right|,$$

reminiscent of [29].

3.1. Convergence analysis. We start by showing that the iteration step is well defined if f is bounded below on B.

LEMMA 3.2. *If f is bounded below on B, then there is a unique point that solves the iteration step of the RPMM with $e^{k+1} = 0$. Thus, a solution to* (3.1)–(3.2) *exists if f is bounded below on B.*

Proof. Let ℓ be a lower bound of f on B. Given $\zeta \in \mathbb{R}$, the level set

$$\left\{ x \in B \;\middle|\; f(x) + \sum_{i=1}^{n} \frac{1}{\alpha_i^k} d_i(x_i, x_i^k) \le \zeta \right\} \subseteq \left\{ x \in B \;\middle|\; \sum_{i=1}^{n} \frac{1}{\alpha_i^k} d_i(x_i, x_i^k) \le \zeta - \ell \right\}.$$

This last set is a level set of $\sum_{i=1}^{n} (1/\alpha_i^k) d_i(\cdot, x_i^k)$ on B, which must be bounded, since by Assumption 2.1.1 this function attains its minimum at the unique point x^k [24, Corollary 8.7.1]. Therefore, $f(\cdot) + \sum_{i=1}^{n} (1/\alpha_i^k) d_i(\cdot, x_i^k)$ attains a minimum on B. The uniqueness of the minimum follows from the strict convexity of $D(\cdot, x^k)$. ∎

To apply the convergence analysis of the previous section to the sequence $\{x^k\}$ computed by the RPMM, it suffices to show that Assumption 2.3 holds. Verification of Assumption 2.3.1 is straightforward.

LEMMA 3.3. *With the definition*

$$\beta_k \stackrel{\text{def}}{=} \min \left\{ \frac{s_k}{\|x^k - x^{k-1}\|}, z_k \right\}$$

for all $k \ge 1$, Assumption 2.3.1 *holds for the RPMM.*

Proof. From the nonnegativity of $\{s_k\}$ and $\{z_k\}$, it follows that $\{\beta_k\}$ is also nonnegative. Since $z_k \to 0$, one also has $\beta_k \to 0$. Moreover, since $\sigma \in [0, 1]$,

$$\left|e_i^k\right| \le \frac{\sigma}{\alpha_i^{k-1}} \left|d_i'(x_i^k, x_i^{k-1})\right| + \beta_k \le \frac{1}{\alpha_i^{k-1}} \left|d_i'(x_i^k, x_i^{k-1})\right| + \beta_k$$

for all k, so Assumption 2.3.1 holds. ∎

As in (2.2), we define for all $k \ge 0$ and $i = 1, \ldots, n$,

$$\gamma_i^k = e_i^k - \frac{1}{\alpha_i^{k-1}} d_i'(x_i^k, x_i^{k-1}),$$

and let $\gamma^k \in \mathbb{R}^n$ be the vector with elements γ_i^k.

LEMMA 3.4. *$\gamma^k \in \partial f(x^k)$ and $\gamma_i^k (x_i^{k-1} - x_i^k) \ge -s_k$ for all $k \ge 0$ and $i = 1, \ldots, n$.*

Proof. The claim that $\gamma^k \in \partial f(x^k)$ follows from the definition of γ^k. For the second claim, we have, using the convexity of $d_i(\cdot, x_i^{k-1})$,

$$\begin{aligned}
\gamma_i^k (x_i^{k-1} - x_i^k) &= \left(e_i^k - \frac{1}{\alpha_i^{k-1}} d_i'(x_i^k, x_i^{k-1}) \right) (x_i^{k-1} - x_i^k) \\
&\ge -\frac{1}{\alpha_i^{k-1}} \underbrace{d_i'(x_i^k, x_i^{k-1})(x_i^{k-1} - x_i^k)}_{\le 0} - \left|e_i^k\right| \left|x_i^{k-1} - x_i^k\right| \\
&= \left(\frac{1}{\alpha_i^{k-1}} \left|d_i'(x_i^k, x_i^{k-1})\right| - \left|e_i^k\right| \right) \left|x_i^{k-1} - x_i^k\right|.
\end{aligned}$$

Using (3.2), it then follows that

$$\begin{aligned}\gamma_i^k(x_i^{k-1}-x_i^k) &\geq \left(\frac{1-\sigma}{\alpha_i^{k-1}}\left|d_i'(x_i^k,x_i^{k-1})\right| - \min\left\{\frac{s_k}{\|x^k-x^{k-1}\|}, z_k\right\}\right)\left|x_i^{k-1}-x_i^k\right| \\ &\geq -\min\left\{\frac{s_k}{\|x^k-x^{k-1}\|}, z_k\right\}\left|x_i^{k-1}-x_i^k\right| \\ &\geq -\min\left\{\frac{s_k}{\|x^k-x^{k-1}\|}, z_k\right\}\left\|x^{k-1}-x^k\right\| \\ &\geq -s_k. \qquad \square\end{aligned}$$

Before proving the next result, we state a helpful technical lemma.

LEMMA 3.5 (see [22, section 2.2]). *Suppose that $\{a_k\}$, $\{\gamma_k\} \subset \mathbb{R}$ are sequences such that $\{a^k\}$ is bounded below, $\sum_{i=1}^{\infty}\gamma_k$ exists and is finite, and the recursion $a_{k+1} \leq a_k + \gamma_k$ holds for all k. Then, $\{a_k\}$ is convergent.*

It is now possible to establish that Assumption 2.3.2 also holds.

LEMMA 3.6. *If f is bounded below on B, then $\{f(x^k)\}$ is convergent and*

$$\left|\gamma_i^k\right|\left|x_i^{k-1}-x_i^k\right| \to 0 \quad \forall i = 1,\ldots,n.$$

Hence Assumption 2.3.2 *holds for the RPMM.*

Proof. Using Lemma 3.4,

$$\begin{aligned}f(x^{k-1}) &\geq f(x^k) + \langle \gamma^k, x^{k-1}-x^k\rangle \\ &\geq f(x^k) - ns_k.\end{aligned}$$

Then, recalling that $\{s_k\}$ is summable, Lemma 3.5 implies that $\{f(x^k)\}$ is a convergent sequence. For $i = 1,\ldots,n$, we also have

$$\begin{aligned}f(x^{k-1}) &\geq f(x^k) + \langle \gamma^k, x^{k-1}-x^k\rangle \\ &\geq f(x^k) - (n-1)s_k + \gamma_i^k(x_i^{k-1}-x_i^k).\end{aligned}$$

Using Lemma 3.4 once again, it follows that

$$f(x^{k-1}) - f(x^k) + (n-1)s^k \geq \gamma_i^k(x_i^{k-1}-x_i^k) \geq -s_k.$$

Taking limits, we conclude that $\gamma_i^k(x_i^{k-1}-x_i^k) \to 0$. $\square$

Thus, Theorem 2.6 implies the optimality of all accumulation points of the sequence $\{x^k\}$. We strengthen this observation below.

THEOREM 3.7. *Suppose that Assumptions* 2.1 *and* 3.1 *hold and that f is bounded below on B. If $\{x^k\}$ has a limit point, then $\{f(x^k)\}$ converges to the infimum of f on B, and all limit points of $\{x^k\}$ will be minimizers of f on B. A condition that guarantees the existence of limit points of $\{x^k\}$ is the boundedness of the solution set, or any other level set of f.*

Proof. As just noted, Lemma 3.6 implies that Assumption 2.3.2 holds, and so Assumption 2.3 holds in its entirety. Assumption 2.1 holds by hypothesis, and, setting $T = \partial f$, Assumption 3.1 implies Assumption 2.2. Thus, the conclusions of Theorem 2.6 apply. Let $\bar{x}$ be a limit point of $\{x^k\}$, i.e., $x^k \to_{\mathcal{K}} \bar{x}$, for some infinite set $\mathcal{K} \subseteq \mathbb{N}$. Theorem 2.6 asserts that $0 \in \partial f(\bar{x}) + N_B(\bar{x})$; by Assumption 3.1, $\bar{x}$ is a

minimizer of f on B. Moreover, since Lemma 2.5 states that $\{\gamma^k\}_{\mathcal{K}}$ is bounded, and since $\{f(x^k)\}$ is convergent by Lemma 3.6,

$$\min_{x\in B} f(x) \;=\; f(\bar{x}) \;\geq\; f(x^k) + \sum_{i=1}^{n} \gamma_i^k(\bar{x}_i - x_i^k) \underset{\mathcal{K}}{\to} \lim_{k\to\infty} f(x^k) \;\geq\; f(\bar{x}).$$

Therefore, $\lim_{k\to\infty} f(x^k) = f(\bar{x})$.

Finally, the boundedness of any level set of a proper closed convex function implies boundedness of all level sets [24, Corollary 8.7.1], and Lemma 3.6 states that $\{f(x^k)\}$ is convergent; consequently it is bounded. Thus, $\{x^k\}$ is also bounded and has limit points. □

3.2. Multiplier methods. We now discuss applying the RPMM to the dual of the convex program (1.5) to obtain multiplier methods. The use of proximal methods to derive multiplier methods for constrained convex optimization is a now-classical subject and may be traced to the seminal paper [26]. In the context of generalized proximal methods, applications can be found, for example, in [30, 13, 19, 21, 31, 3, 17]. In this section, we consider only the case in which the proximal step is done exactly, i.e., we will let $e^k = 0$ for all k, as in [30, 13, 19, 17]. Unfortunately, our approximate-step acceptance rule for the RPMM does not translate directly to an easily verifiable acceptance criterion for an approximate solution of the penalized problem (3.5) below. However, partial results in this direction may be obtained under stringent assumptions on the original problem (1.5); see Appendix B. A criterion in the spirit of (3.2) that does not depend on such assumptions is the subject of ongoing research [15]. We further observe that the approximation criteria of [17, 29] also do not translate readily to a multiplier method setting. On the other hand, under the assumption that the primal objective function g_0 is strongly convex, [26, 21, 3] present some inexact multiplier methods based on a rather different acceptance rule involving optimizing the augmented Lagrangian function to within some tolerance ϵ of its minimum value.

Consider the convex problem (1.5), and let δ_C denote the indicator function of a convex set C. Then we define f to be minus the dual function associated with (1.5), plus $\delta_{\mathbb{R}^n_+}$:

$$f(x) \stackrel{\text{def}}{=} -\inf_{y\in\mathbb{R}^m} \left\{ g_0(y) + \sum_{i=1}^{n} x_i g_i(y) \right\} + \delta_{\mathbb{R}^n_+}(x). \tag{3.3}$$

The dual problem to (1.5) is then equivalent to the minimization of f. Furthermore, we assume the following.

Assumption 3.8.

3.8.1. *The primal problem* (1.5) *has a finite optimal value, and it conforms to the Slater condition.*

3.8.2. *For all* $i = 1, \ldots, n$, *the generalized distances* d_i *conform to Assumption* 2.1 *for* $a_i \leq 0,\ b_i = +\infty$.[2]

3.8.3. *There is an* $\bar{x} > 0$ *such that* $\bar{x} \in \operatorname{dom} f$, *where* f *is as defined in* (3.3).

This assumption has the following consequences: Assumption 3.8.1 implies that the dual solution set is nonempty and bounded [16] and that there is no duality gap. Assumption 3.8.3 implies that Assumption 3.1 holds for f as defined by (3.3).

[2]The case $a_i = -\infty$ is of interest because it includes the classical method of multipliers for problems with inequality constraints [26], along with various extensions described in [13, 20].

Under Assumption 3.8, if we fix $e^k = 0$ for all k, then each iterate x^{k+1} of the RPMM applied to the negative dual functional f may be calculated by the following multiplier method whenever the unconstrained problems (3.5) have solutions:

$$\alpha_i^k \geq c \max\left\{1, d_i''(x_i^k, x_i^k)\right\}, \quad i = 1, \ldots, n, \tag{3.4}$$

$$y^{k+1} \in \operatorname*{Arg\,min}_{y \in \mathbb{R}^m} \left\{ g_0(y) + \sum_{i=1}^n \frac{1}{\alpha_i^k} d_i^{\oplus}\left(\alpha_i^k g_i(y), x_i^k\right) \right\}, \tag{3.5}$$

$$x_i^{k+1} = \nabla_1 d_i^{\oplus}\left(\alpha_i^k g_i(y^{k+1}), x_i^k\right), \quad i = 1, \ldots, n. \tag{3.6}$$

Here, "$\oplus$" denotes the monotone conjugate [24, p. 111] with respect to the first argument, that is, $d_i^{\oplus}(u_i, w_i) = \sup_{x_i \geq 0}\{u_i x_i - d_i(x_i, w_i)\}$.[3] Theorem 3.10 below gives conditions guaranteeing that a y^{k+1} satisfying (3.5) exists.

We relegate to Appendix A the technical aspects of the proof of the equivalence of (3.4)–(3.6) to the RPMM applied to the f defined in (3.3), since they are very similar to earlier proofs for various special cases of (3.5)–(3.6), for example in [30, 13, 19, 21, 17]. In particular, Corollary A.4 establishes the equivalence of the two calculations.

Given this equivalence, Theorem 3.7 asserts the subsequential convergence of the sequence $\{x^k\}$ to a dual solution of (1.5). For the primal sequence, however, it has historically been harder to prove good behavior. For example, in the case of Bregman distances, a guarantee of feasibility of primal accumulation points has relied on stringent assumptions like $\mathbb{R}^n_+ \subset \operatorname{int} B$, as in [13], or strict complementarity [18].

In the case of the RPMM, with its strong stepsize restrictions, the feasibility, and therefore optimality, of accumulation points of $\{y^k\}$ is easily demonstrated.

THEOREM 3.9. *Suppose that Assumption* 3.8 *holds. Pick a scalar* $c > 0$*, let* $x^0 \in \mathbb{R}^n_{++}$*, and suppose that it is possible to obtain a sequence* $\{(\alpha^k, x^k, y^k)\}$ *that obeys the recursions* (3.4)–(3.6)*. Then,* $\{x^k\}$ *is bounded and all its accumulation points are solutions of the dual of* (1.5)*. Moreover,*

$$\limsup_{k\to\infty} g_i(y^k) \leq 0, \qquad i = 1, \ldots, n, \tag{3.7}$$

$$\lim_{k\to\infty} \sum_{i=1}^n x_i^k g_i(y^k) = 0, \tag{3.8}$$

and $\{g_0(y^k)\}$ *converges to the optimal value of the primal problem* (1.5)*. Therefore, any accumulation point of* $\{y^k\}$ *solves the primal problem.*

Proof. As shown in Corollary A.4, the sequence $\{x^k\}$ is the same as would be computed by using the RPMM to solve the dual problem, that is, to minimize f. In particular, $\{x^k\}$ and all its limit points must be nonnegative. Moreover, the Slater condition implies that the dual function has bounded level sets. Then, the boundedness of $\{x^k\}$ and the optimality of its limit points follow from Theorem 3.7.

Let us analyze the primal sequence. For each $i = 1, \ldots, n$, (3.6) implies that

$$g_i(y^k) = \frac{1}{\alpha_i^{k-1}} d_i'(x_i^k, x_i^{k-1}) + \zeta_i^k,$$

where $\zeta_i^k \in \partial\delta_{\mathbb{R}_+}(x_i^k)$. Hence, $\zeta_i^k - g_i(y^k)$ plays the same role as γ_i^k in (2.2), with $e_i^k = 0$.

[3] The classical conjugate ψ^* of a function ψ is defined [24, Chapter 12] via $\psi^*(y) = \sup_{x\in\mathbb{R}^n}\{\langle x, y\rangle - \psi(x)\}$ for any $\psi : \mathbb{R}^n \to (\infty, +\infty]$. The monotone conjugate of ψ is then the classical conjugate of $\psi + \delta_{\mathbb{R}^n_+}$, that is, $\psi^{\oplus}(y) = \sup_{x\geq 0}\{\langle x, y\rangle - \psi(x)\}$.

Let $\{x^k\}_{\mathcal{K}}$ be any convergent subsequence of $\{x^k\}$, and $\bar{x}$ the respective accumulation point, $x^k \to_{\mathcal{K}} \bar{x}$. Lemma 2.4 implies that

$$\begin{array}{rcccll} 0 & = & \lim\limits_{k\to_{\mathcal{K}}\infty} g_i(y^k) - \zeta_i^k & = & \lim\limits_{k\to_{\mathcal{K}}\infty} g_i(y^k) & \text{if } \bar{x}_i > 0, \\ 0 & \geq & \limsup\limits_{k\to_{\mathcal{K}}\infty} g_i(y^k) - \zeta_i^k & \geq & \limsup\limits_{k\to_{\mathcal{K}}\infty} g_i(y^k) & \text{if } \bar{x}_i = 0. \end{array} \tag{3.9}$$

As $\{x^k\}$ is bounded, the above relations imply that

$$0 \geq \limsup_{k\to\infty} g_i(y^k), \qquad i = 1, \ldots, n. \tag{3.10}$$

Now, suppose for the purposes of contradiction that (3.8) does not hold. Then, for some $i = 1, \ldots, n$, there must be an infinite set $\mathcal{K} \subset \mathbb{N}$ and an $\epsilon > 0$ such that

$$\forall k \in \mathcal{K}, \quad \left|x_i^k g_i(y^k)\right| \geq \epsilon. \tag{3.11}$$

Since $\{x^k\}$ is bounded, there exists a refined subsequence $\mathcal{K}' \subseteq \mathcal{K}$ such that $\{x^k\}_{\mathcal{K}'}$ is convergent, with limit $\bar{x} \geq 0$. If $\bar{x}_i > 0$, then (3.11) contradicts (3.9). If $\bar{x}_i = 0$, then (3.11) and (3.9) imply that $g_i(y^k) \to_{\mathcal{K}'} -\infty$. Since Lemma 2.5 asserts that $\{\zeta_i^k - g_i(y^k)\}$ is bounded, we can conclude that $\zeta_i^k \to_{\mathcal{K}'} -\infty$. However, this divergence would imply that x_i^k should be 0 for infinitely many $k \in \mathcal{K}' \subseteq \mathcal{K}$, once again a contradiction of (3.11). Therefore,

$$\lim_{k\to\infty} x_i^k g_i(y^k) = 0, \qquad i = 1, \ldots, n, \tag{3.12}$$

and (3.8) holds.

Finally, we prove that $\{g_0(y^k)\}$ converges to the optimal value. We may use (3.5), (3.6), and the chain rule to see that y^k minimizes the Lagrangian corresponding to the primal problem with the fixed multiplier x^k. Hence,

$$g_0(y^k) + \sum_{i=1}^{n} x_i^k g_i(y^k) = -f(x^k). \tag{3.13}$$

Let $-f^*$ denote the dual optimal value, which is equal to the primal optimal value since there is no duality gap. Theorem 3.7 states that $f(x^k) \to f^*$. Taking limits in (3.13) and using (3.12), it follows that

$$\lim_{k\to\infty} g_0(y^k) = -f^*. \tag{3.14}$$

The feasibility and optimality of the accumulation points of $\{y^k\}$ are then consequences of the continuity of g_i, $i = 0, \ldots, n$, (3.10), and (3.14). ∎

Finally, it is natural to seek conditions under which the penalized subproblems (3.5) must have solutions and the primal sequence $\{y^k\}$ is bounded. The following result addresses these questions under the standard assumption of a bounded solution set.

THEOREM 3.10. *Suppose that the primal solution set is bounded. Given any $\alpha^k > 0$ and (x^k, y^k), there exist (x^{k+1}, y^{k+1}) satisfying the recursions* (3.5)–(3.6). *Moreover, the primal sequence $\{y^k\}$ is bounded.*

Proof. For the first assertion, it suffices to show that the penalized problems (3.5) have solutions. Given any closed proper convex function ψ, we define its *recession*

function ψ_∞ via $\psi_\infty(d) = \lim_{\lambda\to\infty}(\psi(z+\lambda d) - \psi(z))/\lambda$, where $z \in \operatorname{dom}\psi$ may be chosen arbitrarily [24, Theorem 8.5]. The boundedness of the primal solution set is equivalent [7, section 5.3] to

$$(g_i)_\infty(d) \le 0 \quad \forall\, i = 1,\dots,n \quad \Rightarrow \quad (g_0)_\infty(d) > 0. \tag{3.15}$$

Thus, the existence of a solution to (3.5) is a corollary of Lemma A.5 in the appendix, along with the sum rule for recession functions [24, Theorem 9.3].

We now prove that $\{y^k\}$ is bounded. Theorem 3.9 shows that the sequences $\{g_i(y^k)\}$, $i = 1,\dots,n$, are bounded above. From (3.15), unboundedness of $\{y^k\}$ would imply that $g_0(y^k) \to_{\mathcal{K}} \infty$ for some infinite $\mathcal{K} \subseteq \mathbb{N}$. But such unboundedness would contradict $g_0(y^k)$'s convergence to the optimal value. ∎

We remark that the penalty parameter adjustment rule (3.4), as discussed in section 2.2.2, essentially subsumes, in a context broader than φ-divergences, the corresponding rules described in [32] for the exponential method of multipliers and in [5, 3, 4] for a general φ-divergence setting.

We end this section by giving some examples of $d_i^\oplus$ functions that may be derived from separable Bregman distances (see section 2.2.1). Further examples may be obtained from [21, 18, 28]. For a Bregman-derived distance, we have $d_i(x_i, w_i) = h_i(x_i) - h_i(w_i) - h'(w_i)(x_i - w_i)$, whence

$$\begin{aligned} d_i^\oplus(u_i, w_i) &= \sup_{x_i \ge 0} \{u_i x_i - (h_i(x_i) - h_i(w_i) - h'(w_i)(x_i - w_i))\} \\ &= \sup_{x_i \ge 0} \{(u_i + h'(w_i))\, x_i - h_i(x_i)\} + h_i(w_i) - w_i h'(w_i) \\ &= h^\oplus(h'(w_i) + u_i) + h_i(w_i) - w_i h'(w_i), \end{aligned}$$

where $h^\oplus$ denotes the standard monotone conjugate of h. Note that when such a $d_i^\oplus(u_i, w_i)$ is used in the minimization operation in (3.5), the additive terms $h_i(w_i) - w_i h'(w_i)$ are constant and may be discarded. The following examples may now be easily verified:

- If $h_i(x_i) = \frac{1}{2}x_i^2$, then $d_i^\oplus(u_i, w_i) = \frac{1}{2}(\max\{u_i + w_i, 0\}^2 - w_i^2)$, where the $-w_i^2$ term may be disregarded; this choice gives the classical quadratic method of multipliers for inequality constraints.
- If $h_i(x_i) = x_i \log x_i - x_i$, then $d_i^\oplus(u_i, w_i) = w_i e^{u_i} - w_i$, where the $-w_i$ term may be disregarded, yielding the exponentional method of multipliers.
- If $h_i(x_i) = -\log x_i$, then $d_i^\oplus(u_i, w_i) = -\log(1 - w_i u_i)$.

4. Bregman interior point proximal methods for variational inequalities. We now turn our attention to the box-constrained variational inequality problem (1.2), where $T : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is a (possibly set-valued) maximal monotone operator. In this section, we confine ourselves to Bregman distances, as defined in section 2.2.

We augment Assumption 2.2 as follows.

ASSUMPTION 4.1. *T is maximal monotone, the solution set of* (1.2) *is nonempty, and there exists some* $\widetilde{x} \in \operatorname{dom} T \cap \operatorname{int} B$.

Our goal is to show convergence of an approximate version of the iteration (1.4), without further conditions on T. We modify and extend Assumption 2.7 as follows.

ASSUMPTION 4.2. *For $i = 1,\dots,n$, the functions $h_i : \mathbb{R} \to (-\infty, \infty]$ have the same properties specified in Assumption* 2.7*, and furthermore, h_i is continuous on $[a_i, b_i] \cap \mathbb{R}$. Moreover, defining $h(x) = \sum_{i=1}^n h_i(x_i)$ and $D_h(x, y) = \sum_{i=1}^n h_i(x_i) - h_i(y_i) + h_i'(y_i)(x_i - y_i)$,*

4.2.1. *for all* $x \in B$ *and* $\alpha \in \mathbb{R}$, *the level set* $\{y \in \operatorname{int} B \mid D_h(x,y) \le \alpha\}$ *is bounded;*
4.2.2. *if* $\{x^k\} \subset \operatorname{int} B$ *converges to* $x \in \mathbb{R}^n$, *then* $\lim_{k\to\infty} D_h(x,x^k) = 0$*;*
4.2.3. $\operatorname{rge} h' = \mathbb{R}$.

Note that at finite a_i's and b_i's, the corresponding h_i is now required to take a finite value. The algorithm can now be stated.

Box Interior Proximal Point Algorithm (BIPPA).

1. **Initialization:** *Let* $k = 0$, *and fix some scalar* $c > 0$. *Let* $x^0 \in \operatorname{int} B$.
2. **Iteration:** *Choose* α_k *such that* $\alpha_k \ge c\max\{1, h_1''(x_1^k), \dots, h_n''(x_n^k)\}$. *Find vectors* $x^{k+1}, e^{k+1} \in \mathbb{R}^n$ *such that*

$$e^{k+1} \in T(x^{k+1}) + \frac{1}{\alpha_k}\nabla_1 D_h(x^{k+1}, x^k) = T(x^{k+1}) + \frac{1}{\alpha_k}\big(\nabla h(x^{k+1}) - \nabla h(x^k)\big). \tag{4.1}$$

Let $k = k+1$ *and repeat the iteration.* □

4.1. Convergence analysis. First, we cite a result showing that the iteration step of BIPPA is well defined.

Lemma 4.3 (See [13, Theorem 4(i)]). *Under Assumption* 4.2, *there is a unique point* x^{k+1} *that solves the iteration step* (4.1) *of the BIPPA with* $e^{k+1} = 0$.

We note that it is shown in the unpublished dissertation [28] that (4.1) has a unique exact solution even if Assumption 4.2.3 does not hold. This result permits one to dispense completely with Assumption 4.2.3. However, the proof, while essentially a minor modificiation of that of [1, Theorem A.1], is quite involved, so we do not include it here.

To guarantee the convergence of the BIPPA, we must assume some vanishing behavior for $\{e^k\}$; we will use the assumptions of [14]. Although not as general as the criterion used in RPMM, these conditions are better suited to our analysis, since they will permit us to use properties associated with Fejér monotonicity, and are still feasible to enforce computationally.

Assumption 4.4 (See [14]). *The error sequence* $\{e^k\}$ *conforms to*

$$\sum_{k=0}^{\infty} \alpha_k \left\|e^{k+1}\right\| < +\infty;$$

$$\sum_{k=0}^{\infty} \alpha_k \langle e^{k+1}, x^{k+1}\rangle \text{ exists and is finite.}$$

Note that this assumption implies that $\|e^k\| \to 0$, and therefore Assumption 2.3.1 holds with $\beta_k = \|e^k\|_\infty$. We now state some necessary lemmas.

Lemma 4.5 (See [14, Lemma 2]). *Let* $z \in (T + N_B)^{-1}(0)$. *Then, for all* $k \ge 0$,

$$D_h(z, x^{k+1}) \le D_h(z, x^k) - D_h(x^{k+1}, x^k) + \alpha_k \langle e^{k+1}, x^{k+1} - z\rangle. \tag{4.2}$$

Lemma 4.6. *If Assumption* 4.4 *holds, then the sequence* $\{x^k\}$ *is bounded and* $D_h(x^{k+1}, x^k) \to 0$.

Proof. The result will follow from [14, Lemma 3] once we show that, for $z \in (T + N_B)^{-1}(0)$,

$$E(z) \stackrel{\text{def}}{=} \sum_{i=0}^{\infty} \alpha_k \langle e^{k+1}, x^{k+1} - z\rangle$$

exists and is finite. But

$$\sum_{i=0}^{\infty} \left|\alpha_k \langle e^{k+1}, z\rangle\right| \leq \sum_{i=0}^{\infty} \alpha_k \|e^{k+1}\| \|z\|,$$

and Assumption 4.4 implies that the right-hand side of this relation is finite. Hence, $\sum_{i=0}^{\infty} \alpha_k \langle e^{k+1}, z\rangle$ exists and is finite. Using Assumption 4.4 once more, we conclude that $E(z)$ exists and is finite. ∎

We also use a key result from Solodov and Svaiter [29].

THEOREM 4.7 (See [29, Theorem 2.4]). *Let h_i satisfy Assumption* 4.2. *Given two sequences $\{x^k\} \subset B$ and $\{y^k\} \subset \operatorname{int} B$, either one of which is convergent, with $\lim_{k\to\infty} D_h(x^k, y^k) = 0$, then the other sequence also converges to the same limit.*

This theorem implies that $h(x) = \sum_{i=1}^{n} h_i(x_i)$ is a *Bregman function* in the classical sense [8, 10]. Using Theorem 4.7 and Lemma 4.6, we derive the following.

COROLLARY 4.8. *Under Assumptions* 4.1, 4.2, *and* 4.4, *$\{x^k\}$ has at least one limit point. Moreover, if for some infinite set $\mathcal{K} \subseteq \mathbb{N}$ we have $x^k \to_{\mathcal{K}} \bar{x}$, then $x^{k-1} \to_{\mathcal{K}} \bar{x}$. Therefore, Assumption* 2.3.2 *holds.*

Before presenting the main convergence theorem for the BIPPA, we present a final technical lemma that will help us to prove the uniqueness of the accumulation points of $\{x^k\}$.

LEMMA 4.9. *Under Assumption* 4.4, *for all $z \in (T + N_B)^{-1}(0)$, $D_h(z, x^k)$ converges to a value in $[0, +\infty)$ which we will denote by $d(z)$.*

Proof. Consider any $z \in (T + N_B)^{-1}(0)$. Then Lemma 4.5 implies that (4.2) holds. Using Assumption 4.4 and $D_h(x^{k+1}, x^k) \geq 0$, the hypotheses of Lemma 3.5 are satisfied with $a_k = D_h(z, x^k)$ and $\gamma_k = \alpha_k \langle e^{k+1}, x^{k+1} - z\rangle$. Therefore, $\{D_h(z, x^k)\}$ converges, necessarily to a nonnegative value. ∎

Now, the main convergence theorem follows.

THEOREM 4.10. *Under Assumptions* 4.1, 4.2, *and* 4.4, *$\{x^k\}$ converges to a solution of $0 \in T(x) + N_B(x)$.*

Proof. Let $\bar{x}$ be an accumulation point of $\{x^k\}$, i.e., $x^k \to_{\mathcal{K}} \bar{x}$, for some infinite set $\mathcal{K} \subseteq \mathbb{N}$. Such a point exists by Lemma 4.6. From Theorem 2.6, $0 \in T(\bar{x}) + N_B(\bar{x})$.

We now prove the uniqueness of the limit point: from Assumption 4.2.2, we know that $D_h(\bar{x}, x^k) \to_{\mathcal{K}} 0$. Then, $d(\bar{x})$, as defined in Lemma 4.9, is zero. Suppose that $\{x^k\}$ has another accumulation point $x^k \to_{\mathcal{K}'} x'$ for some infinite set $\mathcal{K}' \subseteq \mathbb{N}$. We then have that $D_h(\bar{x}, x^k) \to_{\mathcal{K}'} d(\bar{x}) = 0$, and it follows from Theorem 4.7 that $x' = \bar{x}$. ∎

Another possible application of our fundamental analysis is to try to generalize to solutions of (1.2) the idea of adding the square of the Euclidean norm and an arbitrary generalized distance to obtain Fejér monotonicity, as in [2, 3] for the special case of φ-divergences. The difficulty here is to generalize the condition that defines the class Φ_2 in [3]. This topic is the subject of ongoing research.

Appendix A. Relationship between multiplier and proximal methods. This appendix proves that the RPMM may be applied to minus the dual functional associated with (1.5) via the multiplier method (3.5)–(3.6).

The proof is very similar to the derivation of a special case presented in [17, section 4.2]. Therefore, we will follow the steps in [17], changing notation whenever necessary to suit the present setting.

In particular, as in (1.5), $g_0 : \mathbb{R}^m \to \mathbb{R}$ is the primal objective and $g : \mathbb{R}^m \to \mathbb{R}^n$ is the constraint function, with components g_i, $i = 1, \ldots, n$. We assume that the

g_i, $i = 0, \ldots, n$, are differentiable convex functions and that (1.5) is feasible. Let f be the negative dual function defined in (3.3), which we assume to be somewhere finite. Note that, since f is the pointwise supremum of a nonempty collection of affine functions, it cannot take the value $-\infty$, and is therefore proper. Let $v(\cdot)$ denote the right-hand-side perturbation function associated with the optimization problem (1.5):

$$\forall u \in \mathbb{R}^n,\ v(u) \stackrel{\text{def}}{=} \inf\{g_0(y) \mid y \in \mathbb{R}^m,\ g(y) \leq u\}.$$

It is well known that for all $x \in \mathbb{R}^n$, $f(x) = v^*(-x)$; see [25, Example 1 and Theorem 7].

We also assume the following throughout this section.

ASSUMPTION A.1. *$D : \mathbb{R}^n \to (-\infty, +\infty]$ is a closed, proper, and strictly convex function such that* $\operatorname{ri}(\operatorname{dom} D \cap \mathbb{R}^n_{++}) \cap \operatorname{ri} \operatorname{dom} f \neq \emptyset$.

$D^\oplus$ denotes the monotone conjugate of D, that is, the convex conjugate of $D + \delta_{\mathbb{R}^n_+}$, and $D^\oplus(g(\cdot))$ denotes the usual composition of $D^\oplus$ and g.

We start by proving a slight modification of [6, equation (4.41)] which plays a fundamental role in our analysis.

LEMMA A.2. *If Assumption* A.1 *holds, then*

$$\inf_{y \in \mathbb{R}^m} \{g_0(y) + D^\oplus(g(y))\} = \inf_{u \in \mathbb{R}^n} \{v(u) + D^\oplus(u)\} = \sup_{x \geq 0} \{-f(x) - D(x)\}.$$

Proof. The definition of $D^\oplus$ implies that if $a \geq b$, then $D^\oplus(a) \geq D^\oplus(b)$, i.e., it is nondecreasing.[4] Therefore,

$$\begin{aligned}
\inf_{y \in \mathbb{R}^m} \{g_0(y) + D^\oplus(g(y))\} &= \inf_{y \in \mathbb{R}^m} \{g_0(y) + D^\oplus(g(y))\} \\
&= \inf_{u \in \mathbb{R}^n} \inf_{\substack{y \in \mathbb{R}^m \\ g(y) \leq u}} \{g_0(y) + D^\oplus(g(y))\} \\
&\leq \inf_{u \in \mathbb{R}^n} \inf_{\substack{y \in \mathbb{R}^m \\ g(y) \leq u}} \{g_0(y) + D^\oplus(u)\} \\
&= \inf_{u \in \mathbb{R}^n} \{v(u) + D^\oplus(u)\}.
\end{aligned}$$

On the other hand,

$$\begin{aligned}
\inf_{y \in \mathbb{R}^m} \{g_0(y) + D^\oplus(g(y))\} &\geq \inf_{y \in \mathbb{R}^m} \{v(g(y)) + D^\oplus(g(y))\} \\
&\geq \inf_{u \in \mathbb{R}^n} \{v(u) + D^\oplus(u)\}.
\end{aligned}$$

Hence, the first equality is proved.

Finally, we use Fenchel's duality theorem [24, Theorem 31.1] and the fact that for all $x \in \mathbb{R}^n$, $-f(x) = -v^*(-x)$, to assert that

$$\inf_{u \in \mathbb{R}^n} \{v(u) + D^\oplus(u)\} = \sup_{x \in \mathbb{R}^n} \{-f(x) - D(x) - \delta_{\mathbb{R}^n_+}(x)\} = \sup_{x \geq 0} \{-f(x) - D(x)\}. \qquad \square$$

THEOREM A.3. *Suppose that Assumption* A.1 *holds. Suppose that the (strictly convex) function $f + D$ has the minimizer $\bar{x}$ over $\mathbb{R}^n$, and that there is $\bar{y}$ such that*

$$\bar{y} \in \operatorname*{Arg\,min}_{y \in \mathbb{R}^m} \{g_0(y) + D^\oplus(g(y))\}. \tag{A.1}$$

[4] This inequality is a simple consequence of the definition of the convex conjugate; see [17, Proposition 3].

Then $\bar{x} = \nabla D^{\oplus}(g(\bar{y}))$.

Proof. From the definition of $\bar{y}$ and the nondecreasing property of $D^{\oplus}$, we have that $g_0(\bar{y}) = v(g(\bar{y}))$. Then, defining $\bar{u} = g(\bar{y})$, Lemma A.2 states that

$$v(\bar{u}) + D^{\oplus}(\bar{u}) = \inf_{u \in \mathbb{R}^n} \big\{ v(u) + D^{\oplus}(u) \big\} = \sup_{x \in \mathbb{R}^n} \big\{ -f(x) - D(x) - \delta_{\mathbb{R}^n_+}(x) \big\}.$$

Hence, we may use [23, Theorem 2] to verify that

$$\bar{x} = \arg\max_{x \in \mathbb{R}^n} \big\{ -f(x) - D(x) - \delta_{\mathbb{R}^n_+}(x) \big\} \in \partial D^{\oplus}(\bar{u}) = \big\{ \nabla D^{\oplus}\big(g(\bar{y})\big) \big\},$$

where the last equality is a consequence of $D^{\oplus}$ being the convex conjugate of a strictly convex function, meaning that $\partial D^{\oplus}$ is single-valued throughout its domain [24, Chapter 26]. ∎

COROLLARY A.4. *Let* d_i, $i = 1, \ldots, n$, *conform to Assumption* 2.1, *with* $B \supseteq \mathbb{R}^n_+$. *Suppose that there is an* $\bar{x} > 0$ *such that* $\bar{x} \in \operatorname{dom} f$. *Given* $x^k \in \operatorname{dom} f$, *there is a unique point* x^{k+1} *that satisfies* (1.3). *Moreover, if there exists a point* y^{k+1} *satisfying* (3.5), *then* (3.6) *holds.*

Proof. Since we assumed that primal problem (1.5) is feasible, the weak duality theorem asserts that the dual objective function is bounded above. Hence, f is bounded below and the existence and uniqueness of x^{k+1} is given by Lemma 3.2.

Finally, let $D(\cdot) = \sum_{i=1}^n \frac{1}{\alpha_i^k} d_i(\cdot, x_i^k)$. Then, for all $u \in \mathbb{R}^n$,

$$D^{\oplus}(u) = \sum_{i=1}^n \frac{1}{\alpha_i^k} d_i^{\oplus}(\alpha_i^k u, x_i^k),$$

as the convex conjugate of a separable function is just the sum of the convex conjugates of its components. Also, if we define $h_\alpha(x) = \alpha h(x)$ for some positive number and convex function h, we have

$$h_\alpha^*(x) = \alpha h^*\Big(\frac{x}{\alpha}\Big).$$

The result then follows from the previous theorem. ∎

Now, we analyze the existence of solutions to the penalized problem (A.1). In order to do so, we will use the notation

$$P(\cdot) \stackrel{\text{def}}{=} D^{\oplus}\big(g(\cdot)\big).$$

Note that P is closed because $D^{\oplus} \stackrel{\text{def}}{=} (D + \delta_{\mathbb{R}^n_+})^*$ must be closed [24, Theorem 12.2].

LEMMA A.5. *Suppose that Assumption* A.1 *holds,* $\operatorname{dom} D \supseteq \mathbb{R}^n_{++}$, *and* D *is bounded below. Let* $\mathcal{R} = \{ d \mid (g_i)_\infty(d) \le 0,\ i = 1, \ldots, n \}$. *Then*

$$P_\infty(d) = \begin{cases} 0 & \text{if } d \in \mathcal{R} \\ +\infty & \text{otherwise.} \end{cases}$$

Proof. Let $\bar{y}$ be a feasible point for (1.5). From the definition of P and $g(\bar{y}) \le 0$,

$$P(\bar{y}) \ = \ \sup_{z \ge 0} \big\{ \langle z, g(\bar{y}) \rangle - D(z) \big\} \ \le \ \sup_{z \ge 0} \big\{ -D(z) \big\}.$$

Hence, as D is bounded below, $\bar{y} \in \operatorname{dom} P$. Therefore, since P is a closed convex function,

$$P_\infty(d) = \lim_{t \to \infty} \frac{P(\bar{y} + td) - P(\bar{y})}{t} \tag{A.2}$$

for all $d \in \mathbb{R}^n$.

As $\mathbb{R}^n_{++} \subseteq \operatorname{dom}(D+\delta_{\mathbb{R}^n_+})$ and is an open set, we have $\mathbb{R}^n_{++} \subseteq \operatorname{dom}\partial(D+\delta_{\mathbb{R}^n_+})$, and from $D^{\oplus} = (D+\delta_{\mathbb{R}^n_+})^*$ we then obtain $\mathbb{R}^n_{++} \subseteq \operatorname{rge}\partial D^{\oplus}$. Thus, for all $x > 0$, there exists some $\gamma \in \mathbb{R}^n$ with $x \in \partial D^{\oplus}(\gamma)$. So, for all $x > 0$, there exists some $\gamma \in \mathbb{R}^n$ such that

$$\forall\, t > 0,\, \forall\, d \in \mathbb{R}^m :\; D^{\oplus}(\gamma) + \langle x, g(\bar{y}+td) - \gamma\rangle - P(\bar{y}) \leq P(\bar{y}+td) - P(\bar{y}).$$

Dividing both sides by t and taking limits as $t \to \infty$, (A.2) implies that for all $x \in \mathbb{R}^n_{++}$,

$$\sum_{i=1}^{n} x_i (g_i)_\infty(d) \leq P_\infty(d). \tag{A.3}$$

This summation is well defined since the recession function of a closed proper convex function is also proper [27, Corollary 3.27]. Taking the limit as $x \to 0$, we may conclude that

$$\forall d \in \mathbb{R}^n,\; 0 \leq P_\infty(d). \tag{A.4}$$

Now, we consider two cases:

1. $d \in \mathcal{R}$. Then

   $$g(\bar{y}+td) \leq g(\bar{y}) \quad \forall\, t \geq 0 \quad \Rightarrow \quad P(\bar{y}+td) - P(\bar{y}) \leq 0 \quad \forall\, t \geq 0.$$

   Dividing both sides by t and taking limits as $t \to \infty$, it follows that $P_\infty(d) \leq 0$. Hence, using (A.4), $P_\infty(d) = 0$.
2. $d \notin \mathcal{R}$. Without loss of generality, let us assume that $(g_1)_\infty(d) > \zeta > 0$. Let $x = (M, 1, 1, \ldots, 1) \in \mathbb{R}^n$. From (A.3), it follows that

   $$\forall M > 0,\; M\zeta + \sum_{i=2}^{m} (g_i)_\infty(d) \leq P_\infty(d).$$

   Since $(g_i)_\infty(d) > -\infty$, $i = 1, \ldots, m$, we can take the limit as $M \to \infty$ and conclude that $P_\infty(d) = +\infty$. ∎

Appendix B. Inexact multiplier methods. In this appendix, we present conditions that make it possible to use the RPMM acceptance criterion (3.2) to develop a verifiable test for accepting an approximate solution to the penalized problem (3.5). We retain the assumptions of section 3.2, in particular the differentiability assumptions and Assumption 3.8. Moreover, we assume that $a_i = 0$, $i = 1, \ldots, n$. Then $d_i^{\oplus} = d_i^*$ and, since d_i is essentially smooth, $\mathbb{R}_{++} = \operatorname{int}\operatorname{dom} d_i = \operatorname{dom}\nabla_1 d_i = \operatorname{rge}\nabla_1 d_i^*$ [24, Theorem 23.5].

Let $\sigma \in [0,1]$, $\{s_k\}$ be a nonnegative summable sequence, and $\{z_k\}$ be a nonnegative vanishing sequence. Let y^{k+1} be an approximate solution of the unconstrained minimization (3.5) and let x^{k+1} be defined by (3.6). Note that $x^{k+1} > 0$. To obtain a subgradient of f at x^{k+1}, as required by (3.1), let

$$\widetilde{y} \in \operatorname*{Arg\,min}_{y \in \mathbb{R}^m} \left\{ g_0(y) + \sum_{i=1}^{n} x_i^{k+1} g_i(y) \right\}.$$

Then, for any $x \in \mathbb{R}^n_+$, we have from (3.3) that

$$\begin{aligned} f(x) &\geq -g_0(\widetilde{y}) - \langle x, g(\widetilde{y})\rangle \\ &= -g_0(\widetilde{y}) - \langle x^{k+1}, g(\widetilde{y})\rangle + \langle x^{k+1} - x, g(\widetilde{y})\rangle \\ &= f(x^{k+1}) + \langle x - x^{k+1}, -g(\widetilde{y})\rangle, \end{aligned}$$

whence $-g(\widetilde{y}) \in \partial f(x^{k+1})$. On the other hand, (3.6) and [24, Theorem 23.5] tell us that $g(y^{k+1}) \in \operatorname{diag}(\alpha^k)^{-1}\nabla_1 D(y^{k+1}, y^k)$. Letting $e^{k+1} = g(y^{k+1}) - g(\widetilde{y})$, we then conclude that the acceptance criterion (3.2) will hold if, for $i = 1, \ldots, n$,

$$\left|g_i(\widetilde{y}) - g_i(y^{k+1})\right| \leq \sigma \left|g_i(y^{k+1})\right| + \min\left\{\frac{s_{k+1}}{\|x^{k+1} - x^k\|}, z_{k+1}\right\}. \tag{B.1}$$

Although $\widetilde{y}$ is unknown, the above inequality may be still be verified if we suppose that g_0 is strongly convex with modulus $\zeta > 0$, and the constraints $g_i,\ i = 1, \ldots, n$, are globally Lipschitz continuous with respective constants $L_i,\ i = 1, \ldots, n$.[5] Let

$$\phi_k(y) \stackrel{\text{def}}{=} g_0(y) + \sum_{i=1}^{n} \frac{1}{\alpha_i^k} d_i^*\left(\alpha_i^k g_i(y), x_i^k\right)$$

denote the augmented Lagrangian at step $k \geq 0$. Note that ϕ_k inherits the strong convexity of g_0. Then, since $\nabla\phi_k(\widetilde{y}) = 0$,

$$\zeta \left\|\widetilde{y} - y^{k+1}\right\| \leq \left\|\nabla\phi_k(y^{k+1})\right\|.$$

Using the Lipschitz continuity of the constraints, it follows that

$$\frac{\zeta}{L_i}\left|g_i(\widetilde{y}) - g_i(y^{k+1})\right| \leq \left\|\nabla\phi_k(y^{k+1})\right\|, \quad i = 1, \ldots, n.$$

Therefore, (B.1) holds whenever

$$\left\|\nabla\phi_k(y^{k+1})\right\| \leq \frac{\zeta}{L_i}\left[\sigma\left|g_i(y^{k+1})\right| + \min\left\{\frac{s_{k+1}}{\|x^{k+1} - x^k\|}, z_{k+1}\right\}\right],\ i = 1, \ldots, n. \tag{B.2}$$

This last relation may be readily tested in practice. Furthermore, our final lemma shows that if we choose $s_{k+1}, z_{k+1} > 0$ and use a convergent algorithm to solve the subproblem (3.5), then (B.2) must eventually be satisfied.

LEMMA B.1. *Suppose s_{k+1} and z_{k+1} are both positive, and let $\bar{y}$ be any solution of* (3.5). *There is a neighborhood $\mathcal{N}$ of $\bar{y}$ such that if $y^{k+1} \in \mathcal{N}$, then* (B.2) *holds.*

Proof. Define, for $i = 1, \ldots, n$,

$$\begin{aligned} x_i(y) &\stackrel{\text{def}}{=} \nabla_1 d_i^*\left(\alpha_i^k g_i(y), x_i^k\right), \\ w_i(y) &\stackrel{\text{def}}{=} \|\nabla\phi_k(y)\| - \frac{\zeta}{L_i}\left[\sigma\left|g_i(y)\right| + \min\left\{\frac{s_{k+1}}{\|x(y) - x^k\|}, z_{k+1}\right\}\right], \end{aligned}$$

where $x(y)$ denotes the n-vector of the $x_i(y)$, and the min is taken to be z_{k+1}, as in our standing convention, whenever the enclosed denominator is zero. With this

[5] The strong convexity assumption is usual in the literature; see [3, Remark 5.2] and [21, section 10]. However, these results do not require Lipschitz continuity of the constraints.

convention, the w_i, $i = 1, \ldots, n$, are continuous functions. Moreover, at $\bar{y}$ we have

$$\begin{aligned} w_i(\bar{y}) &\leq 0 - \frac{\zeta}{L_i}\left[\sigma\,|g_i(\bar{y})| + \min\left\{\frac{s_{k+1}}{\|x(\bar{y}) - x^k\|}, z_{k+1}\right\}\right] \\ &\leq -\frac{\zeta}{L_i}\min\left\{\frac{s_{k+1}}{\|x(\bar{y}) - x^k\|}, z_{k+1}\right\} \\ &< 0, \end{aligned}$$

the last inequality following from the positivity of s_{k+1} and z_{k+1}. For each i, the continuity of w_i implies the existence of a neighborhood $\mathcal{N}_i$ of $\bar{y}$ over which w_i is negative. Let $\mathcal{N} = \bigcap_{i=1}^n \mathcal{N}_i$, which is also a neighborhood of $\bar{y}$. Recalling (3.6), we find that (B.2) holds if $y^{k+1} \in \mathcal{N}$. □

REFERENCES

[1] A. Auslender and M. Haddou, *An interior-proximal method for convex linearly constrained problems and its extension to variational problems*, Math. Programming, 71 (1995), pp. 77–100.

[2] A. Auslender, M. Teboulle, and S. Ben-Tiba, *A logarithmic-quadratic proximal method for variational inequalities*, Comput. Optim. Appl., 12 (1999), pp. 31–40.

[3] A. Auslender, M. Teboulle, and S. Ben-Tiba, *Interior proximal and multiplier methods based on second order homogeneous kernels*, Math. Oper. Res., 24 (1999), pp. 645–668.

[4] A. Auslender and M. Teboulle, *Lagrangian duality and related multiplier methods for variational inequality problems*, SIAM J. Optim., 10 (2000), pp. 1097–1115.

[5] A. Ben-Tal and M. Zibulevsky, *Penalty/barrier multiplier methods for convex programming problems*, SIAM J. Optim., 7 (1997), pp. 347–366.

[6] D. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, MA, 1995.

[7] D. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Athena Scientific, Belmont, MA, 1996.

[8] L. M. Bregman, *The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming*, Comput. Math. Math. Phys., 7 (1967), pp. 200–217.

[9] Y. Censor, A. N. Iusem, and S. A. Zenios, *An interior-point method with Bregman functions for the variational inequality problem with paramonotone operators*, Math. Programming, 81 (1998), pp. 373–400.

[10] Y. Censor and A. Lent, *An iterative row-action method for interval convex programming*, J. Optim. Theory Appl., 34 (1981), pp. 321–353.

[11] Y. Censor and S. A. Zenios, *The proximal minimization algorithms with D-functions*, J. Optim. Theory Appl., 73 (1992), pp. 451–464.

[12] G. Chen and M. Teboulle, *Convergence analysis of a proximal-like minimization algorithm using Bregman functions*, SIAM J. Optim., 3 (1993), pp. 538–543.

[13] J. Eckstein, *Nonlinear proximal point algorithms using Bregman functions, with applications to convex programming*, Math. Oper. Res., 18 (1993), pp. 202–226.

[14] J. Eckstein, *Approximate iterations in Bregman-function-based proximal algorithms*, Math. Programming, 83 (1998), pp. 113–124.

[15] J. Eckstein, *A Practical General Approximation Criterion for Methods of Multipliers Based on Bregman Distances*, RUTCOR Research Report RRR 61-2000, Rutgers University, Piscataway, NJ, 2000.

[16] J. Gauvin, *A necessary and sufficient condition to have bounded multipliers in nonconvex programming*, Math. Programming, 12 (1977), pp. 136–138.

[17] C. Humes and P. Silva, *Strict convex regularizations, proximal point and augmented Lagrangians*, RAIRO Oper. Res., 34 (2000), pp. 283–303.

[18] A. N. Iusem, *Augmented Lagrangian methods and proximal points methods for convex optimization*, Investigación Operativa, 8 (1999), pp. 11–49.

[19] A. N. Iusem, B. F. Svaiter, and M. Teboulle, *Entropy-like proximal methods in convex programming*, Math. Oper. Res., 19 (1994), pp. 790–814.

[20] K. C. Kiwiel, *On the twice differentiable cubic augmented Lagrangian*, J. Optim. Theory Appl., 88 (1996), pp. 233–236.

[21] K. C. Kiwiel, *Proximal minimization methods with generalized Bregman functions*, SIAM J. Control Optim., 35 (1997), pp. 1142–1168.
[22] B. Polyak, *Introduction to Optimization*, Optimization Software, New York, 1987.
[23] R. T. Rockafellar, *Extension of Fenchel's duality theorem for convex functions*, Duke Math. J., 33 (1966), pp. 81–89.
[24] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.
[25] R. T. Rockafellar, *Conjugate Duality and Optimization*, SIAM, Philadelphia, 1974.
[26] R. T. Rockafellar, *Augmented Lagrangians and applications of the proximal point algorithm in convex programming*, SIAM J. Control Optim., (1976), pp. 97–116.
[27] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*, Springer-Verlag, Berlin, 1998.
[28] P. Silva, *Tópicos em Métodos de Ponto Proximal*, Ph.D. thesis, Instituto de Matemática e Estatística—University of São Paulo, São Paulo, Brazil, 2000.
[29] M. V. Solodov and B. F. Svaiter, *An inexact hybrid generalized proximal point algorithm and some new results on the theory of Bregman functions*, Math. Oper. Res., 25 (2000), pp. 214–230.
[30] M. Teboulle, *Entropic proximal mappings with applications to nonlinear programming*, Math. Oper. Res., 17 (1992), pp. 670–690.
[31] M. Teboulle, *Convergence of proximal-like algorithms*, SIAM J. Optim., 7 (1997), pp. 1069–1083.
[32] P. Tseng and D. P. Bertsekas, *On the convergence of the exponential multiplier method for convex programming*, Math. Programming, 60 (1993), pp. 1–19.

SIAM J. OPTIM.
Vol. 12, No. 1, pp. 262–282

ROBUST REDUCTION OF A CLASS OF LARGE-SCALE LINEAR PROGRAMS*

ILYA IOSLOVICH†

Abstract. A set of new tests for linear programming presolving analysis is described. These tests are applicable for linear programs with box constraints and positive or zero coefficients. Partial applicability to general linear programming (LP) problems is discussed in a special section. The aim is to detect and remove redundant rows and columns. The tests are based on the solution of some auxiliary LP problems with one constraint and upper bounds on the variables. A comparison with the Klein–Holm numerical test is presented. The tests are applied iteratively to the primal and dual LP problems. The method is also applicable to LP problems with coefficients belonging to some range of uncertainty, providing a robust procedure for scale reduction. A detailed numerical example and results of numerical experiments are presented.

Key words. large-scale linear programming, redundancy, presolving, robust reduction

AMS subject classifications. 90C05, 90C06, 90C11, 90B70, 90C99

PII. S1052623497325454

1. Introduction.

1.1. Presolving analysis. In the last 30 years there has been a great interest in methods for solving large-scale linear programming (LP) problems (see [9, 27, 7, 12, 20, 10]). This is of course motivated by a wide range of applications. Complex methods based on ideas of aggregation and decomposition, as well as modified simplex methods, barrier methods, and interior-point methods [12, 7, 10, 21, 25, 26, 2, 28], have been implemented in various program tools and packages. Simultaneously, computing capabilities have expanded enormously. In spite of this, there is still a big difference between solving an LP problem of intermediate dimension (less than one thousand variables and constraints) and of large dimension (more than ten thousand variables and constraints). Therefore work on producing new versions and more effective solvers is continuing.

Simultaneously, algorithms for presolving analysis have been developed. Usually much more effort is needed for gathering the data for a large-scale problem than for formulating the model and solving the problem on a computer. In the case that a large part of the data is related to redundant constraints, implying that the corresponding rows of inequalities will never be violated, it is much more effective to devote some effort to presolving analysis, which aims to reduce considerably the size of the problem. The same holds for redundant constraints in the dual problem, i.e., columns can be removed that relate to variables that will definitely be on a zero or maximal bound.

In fact, large LP problems almost always contain a significant number of redundant constraints and variables. So the idea is not to solve the LP problem as it was

*Received by the editors July 30, 1997; accepted for publication (in revised form) March 29, 2001; published electronically October 23, 2001. This research was supported by the Wenner–Gren Foundation, NUTEK—the Swedish National Board for Industrial and Technical Development, and also in part by a joint grant from the Center for Absorption in Science of the Ministry of Immigrant Absorption, State of Israel, and the Committee for Planning and Budgeting of the Council for Higher Education under the framework of the Kamea Program.

http://www.siam.org/journals/siopt/12-1/32545.html

†Faculty of Agricultural Engineering, Technion–Israel Institute of Technology, Haifa 32000, Israel (agrilya@tx.technion.ac.il).

formulated, but first to use a special preprocessor that will make a preliminary analysis using a set of tests that evaluate rows and columns of the problem. An effort can then be made to obtain more exact input data for the reduced medium size problem which is then easily and exactly solved. This policy could be briefly expressed in the sentence by Conan Doyle: "Throw off the whole of the impossible and then what remains is the truth." Various presolvers are described in [17] and in [5, 1, 2, 3, 11].

1.2. A prototype: An industrial planning problem. In 1971 a set of very successful algorithms were developed in the former Soviet Union to solve a large number of similar large-scale LP problems of industrial planning. The problem was formulated as follows:

$$\begin{array}{lrcl@{\qquad}rcl} \text{(LPP)} & \varphi & = & f'x \to \max, & f & \geq & 0, \\ & Ax & \leq & l, & A & \geq & 0, \\ & 0 & \leq & x \leq x_u, & l & \geq & 0, \\ & & & & x_u & \geq & 0. \end{array}$$

Here $A \in \mathbf{R}^{m\times n}$, $l \in \mathbf{R}^m$, and $x, x_u, f \in \mathbf{R}^n$. We shall denote the rows of the matrix A as a_i' and the columns as s_j. The dual problem has the form

$$\begin{array}{lrcl@{\qquad}rcl} \text{(DLPP)} & \phi & = & l'y + x_u'u \to \min, & u & \geq & 0, \\ & f & \leq & A'y + u, & y & \geq & 0. \end{array}$$

Here $y \in \mathbf{R}^m, u \in \mathbf{R}^n$, y is the vector of dual variables related to the row constraints, and u is the vector of dual variables related to the upper bounds of primal variables. The primal variables, x, correspond to the vector of planned industrial production, subject to upper limits x_u. If some of these limits are not known, some large numbers could be assigned instead. The "row constraints" $Ax \leq l$ are related to equipment, supplied raw materials, personnel of different professions, and so on. A huge list of production and equipment makes the dimensionality very large. Moreover, the plant consists of several subdivisions, each of them generating their own constraints of the same type. The objective is to maximize the planned profit of the plant. A particular feature is that *all coefficients of the problem are nonnegative*. Only a few of the constraints are known to be limiting—in particular the manpower with fixed salary. A program based on the above approach was designed for this problem, and it operated for a long period (about 15 years). The method made it possible to reduce the initial large-scale problem (15000×5000) to the size of about 100×200. The number of calculations in the single test for one row constraint is of the same order as the problem of finding the median of a set of real numbers (see [8]), namely, $O(q)$ if the row contain only q nonzeros. Therefore the set of tests for all rows of the matrix has approximately the same computational cost as a few steps of the sparse simplex algorithm. Another advantage of the proposed method is that it is convenient for implementation on a parallel computer because it deals with each row or column separately. The innovative method was implemented on a slow Russian mainframe computer, Minsk-22 (whose speed is about that of a PC-286), with only tape volumes; that work is described in [13]. After a long hiatus, this research was continued recently [14, 15, 16].

The algorithms in [13, 14] have only few features in common with the presolvers quoted in [17, 5, 1, 11], and as far as we know, also with commercial presolvers (e.g., [26]).

1.3. Organization of the paper. Here we present an extended version of our method. The subsections related to results previously obtained in [13] are marked with a *. Also the robust variant is especially considered. This means that we consider the problem of finding rows and columns that are redundant for any values of input data in some given range. This range is supposed to be determined by the user who could, for example, point out the possible uncertainty of the data as a percentage (the common situation).

The paper is organized as follows. In section 2 the general statements of the method are made, and in sections 3–4 primal and dual tests are presented. A detailed numerical example illustrates the method. Section 3.10 is devoted to the general LP problem, to which almost all primal tests (but not dual tests) could be applied. Results of numerical experiments with some random sets of problems are presented in section 5. In sections 6–8, robust variants of the tests are derived.

2. General statements. In the previous work [13] we designed a method for reducing large-scale problems of the form (LPP). The main idea was not to solve the very large problem as it is formally defined by the model, but to find its real kernel, extracting those constraints that will always be satisfied because of other constraints, and those variables that can be set in advance (as a result of column redundancy) to their bounds. For this purpose a number of auxiliary small tests are performed, each of them being an LP problem with one row constraint and box-constrained variables. These tests make it possible to find and remove some of the redundant rows. At the same time upper limits for the dual variables are obtained (this theorem was proven in [13], repeated below). As a result of the first step of the tests, the number of rows is reduced, and possibly the structure of the problem can be changed by partitioning (one could check whether the problem is separable). In the second stage, a similar procedure is applied to the dual problem. This leads to a reduction in the number of variables (columns). Then the first stage is repeated, and the testing procedure becomes iterative. Often the problem dimensions reduce by about a hundred-fold, and the sparse matrix becomes dense. In the end, any standard LP method can solve the problem without difficulty, because its size becomes acceptable. Computer time is significantly reduced. Our tests are more complicated than "simple presolving" [1] but still inexpensive in terms of computational time.

First one calculates

$$l_u = Ax_u. \tag{2.1}$$

If, for some component i of the vectors l and l_u, the inequality

$$l_i \ \geq \ l_{iu} \tag{2.2}$$

holds, then obviously row i of matrix A is redundant and may be removed. This test is described in [17]. We assume that this operation already has been done and inequality (2.2) is not satisfied.

For the case when the objective vector f has some zero components, the values l_{iu} in (2.2) should be replaced by the values

$$C = \{j : f_j \ > \ 0\}, \qquad \tilde{l_{iu}} = \sum_{j \in C} a_{ij} x_{ju}.$$

One could easily notice that for any feasible solution there exists a solution with the same value of the objective and $x_j = 0 \ \ \forall j : \ \ f_j = 0$. This equivalent solution will

also be feasible, because the values of the left-hand side in all rows of the constraints $Ax \le l$ of (LPP) will not increase. For example, if $a_i' f = 0$, then the constraint i in (LPP) is redundant.

Now we shall consider the following auxiliary problem with one general constraint:

$$\boxed{\begin{array}{llllllll} \text{(CKLP)} & \psi & = & f'x \to \max, & & f & \ge & 0, \\ & d'x & \le & b, & & d & \ge & 0, \\ & 0 & \le & x \le x_u, & & b & \ge & 0. \end{array}}$$

Here $d \in \mathbf{R}^n$. The solution of this auxiliary problem, called the "continuous knapsack problem," is described in detail in [9, p. 517]. From the optimality conditions, and denoting the dual variable for the single constraint as ξ, it follows that

$$\forall (j : f_j < d_j \xi),\ x_j = 0; \qquad \forall (j : f_j > d_j \xi),\ x_j = x_{ju}. \tag{2.3}$$

The optimal solution will include the variables $x_{j_1}, x_{j_2}, \ldots, x_{j_p}$, ordered by the decreasing sequence f_j/d_j, such that

$$\sum_{k=1}^{p} d_{j_k} x_{j_k} = b. \tag{2.4}$$

All variables x_{j_k} except the last will be set to the upper bound x_{ju}. The last variable x_{j_p}, corresponding to f_{j_p}/d_{j_p}, becomes the basic variable and is included in the solution with an intermediate value:

$$0 \le x_{j_p} \le x_{j_p u}. \tag{2.5}$$

The value f_{j_p}/d_{j_p} will be equal to the dual variable ξ. If the basic variable is not equal to an intermediate value (degenerate case), then we shall assume that the dual variable ξ is equal to f_{j_p}/d_{j_p}, where p is the last value of the sorted index corresponding to the variable included in the solution on its upper bound. This means that $x_{j_{p+1}} = 0$.

Let us consider the problem of type (CKLP), replacing the constraint $d'x \le b$ with a single constraint of the primal problem (LPP) from row i. This problem has the form

$$\begin{cases} \varphi_i & = & f'x \to \max, \\ a_i'x & \le & l_i, \\ 0 & \le & x \le x_u. \end{cases} \tag{2.6}$$

Let the dual variable analogous to ξ be denoted by y_{iu}, and the vector of m components y_{iu} by $y_u : y_u \in \mathbf{R}^m$. We present the proof of the following theorem [13].

THEOREM 2.1. *For the pair of problems* (LPP) *and* (DLPP) *and the set of problems* (2.6) *the following inequalities hold:*

$$y_i{}^* \le y_{iu} \quad \forall (i = 1, \ldots, m), \tag{2.7}$$

where $y_i{}^*$ *is the* i*th component of the optimal solution of the dual problem* (DLPP).

Proof. Let us suppose that the opposite of inequality (2.7) holds for some i:

$$y_i^* > y_{iu}. \tag{2.8}$$

Then taking into account that A is nonnegative, one can easily check that the inequality

$$A'y^* \geq a_i y_i^* \geq a_i y_{iu} \tag{2.9}$$

must hold and that the second inequality in (2.9) is strict for any nonzero component of the vector a_i. For a nonzero a_{ij} from $a_{ij}y_{ui} \geq f_j$ it follows that

$$a_{ij}y_i^* > a_{ij}y_{iu} \geq f_j. \tag{2.10}$$

Let us denote the optimal solution of problem (2.6) as x^{ai}. Comparing the solutions of (LPP) and (2.6), one finds from (2.10) that for any zero components of x^{ai},

$$x_j^* \;=\; 0 \quad \forall\, j:\; x_j^{ai} \;=\; 0. \tag{2.11}$$

We have already noticed that the dual variable in (2.6) was chosen in such a way that for the degenerate case the basic variable $x_{j_p}^{ai}$ is set to its upper bound. The column corresponding to the basic variable of (2.6) must satisfy

$$a_{ij_p}y_i^* > a_{ij_p}y_{iu} = f_{j_p}. \tag{2.12}$$

It follows that $x_{j_p}^* = 0$ and

$$a_i'x^{ai} - a_i'x^* \geq a_{ij_p}x_{j_p}^{ai} > 0. \tag{2.13}$$

Solving (2.6) we obtain

$$a_i'x^{ai} = l_i. \tag{2.14}$$

Subtracting (2.13) from (2.14) one finds that $a_i'x^* < l_i$ and hence $y_i^* = 0$, contradicting assumption (2.8). □

Using the upper limits of the dual variables in (2.7), one can add additional box constraints to the dual problem (DLPP). Theorem 2.1 provides a link between tests on the primal and dual problems.

3. Primal tests.

3.1. Upper bound for the objective. Suppose (CKLP) is aggregated, meaning that all row constraints of (LPP) are summed with nonnegative coefficients. The aggregated problem has an equal or greater feasible set than the feasible set of the primal problem (LPP). Therefore the optimal value of the objective for the aggregated problem can be used as an upper bound for the objective of the original problem (LPP). In the simpest case, all the coefficients of aggregation are zero except the coefficient for constraint i, which is equal to 1. Another possible way to obtain such an evaluation is to use aggregation with a vector of weights y_u and to solve the corresponding aggregated problem

$$\begin{cases} \varphi_u &= f'x \to \max, \\ y_u'Ax &\leq y_u'l, \\ 0 &\leq x \leq x_u. \end{cases} \tag{3.1}$$

The objective value of any such aggregated problem is an upper bound for the objective of (LPP):

$$\varphi \leq \varphi_u, \qquad \varphi \leq \varphi_i. \tag{3.2}$$

Denoting

$$\varphi_{il} = \min_i \varphi_i, \qquad \varphi_l = \min(\varphi_{il}, \varphi_u), \tag{3.3}$$

we obtain the inequality

$$f'x \le \varphi_l. \tag{3.4}$$

This can be considered as an additional constraint of (LPP) that may be useful for detecting redundancy in row constraints.

3.2. Test 1*. Let us consider the problem

$$\begin{cases} \alpha_i^f & = & a_i'x \to \max, \\ f'x & \le & \varphi_l, \\ 0 & \le & x \le x_u. \end{cases} \tag{3.5}$$

If the test

$$\alpha_i^f < l_i \tag{3.6}$$

holds, then the ith row of A is redundant. This test is more effective if the value φ_l is nearly optimal. The test could also be useful for the basis identification problem after the optimal solution is obtained by an interior-point method, in which the optimal objective value is known almost exactly [4].

3.3. Test 2. It is important to note that a similar test could be performed not only with the objective vector f but with any positive vector; for example, it is useful to take a vector e of ones: $e \in \mathbf{R}^n$, $e_i = 1$ $(i = 1, \dots, n)$. In this case we solve the problems

$$\begin{cases} \epsilon_i & = & e'x \to \max, \\ a_i'x & \le & l_i, \\ 0 & \le & x \le x_u. \end{cases} \tag{3.7}$$

The value of ϵ_l is found as

$$\epsilon_l = \min_i \epsilon_i. \tag{3.8}$$

Now solving the problem

$$\begin{cases} \alpha_i^\epsilon & = & a_i'x \to \max, \\ e'x & \le & \epsilon_l, \\ 0 & \le & x \le x_u, \end{cases} \tag{3.9}$$

we obtain the test

$$\alpha_i^\epsilon < l_i. \tag{3.10}$$

If this inequality holds, it means that the row constraint i is redundant. The advantage of using Test 1* (3.6) with the objective vector f is the opportunity to find upper bounds for the dual variables. These are then used in the set of tests for the dual problem.

3.4. Test 3. Another set of tests can be found that uses the properties of the feasible set itself, without using additional constraints that are implied by the objective function, or by some other vector as mentioned above.

One could choose the kth row of A and solve the problem

$$\begin{cases} \alpha_i^k &= a_i' x \to \max, \\ a_k' x &\le l_k, \\ 0 &\le x \le x_u. \end{cases} \tag{3.11}$$

If the inequality

$$\alpha_i^k < l_i \tag{3.12}$$

holds, then the ith row of A is redundant.

3.5. Test 4. Instead of a single row k, the sum (aggregate) of rows could be used for this test. For example, the aggregated row constraint with the weights of upper bounds on dual variables, y_u, can be used for this purpose. Then we obtain

$$\begin{cases} \alpha_i^a &= a_i' x \to \max, \\ y_u' A x &\le y_u' l, \\ 0 &\le x \le x_u. \end{cases} \tag{3.13}$$

If the inequality

$$\alpha_i^a < l_i \tag{3.14}$$

now holds, it means that the row constraint i is redundant.

3.6. Test 5. Another approach is to choose for the aggregation the rows that have a significant number of nonzero elements and have a low value of the normalized right-hand side, defined as $l_i^n = l_i / \sqrt{\sum_{j=1}^n a_{ij}^2}$. Different heuristic procedures can be used for this normalization [28].

For the test (3.12) the most restrictive row constraint should be used, for which a good candidate is the row k from the equality

$$k = \arg\min_i \varphi_i. \tag{3.15}$$

Let us assume that

$$\varphi_l = \varphi_{il}. \tag{3.16}$$

We shall define sets L and F as

$$\begin{aligned} L &= \{x : a_k' x \le l_k;\ 0 \le x \le x_u\}, \\ F &= \{x : f' x \le \varphi_l;\ 0 \le x \le x_u\}. \end{aligned} \tag{3.17}$$

From (3.15)–(3.17) one can see that

$$L \subseteq F. \tag{3.18}$$

According to (3.18) the tests that use the constraint $a_k' x \le l_k$ are preferable.

3.7. Separation of LP problem. The redundant rows that are removed from $Ax \le l$ could have been the rows that prevent the problem from being separated into a set of problems of lower dimension. Hence, if a significant number of rows have been removed, it is worthwhile to check whether this opportunity for separation exists. If so, then all further analysis will proceed for each subproblem separately.

3.8. Klein–Holm test: A comparison. Klein and Holm [18] have suggested a test that we present now for comparison. Let us determine the functions

$$\begin{cases} S_i &= l_i - a_i'x, \\ S_{ik}^m &= S_i - S_k \to \min, \\ 0 &\le x \le x_u. \end{cases} \tag{3.19}$$

If the inequality

$$S_{ik}^m > 0 \tag{3.20}$$

holds, then the constraint i is redundant. Test (3.20) is weaker than test (3.12), meaning that it is able to detect a smaller number of redundant rows. If the inequality (3.20) is satisfied, then it implies that

$$\begin{cases} S_i - S_k \ge S_{ik}^m > 0, \\ 0 \le x \le x_u, \\ l_i - a_i'x > l_k - a_k'x \ge 0. \end{cases} \tag{3.21}$$

Inequality (3.12) follows from (3.21). The opposite statement is wrong. Let us multiply the constraint $a_k'x \le l_k$ by a small positive value ε. Then, when ε tends to zero, the term S_k in (3.19) will be less than any given positive value, and test (3.20) will tend to test (2.2), which is clearly weaker than test (3.12).

3.9. Integer and mixed-integer linear programs. Notice that all Tests 1–4 for the primal problem are also valid for the detection of row redundancy in integer or mixed-integer problems of type (LPP).

3.10. General LP problems: A discussion. The main scheme of iterative application of the primal and dual tests is based on Theorem 2.1, which assumed that all conditions of (LPP) are satisfied.

The general LP problem can be considered in the form

$$\text{(LPG)} \qquad \begin{array}{rcl} \varphi &=& f'x \to \max, \\ Ax &|& l, \\ 0 &\le& x \le x_u, \end{array}$$

where $|$ denotes $=$ or $\le$, and the components of A, l, and f can be of any sign. The set of primal tests separately (without dual tests) can be applied to (LPG) but the existence of negative coefficients may decrease the effectiveness. The evaluations are valid with some minor changes. Instead of (CKLP), the general continuous knapsack linear problem (CKLPG) must be solved without conditions of nonnegativeness of the coefficients. But (CKLPG) can be easily transformed to the form of (CKLPGT) below, which can be solved by the same algorithm as (CKLP) [9, p. 517]. There is no difference in this algorithm if some of f_j are negative, so the only remaining problem is to make all the coefficients d_j in a single row constraint nonnegative. Let us define

$$P = \{j : d_j \ge 0\}, \qquad N = \{j : d_j < 0\}.$$

The following transformation of variables is applied:

$$\begin{array}{llll} \forall j \in N, & Y_j = x_{ju} - x_j, & \psi_N = \sum_{j\in N} f_j x_{uj}, \\ Y_{uj} = x_{uj}, & 0 \le Y_j \le Y_{uj}, & b_m = b - \sum_{j\in N} d_j x_{uj}. \end{array}$$

Now (CKLPG) will be transformed to the form of (CKLPGT):

$$\boxed{\begin{array}{ll} \text{(CKLPGT)} & \psi \;=\; \sum_{j\in P} f'_j x_j - \sum_{j\in N} f'_j Y_j + \psi_N \to \max, \\ & \sum_{j\in P} d_j x_j + \sum_{j\in N} \mid dj \mid Y_j \le b_m, \\ & \forall j \in P,\ 0 \;\le x_j \;\le x_{uj}, \qquad \forall j \in N,\ 0 \;\le\; Y_j \;\le\; Y_{uj}. \end{array}}$$

One can see that in (CKLPGT) all the variables have nonnegative coefficients in a single row constraint and thus the algorithm in [9, p. 517], described in section 2, can be applied. If $b_m < 0$, then this problem is infeasible. One must also note that if the inequality

$$\sum_{j\in P} d_j x_{uj} \le b$$

is satisfied, then this constraint is redundant. This test replaces (2.2) in the case of (LPG).

In the first stage we temporarily assume that all constraints are inequality constraints of the form $\le$ (which is not a restriction, because otherwise they can be multiplied by -1). If primal tests indicate redundancy of the constraint i, and it is really an equality constraint, then it means that this constraint is not redundant but infeasible. One must hold a list of equality constraints (as described in [26]) and compare it with the list of redundant constraints generated by the primal tests. For example, if the row constraint i is an equality, then inequality (3.6), as a result of solving problem (3.5), implies infeasibility of the primal problem.

3.11. Numerical example. Here we present an example to illustrate some of the tests. The data has the following values:

$$\left\{\begin{array}{lcl} f' & = & (3,\ 2,\ 1), \\ x'_u & = & (1,\ 1,\ 2), \\ A & = & \begin{pmatrix} 1 & 5 & 20 \\ 2.5 & 3 & 1 \\ 3 & 1 & 5 \end{pmatrix}, \\ l' & = & (5,\ 6.2,\ 3.8). \end{array}\right. \tag{3.22}$$

We note that the presolver of the program LIPSOL [28] does not detect any redundancy in this problem.

Calculating the vector l_u according to (2.1), we obtain

$$l_{1u} = 46.0, \quad l_{2u} = 7.5, \quad l_{3u} = 14.0.$$

This means that the simple test (2.2) does not show any row redundancy. Applying the Klein–Holm test, we obtain

$$\begin{aligned} S_1 &= 5 \;-\; x_1 - 5x_2 - 20x_3, \\ S_2 &= 6.2 - 2.5x_1 - 3x_2 - x_3, \\ S_{21}^m &= \min_{0\le x\le x_u} (1.2 - 1.5x_1 + 2x_2 + 19x_3) = \; -0.3, \\ S_3 &= 4.5 \;-\; 3x_1 - x_2 - 5x_3, \\ S_{31}^m &= \min_{0\le x\le x_u} (-0.5 - 2x_1 + 4x_2 + 15x_3) \;=\; -2.5. \end{aligned}$$

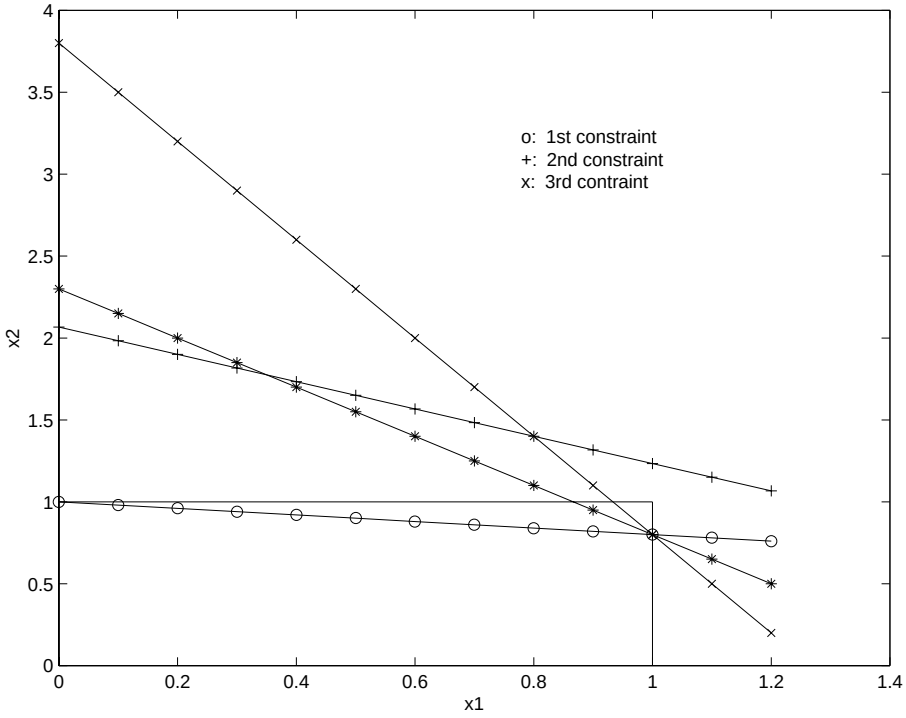


FIG. 3.1. *Constraints on the (x_1, x_2) plane. $*$ denotes upper bound for the objective.*

From (3.20) one can see that this simple test does not show any redundancy. Solving (CKLP) with rows $i = 1, 2, 3$, one obtains

$$\begin{array}{ll} \varphi_1 = 4.6, & y_{1u} = 0.4, \\ \varphi_2 = 6.1333, & y_{2u} = 0.6667, \\ \varphi_3 = 4.8, & y_{3u} = 1.0. \end{array}$$

Using aggregation with weights y_u for (3.1) we obtain

$$\begin{array}{l} y_u' A = (5.0667,\ 5.0,\ 13.6667), \quad y_u' l = 9.9333, \\ \varphi_u = 4.9467. \end{array}$$

As a result, we obtain from (3.3) that

$$\varphi_l = 4.6, \quad k = \arg\min_i \varphi_i = 1.$$

The projections of the constraints in the plane (x_1, x_2) together with the upper bound on the objective, $\varphi_l \geq f'x$, are shown in Figure 3.1.

Notice that the objective for the aggregated equation with weights y_u, namely $\varphi_u = 4.9467$, is very close to the best evaluation, $\varphi_l = 4.6$. Now solving problem (3.5), and using in sequence the rows $i = 2, 3$ as the objective, one obtains

$$\begin{array}{lcl} \alpha_2^f = 5.5 & < & l_2 = 6.2, \\ \alpha_3^f = 12.6 & > & l_3 = 3.8. \end{array}$$

This means that according to test (3.6), the row constraint with $i = 2$ is redundant, while the redundancy of row constraint $i = 3$ is not detected.

Although we already know that row $i = 2$ is redundant, let us demonstrate how the other tests proceed. Let us use test (3.12). The row constraint $k = 1$ here is used as a single row constraint in problem (3.11). As a result one obtains

$$\begin{aligned} \alpha_2^1 = 4.9 &< l_2 = 6.2, \\ \alpha_3^1 = 4.0 &> l_3 = 3.8. \end{aligned}$$

This test also shows that row $i = 2$ is redundant. Now let us check test (3.10), which is similar to (3.6) but uses the vector e of ones instead of the objective vector f, as mentioned above. From (3.7) we obtain

$$\epsilon_1 = 1.8, \qquad \epsilon_2 = 3.5667, \qquad \epsilon_3 = 1.9333,$$

which gives the minimal value for $i = 1$, namely,

$$\epsilon_l = \min_i \epsilon_i = 1.8, \qquad \arg\min_i \epsilon_i = 1.$$

Now solving (3.9) for rows $i = 2, 3$, one obtains

$$\alpha_2^\epsilon = 5 < l_2 = 6.2, \qquad \alpha_3^\epsilon = 9 > l_3 = 3.8.$$

Thus row $i = 2$ is again detected as redundant according to test (3.10). Also test (3.14) based on the single aggregated constraint can be used. Solving (3.13) with rows $i = 2, 3$, one obtains

$$\begin{aligned} \alpha_2^a = 5.4342 &< l_2 = 6.2, \\ \alpha_3^a = 4.7805 &> l_3 = 3.8. \end{aligned}$$

This means that row $i = 2$ is redundant according to this test as well. The example is continued below with dual tests.

4. Dual tests.

4.1. Auxiliary inequalities*. Similar tests can be performed for the dual problem by evaluating the columns of A and detecting two types of column redundancy: columns that correspond to zero variables and columns that correspond to variables attaining their upper bounds. For the optimal values of the dual variables the following inequality holds:

$$l'y + x'_u u \le \varphi_l. \tag{4.1}$$

This inequality is the corollary of the equality of the optimal values of the objective for the primal and dual problems (see the duality theorem of [9]). Multiplying the inequality in (DLPP) by the vector x_u and summing, one obtains

$$y' l_u + u' x_u \ge f' x_u. \tag{4.2}$$

From (4.1) and (4.2) it follows that $y'(l_u - l) \ge f' x_u - \varphi_l$. From (4.1) it also follows that $l'y \le \varphi_l$. Thus we have obtained two inequalities for the dual variables y, without the dual variables u.

4.2. Test 6*. Let us consider the problem

$$\begin{cases} \eta_{jl} &= s_j' y \to \min, \\ (l_u - l)' y &\geq f' x_u - \varphi_l, \\ 0 &\leq y \leq y_u. \end{cases} \tag{4.3}$$

If

$$\eta_{jl} > f_j, \tag{4.4}$$

then, taking into account that u is nonnegative, it follows from (DLPP) that $x_j = 0$ and the jth column is redundant.

4.3. Test 7*. In a similar way, the following problem could be considered:

$$\begin{cases} \eta_{ju} &= s_j' y \to \max, \\ l' y &\leq \varphi_l, \\ 0 &\leq y \leq y_u. \end{cases} \tag{4.5}$$

If

$$\eta_{ju} < f_j, \tag{4.6}$$

then $u_j > 0$ and $x_j = x_u$, from which another type of column redundancy follows. Before these problems are solved, a simple test to find the columns that satisfy the condition

$$s_j' y_u < f_j \tag{4.7}$$

must be performed. Inequality (4.7) means that the primal variable x_j is equal to its upper bound. This test also automatically detects the empty columns that could have been produced during the previous iterations of the reduction tests.

Removing the column that corresponds to the variable that is fixed on its upper bound implies the recalculation of vectors l and l_u and also of the value φ_l.

4.4. Test 8*. One more dual test can be suggested. First the value of μ is determined by solving

$$\begin{cases} \mu &= l' y \to \min, \\ (l_u - l)' y &\geq f' x_u - \varphi_l, \\ 0 &\leq y \leq y_u. \end{cases} \tag{4.8}$$

Then the following set of problems must be solved, one for each column j:

$$\begin{cases} \eta_{jl}^m &= s_j' y \to \min, \\ l' y &\geq \mu, \\ 0 &\leq y \leq y_u. \end{cases} \tag{4.9}$$

If

$$\eta_{jl}^m > f_j, \tag{4.10}$$

then the variable x_j is fixed at zero value, and this column is redundant.

Let us define the sets U and M as follows:

$$\begin{aligned} U &= \{y : (l_u - l)' y \geq f' x_u - \varphi_l,\ 0 \leq y \leq y_u\}, \\ M &= \{y : l' y \geq \mu,\ 0 \leq y \leq y_u\}. \end{aligned}$$

Taking (4.8) into account, one can easily see that $U \subset M$. It follows that test (4.10) is less effective than test (4.4), but numerically it is somewhat cheaper.

4.5. Test 9. Another test can be suggested. This test uses the vector p of ones: $p \in \mathbf{R}^m,\ p_j = 1\ (j = 1, \dots, m)$. A lower bound for $p'y$ is calculated from

$$\begin{cases} \gamma_l &= p'y \to \min, \\ (l_u - l)'y &\geq f'x_u - \varphi_l, \\ 0 &\leq y \leq y_u. \end{cases} \tag{4.11}$$

Now the lower bound of $s_j'y$ is calculated from

$$\begin{cases} \eta_{jl}^{\gamma} &= s_j'y \to \min, \\ p'y &\geq \gamma_l, \\ 0 &\leq y \leq y_u. \end{cases} \tag{4.12}$$

The test has the form

$$\eta_{jl}^{\gamma} > f_j. \tag{4.13}$$

If this is satisfied, variable x_j must be set to its zero bound and column j must be removed from the matrix A. The vectors $f,\ x_u$ must be changed accordingly.

4.6. Test 10. An upper bound on $p'y$ can be calculated from

$$\begin{cases} \gamma_u &= p'y \to \max, \\ l'y &\leq \varphi_l, \\ 0 &\leq y \leq y_u. \end{cases} \tag{4.14}$$

An upper bound on $s_j'y$ is calculated similarly:

$$\begin{cases} \eta_{ju}^{\gamma} &= s_j'y \to \max, \\ p'y &\leq \gamma_u, \\ 0 &\leq y \leq y_u. \end{cases} \tag{4.15}$$

This test has the form

$$\eta_{ju}^{\gamma} < f_j. \tag{4.16}$$

If this is satisfied, variable x_j must be set to its upper bound. It is clear that removing redundant columns can cause separation of the original LP problem. As soon as a significant number of redundant columns is removed, this possibility should be checked.

4.7. Numerical example (continuation). Using the numerical example from section 3.11 and taking into account that the redundancy of row $i = 2$ was detected, we can assume that $y_2 = 0$. Let us see what follows from the dual tests. According to (4.7) we obtain

$$\begin{cases} s_1'y_u &= 1 \cdot 0.4 + 3 \cdot 1 &= 3.4 &> f_1 &= 3, \\ s_2'y_u &= 5 \cdot 0.4 + 1 \cdot 1 &= 3 &> f_2 &= 2, \\ s_3'y_u &= 20 \cdot 0.4 + 5 \cdot 1 &= 13 &> f_3 &= 1. \end{cases} \tag{4.17}$$

Thus we have not detected any x_j that must be set to its upper bound.

According to (4.3) we obtain

$$
\left\{\begin{array}{rcl}
l_{1u}-l_1 &=& 41, \quad l_{3u}-l_3 \;=\; 10,\\
f'x_u-\varphi_l &=& 3\cdot 1+2\cdot 1+1\cdot 2-4.6 \;=\; 2.4,\\
41y_1+10y_3 &\geq& 2.4,\\
0 \;\leq\; y_1 \;\leq\; y_{1u} &=& 0.4, \quad 0 \;\leq\; y_3 \;\leq\; 1,\\
\eta_{1l} \;=\; 0.0585 &<& f_1 \;=\; 3,\\
\eta_{2l} \;=\; 0.2353 &<& f_2 \;=\; 2,\\
\eta_{3l} \;=\; 1.17 &>& f_3 \;=\; 1.
\end{array}\right. \tag{4.18}
$$

From test (4.4) and the last inequality of (4.18) it follows that variable x_3 is redundant—it must be set to zero. Solving problem (4.5) and checking tests (4.6), we obtain

$$
\left\{\begin{array}{rcl}
5y_1 \;+\; 3.8y_3 &\leq& \varphi_l \;=\; 4.6,\\
\eta_{1u} \;=\; 3.16 &>& f_1 \;=\; 3,\\
\eta_{2u} \;=\; 2.6842 &>& f_2 \;=\; 2.
\end{array}\right. \tag{4.19}
$$

These tests do not detect any variables that must be set at their upper bounds. Now only variables x_1, x_2 remain, and the next iteration of the primal tests can be performed. Notice that row $i = 2$ and column $j = 3$ have been removed. Checking test (2.2), we obtain

$$
\begin{array}{rcl}
l_{1u} &=& 1\cdot 1 \;+\; 5\cdot 1 \;=\; 6 \;>\; l_1 \;=\; 5,\\
l_{3u} &=& 3\cdot 1 \;+\; 1\cdot 1 \;=\; 4 \;>\; l_3 \;=\; 3.8.
\end{array}
$$

This simple test does not show any further row redundancy. Solving (2.6), with already reduced dimensions ($i = 1, 3$; $j = 1, 2$), and (3.1) we obtain

$$
\begin{array}{rclrcl}
\varphi_1 &=& 4.6, & y_{1u} &=& 0.4,\\
\varphi_3 &=& 4.8, & y_{3u} &=& 1,\\
\varphi_u &=& 4.6, & \varphi_l &=& 4.6,
\end{array}
$$

which means that the value φ_l has remained unchanged. Solving (3.5) for $i = 3$, we obtain

$$
\alpha_3^f \;=\; 3.8 \;\leq\; l_3 \;=\; 3.8. \tag{4.20}
$$

Thus we see that the row inequality $i = 3$ is redundant. This implies that only the single row inequality $i = 1$ remains, and the optimal solution corresponds to the objective value $\varphi^* = \varphi_l = 4.6$, as found earlier.

5. Numerical experiments. We designed a toolbox, IVITEST (using the MATLAB language [22]), to demonstrate the algorithms. (The toolbox together with test problems will be sent by e-mail on request.) It consists of two main programs: ivitest5.m for (LPP) and ivitest5n.m for general LP problems (see section 3.10). In the first program both primal and dual tests proceed iteratively indicating redundant rows and columns, and in the second program only primal tests are applied to detect redundant and infeasible rows. Two iterations are applied to each problem.

A special set of random nonsparse data was used for the numerical experiments. First a random matrix and objective were generated and then some data were modified to provide more stiff constraints and more unequal coefficients in the objective. The results were compared with the solution obtained using the primal simplex method

TABLE 5.1
Results of numerical experiments (MATLAB/CPLEX).

Name	Nonnegative	Size	Basic var.	k1	k2	k3	k5
primer2000	yes	2000×100	3	1993	42	40	-
primer6000	yes	6000×100	5	5991	41	40	-
primer6200	yes	6000×200	4	5990	53	40	-
primer12200	yes	12000×200	51	11990	51	40	-
primer2000nr	no	2000×100	-	1891	-	-	101
primer6000n	no	6000×100	5	5690	-	-	0

by (CPLEX) [26]. In Table 5.1 we show the name of the problem, its type and size. Next, the number of basic variables (from CPLEX), k1, and k2, k3, or k5 are shown. Here k1 is the list of redundant row constraints indicated by the tests, k2 is the list of indicated variables on the zero bound, k3 is the list of indicated variables on the upper bound, k5 is the list of indicated infeasible constraints. Altogether six problems were tested: four of (LPP) type, one with some negative coefficients and some infeasible constraints (primer2000nr), and one with some number of negative coefficients and redundant row constraints (primer6000n). We noticed that for the feasible problems the upper bound of the objective φ_l that was found by our tests was very close to the optimal value found by CPLEX (not shown in the table). One can see that these problems are very redundant and the redundancy is detected by the proposed tests rather well.

6. Robust analysis of LP problems. It is well known that the input data are not exact for most large-scale problems. The usual situation is that each data item is given in some range, e.g., as a relative deviation from the given value. Of course there are some exactly known parameters (e.g., 0 or 1), for which the upper and lower ranges coincide. Now we have the important problem, to develop scale reduction algorithms that are robust with respect to the uncertainty of the input data. All the evaluations must be valid while the parameters of the problem (matrix and objective coefficients, values of bounds, etc.) are known only within some given range.

Let us consider the set of (LPP) problems including the bounds of uncertainty for the data. They have the following form:

$$\text{(RLPP)} \qquad \begin{array}{rcl} \varphi & = & f'x \to \max, \\ Ax & \le & l, \\ 0 & \le & x \le x_u, \end{array} \qquad \begin{array}{ccccccc} \overline{f} & \ge & f & \ge & \underline{f} & \ge & 0, \\ \overline{A} & \ge & A & \ge & \underline{A} & \ge & 0, \\ \overline{l} & \ge & l & \ge & \underline{l} & \ge & 0, \\ \overline{x}_u & \ge & x_u & \ge & \underline{x}_u & \ge & 0, \end{array}$$

where the matrix $\underline{A}$ consists of elements $\underline{a}_{ij}$, and the matrix $\overline{A}$ consists of elements $\overline{a}_{ij}$, and where the inequalities should be interpreted elementwise. A set of (DLPP) problems (RDLPP) is formed accordingly. For this set of LP problems the method must be modified. Usually the matrix coefficients a_{ij} are the most uncertain. There is a waste of expensive work to find more exact values for data that will not be used and actually not needed for the solution of the problem. The reduction of the size of the problem as a result of the robust presolving tests permits us to detect the potentially nonredundant input data for which more certainty might be needed. For the following analysis we need to evaluate the inequalities for the interval sets of the data in (RLPP). The related problems of interval analysis were investigated in [23, 24, 19].

7. Robust primal tests.

7.1. Simple robust test SR. First let us look at the simple test (2.2). In order for it to be valid for all sets of data in (RLPP), it must be modified to the form

$$\begin{aligned} \bar{l}_u &= \overline{A}\overline{x}_u, \\ \bar{l}_{iu} &< \underline{l}_i. \end{aligned}$$

This test gives sufficient conditions for redundancy of the row constraint i for all (LPP) that are included in the set (RLPP).

7.2. Test 1R. Let us now consider the robust variant of test (3.6). This test involves an inequality, where we must replace the left-hand side by its upper bound and the right-hand part by its lower bound. Thus it has the following form:

$$\overline{\alpha}_i^f < \underline{l}_i, \tag{7.1}$$

where the value $\overline{\alpha}_i^f$ is the upper bound for α_i^f that must be found. $\overline{\alpha}_i^f$ is a solution of the auxiliary (CKLP) problem, where the single row constraint $f'x \le \varphi_l$ must correspond to the widest feasible set according to (RLPP), and the objective $\alpha_i^f = a_i'x \to \max$ must correspond to the upper bound on the set of objective vectors. Thus we obtain

$$\begin{cases} \overline{\alpha}_i^f &= \overline{a}_i'x \to \max, \\ \underline{f}'x &\le \overline{\varphi}_l, \\ 0 &\le x \le \overline{x}_u. \end{cases} \tag{7.2}$$

Here the value $\overline{\varphi}_l$ is not yet defined and must be found as the upper bound on φ_l (3.3). This value in turn is the minimum of the set of upper bounds on values of the objective of (LPP), namely, φ_i (2.6) and φ_u (3.1). All of these upper bounds must be found for the complete set of (LPP) problems according to (RLPP). The value of the upper bound on φ_i can be found in the form

$$\begin{cases} \overline{\varphi}_i &= \overline{f}'x \to \max, \\ \underline{a}_i'x &\le \bar{l}_i, \\ 0 &\le x \le \overline{x}_u. \end{cases} \tag{7.3}$$

The vector of dual variables found from the solutions of the set of problems (7.3) is denoted by y^r. Its components do not, in general, constitute upper bounds on the dual variables for the full set (RLPP), in contrast to the certain case. This vector can be used to find an aggregated upper bound on the objective of the set (RLPP). One obtains

$$\begin{cases} \overline{\varphi}_u &= \overline{f}'x \to \max, \\ y^{r\prime}\underline{A}x &\le y^{r\prime}\bar{l}, \\ 0 &\le x \le \overline{x}_u. \end{cases} \tag{7.4}$$

Now the values $\overline{\varphi}_{il}$ and $\overline{\varphi}_l$ can be found as

$$\overline{\varphi}_{il} = \min_i(\overline{\varphi}_i), \qquad \overline{\varphi}_l = \min(\overline{\varphi}_{il}, \overline{\varphi}_u). \tag{7.5}$$

With this value $\overline{\varphi}_l$, (7.2) can be solved and the robust test (7.1) can be checked. If this inequality holds, then row i is redundant for the full set (RLPP). We note that other coefficients of aggregation can also be used in the calculation of $\overline{\varphi}_u$, and a test similar to (7.1) will still be robust.

7.3. Test 2R. Using the vector e of ones we can obtain a robust analogue of test (3.10) in the form

$$\overline{\alpha}_i^{\epsilon} \ < \ \underline{l}_i. \tag{7.6}$$

Inequality (7.6) implies redundancy of the row constraint i for the full set (RLPP). Here the value $\overline{\alpha}_i^{\epsilon}$ is found by solving the (7.7)–(7.9), which follows.

First the values $\overline{\epsilon}_i$ must be found by solving the set of problems

$$\begin{cases} \overline{\epsilon}_i & = & e'x \to \max, \\ \underline{a}_i x & \leq & \overline{l}_i, \\ 0 & \leq & x \ \leq \ \overline{x}_u. \end{cases} \tag{7.7}$$

The values $\overline{\epsilon}_i$ are upper bounds on the value $e'x$ for the set of inequalities in (RLPP). The least of these upper bounds can be found as

$$\overline{\epsilon}_l \ = \ \min_i \overline{\epsilon}_i. \tag{7.8}$$

The value $\overline{\alpha}_i^{\epsilon}$ can now be found from the problem

$$\begin{cases} \overline{\alpha}_i^{\epsilon} & = & \overline{a}_i' x \to \max, \\ e'x & \leq & \overline{\epsilon}_l, \\ 0 & \leq & x \ \leq \ \overline{x}_u. \end{cases} \tag{7.9}$$

This value constitutes an upper bound on the left-hand side of the row constraint i for the set of inequalities in (RLPP). Thus all values that are needed for test (7.6) are now found.

The row constraints that are detected as redundant according to one of the robust tests described above can be removed from (RLPP).

7.4. Test 3R. The robust analogue of test (3.12) has the form

$$\overline{\alpha}_i^{k} \ < \ \underline{l}_i. \tag{7.10}$$

Here the value $\overline{\alpha}_i^{k}$ is the solution of the problem

$$\begin{cases} \overline{\alpha}_i^{k} & = & \overline{a}_i' x \to \max, \\ \underline{a}_k' x & \leq & \overline{l}_k, \\ 0 & \leq & x \ \leq \ \overline{x}_u. \end{cases} \tag{7.11}$$

7.5. Test 4R. When the aggregated constraint for the set of inequalities in (RLPP) is calculated, it is also of use for another robust test of redundancy on row i. This test presents the robust analogue of test (3.14), and it can be used in the form

$$\overline{\alpha}_i^{a} \ < \underline{l}_i. \tag{7.12}$$

Here the value $\overline{\alpha}_i^{a}$ is the solution of the problem

$$\begin{cases} \overline{\alpha}_i^{a} & = & \overline{a}_i' x \to \max, \\ y^{r\prime} \underline{A} x & \leq & y^{r\prime} \overline{l}, \\ 0 & \leq & x \ \leq \ \overline{x}_u. \end{cases} \tag{7.13}$$

8. Robust dual tests.

8.1. Inequalities for uncertain duals. Dual problems for each (LPP) in the set (RLPP) constitute the set (RDLPP). For all of them the inequalities (4.1) and (4.2) must be satisfied. From this it follows that

$$\begin{cases} \bar{l}_u &= \overline{A}\overline{x}_u, \\ \underline{l}'y + x'_u u &\le \overline{\varphi}_l, \\ y'\bar{l}_u + x'_u u &\ge \underline{f}'\underline{x}_u. \end{cases} \tag{8.1}$$

Combining the second and third inequalities from (8.1), one obtains

$$y'(\bar{l}_u - \underline{l}) \ \ge \ \underline{f}'\underline{x}_u - \overline{\varphi}_l. \tag{8.2}$$

From the second inequality in (8.1) it also follows that

$$\underline{l}'y \ \le \ \overline{\varphi}_l. \tag{8.3}$$

One can see that (8.2) and (8.3) follow from (DLPP) and (RLPP).

8.2. Test 6R. Using inequality (8.2) we can solve the problem

$$\begin{cases} \underline{\eta}_{jl} &= \underline{s}'_j y \to \min, \\ y'(\bar{l}_u - \underline{l}) &\ge \underline{f}'\underline{x}_u - \overline{\varphi}_l, \\ 0 &\le y. \end{cases} \tag{8.4}$$

Now we obtain the robust test

$$\underline{\eta}_{jl} \ > \ \overline{f}_j. \tag{8.5}$$

If (8.5) is satisfied, then variable x_j must be set to zero for all problems in (RLPP), and column j can be removed.

8.3. Test 7R. Another robust test can be obtained by solving the problem

$$\begin{cases} \overline{\eta}_{ju} &= \overline{s}'_j y \to \max, \\ \underline{l}'y &\le \overline{\varphi}_l, \\ 0 &\le y. \end{cases} \tag{8.6}$$

This test has the form

$$\overline{\eta}_{ju} \ < \ \underline{f}_j. \tag{8.7}$$

If (8.7) is satisfied, then variable j must be set to its upper bound for all problems in (RLPP). It means that for the next step of presolving analysis, column j must be removed, $\underline{s}_j\underline{x}_{ju}$ must be subtracted from $\bar{l}$, and $\overline{s}_j\overline{x}_{ju}$ must be subtracted from $\underline{l}$ in (RLPP).

8.4. Test 8R. Similarly to (4.8), another set of robust dual tests can be derived. First a lower bound for $l'y$ can be found by solving

$$\begin{cases} \underline{\mu} &= \underline{l}'y \to \min, \\ (\bar{l}_u - \underline{l})'y &\ge \underline{f}'\underline{x}_u - \overline{\varphi}_l, \\ 0 &\le y. \end{cases} \tag{8.8}$$

Now the following set of problems can be solved, one for each column j:

$$\begin{cases} \underline{\eta}^m_{jl} & = & \underline{s}'_j y \to \min, \\ \bar{l} y & \geq & \underline{\mu}, \\ 0 & \leq & y. \end{cases} \tag{8.9}$$

If

$$\underline{\eta}^m_{jl} \; > \; \overline{f} j \tag{8.10}$$

holds, it means that column j must be set to zero for the whole set (RLPP). This test can be used instead of test (8.5), though it is clearly weaker. It does not have the advantage of test (8.5) because in the robust case the single row constraint for the maximization (8.6) and for the minimization (8.8) have different coefficients.

8.5. Test 9R. Instead a cheaper test can be used, based on the vector p of ones. First a lower bound on $p'y$ is calculated from

$$\begin{cases} \underline{\gamma}_l & = & p'y \to \min, \\ (\bar{l}_u - \underline{l})' y & \geq & \underline{f}' \underline{x}_u - \overline{\varphi}_l, \\ 0 & \leq & y. \end{cases} \tag{8.11}$$

Now a lower bound for column j is calculated from

$$\begin{cases} \underline{\eta}^\gamma_{jl} & = & \underline{s}'_j y \; \to \min, \\ p' y & \geq & \underline{\gamma}_l, \\ 0 & \leq & y. \end{cases} \tag{8.12}$$

The test has the form

$$\underline{\eta}^\gamma_{jl} \; > \; \overline{f}_j. \tag{8.13}$$

If (8.13) is satisfied, then variable j must be set to its zero bound for all of (RLPP).

8.6. Test 10R. An upper bound for the artificial constraint can also be calculated in the form

$$\begin{cases} \overline{\gamma}_u & = & p'y \to \max, \\ \underline{l}' y & \leq & \overline{\varphi}_l, \\ 0 & \leq & y. \end{cases} \tag{8.14}$$

Now the upper bound for column j is calculated from

$$\begin{cases} \overline{\eta}^\gamma_{ju} & = & \overline{s}'_j y \to \max, \\ p' y & \leq & \overline{\gamma}_u, \\ 0 & \leq & y. \end{cases} \tag{8.15}$$

This test has the form

$$\overline{\eta}^\gamma_{ju} \; < \; \underline{f}_j. \tag{8.16}$$

If (8.16) is satisfied, variable j must be set to its upper bound for all of (RLPP).

9. Conclusions. In this paper we have presented a new set of presolving tests for both primal and dual large-scale LP problems with zero or positive coefficients. The tests do not replace "simple tests" (see [1]) but should be done in addition, resulting in a more significant size reduction. They are based on the inherent properties of large-scale LP problems, rather than on exceptions like empty rows or singletons. Robust tests are suggested for evaluation in the case of uncertain input data. The remaining data for the reduced size problem may be obtained anew, thus reducing uncertainty in the final solution.

Acknowledgments. This paper was written during my visit to the Division of Optimization and System Theory, Department of Mathematics, Royal Institute of Technology, Stockholm, Sweden. I am grateful to the participants in the seminar of the Division for their attention and interest. I especially wish to thank Professor Anders Lindquist, Professor Per-Olof Gutman, Dr. Erling D. Andersen, and Dr. Anders Forsgren for fruitful discussions. I am very grateful to Professor Michael Saunders for his useful suggestions and remarks during the process of revising this paper.

REFERENCES

[1] E. D. Andersen and K. D. Andersen, *Presolving in linear programming*, Math. Programming, 71 (1995), pp. 221–245.

[2] E. D. Andersen, J. Gondzio, J. Meszaros, and X. Xu, *Implementation of Interior Point Methods for Large Scale Linear Programming*, Tech. report 1996.3, Logilab, HEC Geneva, Section of Management Studies, University of Geneva, Switzerland, 1996.

[3] E. D. Andersen, *Finding all linearly dependent rows in large scale linear programming*, Optim. Methods Softw., 6 (1995), pp. 219–227.

[4] E. D. Andersen, *On exploiting problem structure in a basis identifications procedure for linear programming*, Publications from Department of Management, No 6/1996, Odense University, Odense, Denmark, 1996.

[5] A. L. Brearly, G. Mitra, and H. P. Williams, *Analysis of mathematical programming problems prior to applying the simplex algorithm*, Math. Programming, 8 (1975), pp. 54–83.

[6] A. Brooke, D. Kendrick, and A. Meerus, *GAMS—A Users Guide*, The Scientific Press, Redwood City, CA, 1988.

[7] T. F. Carleman, *Large Sparse Numerical Optimization*, Springer-Verlag, Berlin, 1984.

[8] T. H. Cormen, R. L. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, The MIT Press, Cambridge, MA, 1990.

[9] G. B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, Princeton, NJ, 1963.

[10] L. M. Dudkin, I. Rabinovich, and I. Vakhutinsky, *Iterative Aggregating Theory*, Dekker, New York, 1987.

[11] J. Gondzio, *Presolve Analysis of Linear Programs Prior to Applying an Interior Point Method*, Tech. report 1994.3, Logilab, Section of Management Studies, University of Geneva, Switzerland, 1994.

[12] D. M. Himmelblau, *Decomposition of Large-Scale Problems*, American Elsevier, New York, 1973.

[13] I. V. Ioslovich and Y. M. Makarenkov, *On methods of dimensionality reduction in linear programming*, Econ. Math. Methods, 11 (1975), pp. 316–324 (in Russian).

[14] I. Ioslovich, *Robust Reduction of Large Scale Linear Programming Problems*, Tech. report TRITA/MAT-96-OS6, ISSN 0348-405X, Department of Mathematics, Royal Institute of Technology, Stockholm, Sweden, 1996.

[15] I. Ioslovich, *IVITEST—A Toolbox for Redundancy Determination in LP Problems, for Use with MATLAB* 5.2. *Users Guide*, Research report TRITA/MAT-99-OS04, Department of Mathematics, Royal Institute of Technology, Stockholm, Sweden, 1999.

[16] I. Ioslovich, *Numerical software for redundancy determination and presolving analysis of large-scale linear programming problems, using MATLAB* 5.2, in Proceedings of the Second NICONET Workshop: Numerical Control Software, INRIA, Rocquencourt, France, pp. 67–71.

[17] M. H. Karwan, V. Lotfi, J. Telgen, and S. Zionts, *Redundancy in Mathematical Programming*, Springer-Verlag, Berlin, 1983.
[18] D. Klein and S. Holm, *Some reduction of linear programs using bounds on problem variables*, in Redundancy in Mathematical Programming, M. H. Karwan, V. Lotfi, J. Telgen, and S. Zionts, eds., Springer-Verlag, Berlin, 1983.
[19] V. M. Kuntzevich and M. Lychak, *Guaranteed Estimates, Adaptation, and Robustness in Control Systems*, Lecture Notes in Control and Inform. Sci. 169, Springer-Verlag, New York, 1992.
[20] L. S. Lasdon, *Optimization Theory for Large Systems*, MacMillan, London, UK, 1973.
[21] I. S. Litvinchev, *Evaluations of suboptimality of aggregation in convex programming*, Comput. Math. Math. Phys., 33 (1993), pp. 1007–1015.
[22] *MATLAB, The Language of Technical Computing. Using MATLAB*, The MathWorks Inc., Natick, MA, 1996.
[23] R. E. Moore, *Interval Analysis*, Prentice–Hall, Englewood Cliffs, NJ, 1966.
[24] K. Nickel, *Interval Mathematics*, Lecture Notes in Comput. Sci. 212, Springer-Verlag, Berlin, 1985.
[25] D. F. Rogers, R. D. Plante, R. T. Wong, and J. R. Evans, *Aggregation and disaggregation techniques and methodology in optimization*, Oper. Res., 39 (1991), pp. 553–558.
[26] *Using the CPLEX Callable Library*, ILOG, Incline Village, NV, 1997.
[27] D. B. Yudin and E. G. Goldstein, *Linear Programming, Theory, Methods and Applications*, Nauka, Moscow, 1969 (in Russian).
[28] Y. Zhang, *Solving Large Scale Linear Programs by Interior-Point Methods under the MATLAB Environment*, Tech. report TR96-01, Department of Mathematics and Statistics, University of Maryland Baltimore County, Baltimore, MD, 1996.

SIAM J. OPTIM.
Vol. 12, No. 2, pp. 283–302

ANALYSIS OF INEXACT TRUST-REGION SQP ALGORITHMS*

MATTHIAS HEINKENSCHLOSS† AND LUÍS N. VICENTE‡

Abstract. In this paper we extend the design of a class of composite-step trust-region SQP methods and their global convergence analysis to allow inexact problem information. The inexact problem information can result from iterative linear system solves within the trust-region SQP method or from approximations of first-order derivatives. Accuracy requirements in our trust-region SQP methods are adjusted based on feasibility and optimality of the iterates. Our accuracy requirements are stated in general terms, but we show how they can be enforced using information that is already available in matrix-free implementations of SQP methods. In the absence of inexactness our global convergence theory is equal to that of Dennis, El-Alem, and Maciel [*SIAM J. Optim.*, 7 (1997), pp. 177–207]. If all iterates are feasible, i.e., if all iterates satisfy the equality constraints, then our results are related to the known convergence analyses for trust-region methods with inexact gradient information for unconstrained optimization.

Key words. nonlinear programming, trust-region methods, inexact linear systems solvers, Krylov subspace methods, optimal control

AMS subject classifications. 49M37, 90C06, 90C30, 90C55

PII. S1052623499361543

1. Introduction.

In this paper we study a class of trust-region sequential quadratic programming (SQP) algorithms for the solution of minimization problems with nonlinear equality constraints. Our aim is to extend the design of these algorithms and their convergence theory to allow the use of inexact problem information that originates from inexact first-order derivative information or from the use of inexact linearized constraint equation or adjoint equation solves.

The problems we are interested in are of the form

$$\begin{aligned} &\min && f(y,u) \\ &\text{subject to (s.t.)} && C(y,u) = 0, \end{aligned} \tag{1.1}$$

where $y \in \mathbb{R}^m$, $u \in \mathbb{R}^{n-m}$, $f : \mathbb{R}^n \longrightarrow \mathbb{R}$, $C : \mathbb{R}^n \longrightarrow \mathbb{R}^m$, $m < n$. Our theory assumes that f and C are at least twice continuously differentiable. Variants of the algorithms, however, require only first-order derivative information. Our research is motivated by discretized optimal control problems [16, 18, 21], parameter identification problems and inverse problems [28, 31], and design optimization [4, 24]. In these applications, y represents the discretized state variables, u represents the discretized controls, parameters, or design variables, respectively, and the nonlinear constraint $C(y,u) = 0$ is the discretized state equation. For many of the above-mentioned applications, the solution of linear equations of the type

$$C_y(y,u)z = d \quad \text{or} \quad C_y(y,u)^T z = d, \tag{1.2}$$

*Received by the editors September 20, 1999; accepted for publication (in revised form) May 10, 2001; published electronically November 7, 2001.

http://www.siam.org/journals/siopt/12-2/36154.html

†Department of Computational and Applied Mathematics, Rice University, Houston, TX 77005–1892 (heinken@caam.rice.edu). This author's work was supported by the DoE under grant DE–FG03–95ER25257 and by CRPC CCR–9120008.

‡Departamento de Matemática, Universidade de Coimbra, 3001–454 Coimbra, Portugal (lvicente@mat.uc.pt). This author's work was supported by Centro de Matemática da Universidade de Coimbra, FCT, and Praxis XXI 2/2.1/MAT/346/94.

where y, u, and d are given and where $C_y(y,u)$ and $C_u(y,u)$ are the partial Jacobians with respect to y and u, respectively, is costly and has to be accomplished by iterative methods. In optimal control, parameter identification, or optimal design problems, equations (1.2) are related to the linearized state equations and the adjoint equations, respectively, and it is often desirable to solve such equations using application-specific methods such as Krylov subspace, multigrid, or domain decomposition methods. Hence exact solutions of linear systems (1.2) are not available; only approximate solutions with a specified residual tolerance can be obtained.

Composite-step trust-region SQP methods are used successfully to solve large scale optimization problems. However, existing convergence theories, which are nicely reviewed in [5], rely on the exact solution of linear systems of the form (1.2). Most existing implementations of SQP methods use dense or sparse linear algebra methods to accomplish the linear system solves. As we have mentioned before, this is not feasible for several of the applications we have in mind. Our main motivation in this paper is the control of inexactness arising from iterative system solves (1.2) in composite-step trust-region SQP methods. However, our assumptions on the inexactness are more general and cover inexact first-order derivative information. The novel aspect of our work is the ability to deal with inexact first-order derivative information or inexact linearized constraint equation solves. Of course, we also allow the inexact solution of trust-region subproblems, which is a standard ingredient of trust-region convergence theories and implementations.

In the context of Newton methods for nonlinear equations and unconstrained optimization, the control of inexactness is relatively well understood; see, e.g., [2, 7, 12, 13, 14, 25]. Generalizations of the inexact Newton method concepts to the local convergence analysis of inexact SQP methods can be found, e.g., in [8, 9, 15, 22, 26]. In [23] global convergence of line-search reduced SQP methods is studied. The influence of inexact problem information on the global convergence of trust-region SQP methods, however, is to our knowledge not yet studied. Our analysis and our assumptions on inexactness are different from those of [23]. In particular, our bounds on the inexactness do not rely on Lipschitz constants, derivative bounds, and other quantities that are difficult to obtain in practice. Our bounds on the inexactness depend on quantities that are readily available in our algorithms.

We give a global convergence analysis of a class of composite-step trust-region SQP algorithms for (1.1), which are reviewed in [5, section 15.4] and [10, section 4]. In the absence of inexactness, our global convergence theory is that of [10]. If all iterates are feasible, i.e., if all iterates satisfy $C(y_k, u_k) = 0$, then our results are related to the convergence analyses in [3, 5] for trust-region methods with inexact function and gradient information for unconstrained optimization.

This paper is organized as follows. In section 2 we will consider the reduced problem $\min f(y(u), u)$ obtained from (1.1) by eliminating the variables y. We will briefly discuss the convergence analyses in [3] and [5, sections 8.4, 10.6] for trust-region methods with inexact function or gradient information for the reduced problem. This will reveal some useful problem information and it will later motivate our assumptions on the inexactness for problem (1.1). Section 3 contains a brief review of composite-step trust-region SQP algorithms and of their global convergence analyses given in [10]. Our inexact trust-region SQP algorithms and their global convergence analyses will be described in section 4. Assumptions on the inexactness in section 4 are stated in a general way. In section 5 we will discuss how they could be satisfied in an implementation. In the conclusions, section 6, we point to some possible extensions.

We use the following notation. We often set $x = (y, u)$ and use z_y and z_u to represent the subvectors of $z \in \mathbb{R}^n$ corresponding to the y and u components, respectively. The SQP iterates are indexed by k and the symbol of a function with subscript k is used to represent the value of that function at x_k and, possibly, λ_k. For instance, $f_k = f(x_k) = f(y_k, u_k)$. The vector and matrix norms used are the ℓ_2 norms, i.e., $\|\cdot\| = \|\cdot\|_2$. The $l \times l$ identity matrix is denoted by I_l.

2. Trust-region methods for the black-box formulation with inexactness. Under the assumptions of the implicit function theorem, problem (1.1) can be locally reduced to an unconstrained problem in the variable u. Since the type of inaccuracies we are interested in for (1.1) relate to function and gradient inaccuracies for the reduced problem, it is worthwhile to review existing results on trust-region methods with inexact function and gradient evaluations for unconstrained problems. To simplify this presentation, we impose conditions that ensure that (1.1) is equivalent to an unconstrained problem. We suppose that for all $u \in \mathbb{R}^{n-m}$ the constraint equation $C(y, u) = 0$ has a unique solution y, and that $C_y(y, u)$ is invertible for all (y, u) with $C(y, u) = 0$. In this case the implicit function theorem guarantees the existence of a twice continuously differentiable function $u \mapsto y(u)$ defined through the solution of $C(y, u) = 0$. Instead of (1.1) we can consider the equivalent reduced problem

$$\min \widehat{f}(u) = f(y(u), u). \tag{2.1}$$

This problem is also called the black-box formulation of the optimization problem (1.1) because the solution of $C(y, u) = 0$ is treated as a black-box in the optimization algorithms for (2.1). It can be shown that

$$\nabla \widehat{f}(u) = W(y, u)^T \nabla f(y, u)|_{y=y(u)} = W(y, u)^T \nabla \ell(y, u, \lambda)|_{y=y(u),\lambda=\lambda(u)}, \tag{2.2}$$

where

$$W(y, u) = \begin{pmatrix} -C_y(y, u)^{-1} C_u(y, u) \\ I_{n-m} \end{pmatrix}, \tag{2.3}$$

and $\lambda(u)$ solves $C_y(y(u), u)^T \lambda = -\nabla_y f(y(u), u)$. For details see, e.g., [11, 19].

Now suppose that the nonlinear equations $C(y, u_k) = 0$ cannot be solved exactly for $y_k = y(u_k)$ but that an approximation $\widetilde{y}(u_k)$ of $y_k = y(u_k)$ is computed by applying an iterative method to $C(y, u_k) = 0$. In this case the function $\widehat{f}$ and its gradient can not be evaluated exactly. Gradient computation also requires the solution of a linear system of the form $C_y(y_k, u_k)^T z = -\nabla_y f(y_k, u_k)$; if such systems are solved iteratively, this will introduce another source of inexactness in the gradient. How does one need to control the inexactness in function values and gradients in trust-region methods for (2.1)? The influence of inexact gradient information is analyzed in [3], [5, section 8.4], [35] (for a detailed literature review see [5, p. 296]), and the influence of inexact function evaluations is studied in [5, section 10.6]. We want to ensure that our inexactness assumptions for the trust-region method for (1.1) are compatible with the existing inexactness assumptions for trust-region methods for (2.1) in the case that the SQP iterate (y_k, u_k) satisfies $C(y_k, u_k) = 0$. Therefore we briefly review the theory in [5, sections 8.4, 10.6].

In a trust-region method for the solution of (2.1), one computes an approximate solution of

$$\min_{\|s_u\| \le \Delta_k} \widehat{m}_k(s_u) \stackrel{\text{def}}{=} \widehat{f}_k + \widehat{g}_k^T s_u + \tfrac{1}{2} s_u^T \widehat{H}_k s_u,$$

where $\widehat{g}_k$ is an approximation of $\nabla \widehat{f}(u_k)$, and $\widehat{H}_k$ replaces $\nabla^2 \widehat{f}(u_k)$. The decision about the acceptance of $u_k + (s_u)_k$ as the next iterate and about how to update the trust-region radius is based on the ratio of actual decrease $\widehat{\text{ared}}_k = \widehat{f}(u_k) - \widehat{f}(u_k + (s_u)_k)$ to predicted decrease $\widehat{\text{pred}}_k = \widehat{m}_k(0) - \widehat{m}_k((s_u)_k)$. Let $\eta_2 \in (0,1)$ be the constant so that the trust-region radius is reduced if and only if $\widehat{\text{ared}}_k/\widehat{\text{pred}}_k < \eta_2$, and let $\eta_1 \in (0, \eta_2]$ be the constant so that the step is rejected if and only if $\widehat{\text{ared}}_k/\widehat{\text{pred}}_k < \eta_1$.

In [5, section 8.4] it is shown that if the relative gradient error satisfies

$$\|\widehat{g}_k - \nabla \widehat{f}(u_k)\|/\|\widehat{g}_k\| \leq \xi < (1-\eta_2)/2, \tag{2.4}$$

then global convergence of the trust-region algorithm to stationary points can be guaranteed. This accuracy requirement for the gradient approximation is rather weak.

Inexact evaluation of $\widehat{f}$ influences the computation of $\widehat{\text{ared}}_k$. The influence of inexact function evaluations is analyzed in [5, section 10.6]. It is sufficient that

$$\begin{array}{rcl} |f(\widetilde{y}(u_k), u_k) - f(y(u_k), u_k)| & \leq & \eta_0 \widehat{\text{pred}}_k, \\ |f(\widetilde{y}(u_k + (s_u)_k), u_k + (s_u)_k) - f(y(u_k + (s_u)_k), u_k + (s_u)_k)| & \leq & \eta_0 \widehat{\text{pred}}_k, \end{array} \tag{2.5}$$

where $\eta_0 < \frac{1}{2}\eta_1$. In particular, these accuracy requirements guarantee that if the ratio of actual decrease to predicted decrease indicates acceptance of the step, i.e., if $\widehat{\text{ared}}_k/\widehat{\text{pred}}_k \geq \eta_1$, where $\widehat{\text{ared}}_k$ is computed with the inexact function values, then one still obtains a sufficient decrease $\widehat{f}(u_k) - \widehat{f}(u_k + (s_u)_k) \geq (\eta_1 - 2\eta_0)\widehat{\text{pred}}_k$ in the exact function values. Note also that the accuracy requirement for $f(\widetilde{y}(u_k), u_k)$ depends on the trust-region step $(s_u)_k$, which is not known when $f(\widetilde{y}(u_k), u_k)$ is computed the first time. Therefore, $f(\widetilde{y}(u_k), u_k)$ might have to be recomputed if $\widehat{\text{pred}}_k$ becomes too small to meet the required accuracy requirement. For more details see [5, section 10.6].

3. Trust-region SQP methods. In this section we describe the class of composite-step trust-region algorithms assuming exact f and C derivative information and assuming exact solutions of linear systems of the form (1.2). Our representation follows [10, 11]. This section is needed to introduce some basic terminology and notation, as well as to describe later on what can go wrong if f or C derivative information or linear system (1.2) solutions are inexact.

3.1. The main components of our composite-step trust-region algorithms. Given a local minimizer $x_* = (y_*, u_*)$ for problem (1.1), there exists a Lagrange multiplier λ_* such that the gradient $\nabla \ell(x_*, \lambda_*)$ of the Lagrangian function

$$\ell(y, u, \lambda) = f(y, u) + \lambda^T C(y, u)$$

is zero. If $C_y(x_*)$ is assumed to be nonsingular, then the Lagrange multiplier λ_* is determined by $\nabla_y \ell(x_*, \lambda_*) = \nabla_y f(x_*) + C_y(x_*)^T \lambda_* = 0$, and the first-order necessary optimality conditions can be written as

$$\begin{array}{rcl} \nabla_u \ell(x_*, \lambda(x_*)) & = & W(x_*)^T \nabla f(x_*) \; = \; 0, \\ \nabla_\lambda \ell(x_*, \lambda(x_*)) & = & C(x_*) \; = \; 0, \end{array} \tag{3.1}$$

where $W(x_*)$ is given by (2.3).

Given approximations $x_k = (y_k, u_k)$ and λ_k for the solution (y_*, u_*) and the corresponding Lagrange multiplier λ_* of (1.1), SQP algorithms compute an (approximate) solution of the quadratic programming (QP) problem

$$\begin{aligned} \min \quad & q_k(s) \stackrel{\text{def}}{=} \ell(x_k, \lambda_k) + \nabla_x \ell(x_k, \lambda_k)^T s + \tfrac{1}{2} s^T H_k s, \\ \text{s.t.} \quad & C_y(x_k) s_y + C_u(x_k) s_u + C(x_k) = 0, \end{aligned} \tag{3.2}$$

where H_k is a symmetric approximation to the Hessian $\nabla^2_{xx} \ell(x_k, \lambda_k)$ of the Lagrangian at (y_k, u_k, λ_k) or the Hessian itself, and then generate a new iterate (y_{k+1}, u_{k+1}) from this QP solution and, possibly, the corresponding Lagrange multiplier λ_{k+1}. To ensure global convergence, a trust-region condition of the form $\|s\| \le \Delta_k$ is imposed. However, the linear constraints in (3.2) and this trust-region constraint can be incompatible. To deal with the possibility of incompatible constraints, composite-step trust-region algorithms, many of which are reviewed in [5, section 15.4], [10, section 4], split the step s into a sum of two steps s^{n} and s^{t}. We assume that $C_y(x_k)$ is invertible. In this case the step decomposition takes the form

$$s = \begin{pmatrix} s_y \\ s_u \end{pmatrix} = s^{\mathsf{n}} + s^{\mathsf{t}} = \begin{pmatrix} s_y^{\mathsf{n}} \\ 0 \end{pmatrix} + \begin{pmatrix} s_y^{\mathsf{t}} \\ s_u \end{pmatrix}.$$

3.1.1. The quasi-normal step toward feasibility. First, composite-step trust-region algorithms compute a so-called quasi-normal step s_k^{n}, which is responsible for moving towards feasibility. Since we assume that $C_y(x_k)$ is invertible, the y-component of s_k^{n} is an approximate solution of

$$\begin{aligned} \min \quad & \|C_y(x_k) s_y^{\mathsf{n}} + C(x_k)\| \\ \text{s.t.} \quad & \|s_y^{\mathsf{n}}\| \le \Delta_k, \end{aligned} \tag{3.3}$$

and the u-component of s_k^{n} is given by $(s_u^{\mathsf{n}})_k = 0$. Subproblem (3.3) is not solved exactly. A rather coarse solution is sufficient to guarantee basic global convergence. The quasi-normal component s_k^{n} is required to satisfy

$$\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|^2 \ge \kappa_1 \|C_k\| \min\{\kappa_2 \|C_k\|, \Delta_k\}, \tag{3.4}$$

where κ_1 and κ_2 are positive constants independent of k.

3.1.2. The tangential step toward optimality. In a second step, composite-step trust-region algorithms compute a so-called tangential step s_k^{t}, which is responsible for moving towards optimality but has to maintain linearized feasibility, i.e., has to be in the null-space of the linearized constraints. The tangential step is an approximate solution of

$$\begin{aligned} \min \quad & q_k(s_k^{\mathsf{n}} + s^{\mathsf{t}}) \\ \text{s.t.} \quad & C_y(x_k) s_y^{\mathsf{t}} + C_u(x_k) s_u = 0, \\ & \|s_u\| \le \Delta_k. \end{aligned} \tag{3.5}$$

From the constraints in (3.5) we see that $s^{\mathsf{t}} = W_k s_u$, where W_k is defined in (2.3). Therefore we can write

$$q_k(s_k^{\mathsf{n}} + s^{\mathsf{t}}) = q_k(s_k^{\mathsf{n}}) + \left(W_k^T (H_k s_k^{\mathsf{n}} + \nabla_x \ell_k)\right)^T s_u + \tfrac{1}{2} s_u^T W_k^T H_k W_k s_u \tag{3.6}$$

and pose problem (3.5) entirely in s_u:

$$\begin{array}{ll} \min & \widehat{q}_k(s_u) \stackrel{\text{def}}{=} q_k(s_k^{\mathsf{n}}) + \left(W_k^T(H_k s_k^{\mathsf{n}} + \nabla_x \ell_k)\right)^T s_u + \frac{1}{2}(s_u)^T W_k^T H_k W_k(s_u), \\ \text{s.t.} & \|s_u\| \leq \Delta_k. \end{array} \tag{3.7}$$

Reduced SQP algorithms do not approximate the Hessian $\nabla^2_{xx}\ell(x_k,\lambda_k)$ but the reduced Hessian $W_k^T\nabla^2_{xx}\ell(x_k,\lambda_k)W_k$. In this case $W_k^T H_k W_k$ in (3.7) is replaced by the reduced Hessian approximation $\widehat{H}_k$, and the term $H_k s_k^{\mathsf{n}}$ is approximated. The details of the latter approximation are not important in our global analysis and we refer to, e.g., [1] for more details.

The tangential step does not need to solve (3.5) or (3.7) exactly. It is sufficient that the tangential component $(s_u)_k$ satisfies a fraction of the Cauchy decrease condition associated with the trust-region subproblem (3.7). In other words, $(s_u)_k$ has to provide as much decrease in the quadratic $\widehat{q}_k(s_u)$ as the decrease achieved in the direction $-\nabla\widehat{q}_k(0) = -W_k^T(H_k s_k^{\mathsf{n}} + \nabla_x\ell_k)$ inside the trust-region. It can be proved that such a condition implies

$$\widehat{q}_k(0) - \widehat{q}_k((s_u)_k) \geq \kappa_4 \|W_k^T(H_k s_k^{\mathsf{n}} + \nabla_x\ell_k)\| \min\left\{\kappa_5\|W_k^T(H_k s_k^{\mathsf{n}} + \nabla_x\ell_k)\|,\ \kappa_6\Delta_k\right\}, \tag{3.8}$$

where κ_4, κ_5, and κ_6 are positive constants independent of k.

3.1.3. Measuring progress and evaluating the trial step. To decide about acceptance of the step $s_k = s_k^{\mathsf{n}} + s_k^{\mathsf{t}}$, we follow [10] and use the augmented Lagrangian merit function

$$L(x,\lambda;\rho) = f(x) + \lambda^T C(x) + \rho C(x)^T C(x) = \ell(x,\lambda) + \rho C(x)^T C(x).$$

The decision about acceptance of the step and update of the trust-region radius Δ_k is based on the ratio of actual decrease $\text{ared}(s_k;\rho_k)$, given by

$$\text{ared}(s_k;\rho_k) \stackrel{\text{def}}{=} L(x_k,\lambda_k;\rho_k) - L(x_k+s_k,\lambda_{k+1};\rho_k), \tag{3.9}$$

and predicted decrease $\text{pred}(s_k;\rho_k)$, given by

$$\text{pred}(s_k;\rho_k) \stackrel{\text{def}}{=} L(x_k,\lambda_k;\rho_k) - \left(q_k(s_k) + \Delta\lambda_k^T(J_k s_k + C_k) + \rho_k\|J_k s_k + C_k\|^2\right), \tag{3.10}$$

where q_k is defined in (3.2), where $J(y,u) = (C_y(y,u) \mid C_u(y,u))$ is the Jacobian of C, and where $\Delta\lambda_k = \lambda_{k+1} - \lambda_k$. Since the tangential step lies in the null-space of J_k, we have $J_k s_k^{\mathsf{t}} = C_y(x_k)(s_y^{\mathsf{t}})_k + C_u(x_k)(s_u)_k = 0$, and it can be easily seen that

$$\begin{array}{rcl} \text{pred}(s_k;\rho_k) & = & \widehat{q}_k(0) - \widehat{q}_k((s_u)_k) \\ & & + q_k(0) - q_k(s_k^{\mathsf{n}}) - (\Delta\lambda_k)^T(C_y(x_k)(s_y^{\mathsf{n}})_k + C_k) \\ & & + \rho_k\left(\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|^2\right). \end{array} \tag{3.11}$$

Recall that $\widehat{q}_k((s_u)_k) = q_k(s_k^{\mathsf{n}} + W_k(s_u)_k)$ (see (3.7)).

Because of the requirements (3.4), (3.8) on the quasi-normal step and tangential step, respectively, we have that $\widehat{q}_k(0) - \widehat{q}_k((s_u)_k) + \rho_k\left(\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|^2\right) > 0$, provided x_k does not satisfy the first-order necessary optimality conditions (3.1). To ensure that $\text{pred}(s_k;\rho_k)$ is sufficiently positive, the penalty parameter ρ_k is increased if necessary. In fact, the penalty parameter ρ_k will be chosen so that

$$\text{pred}(s_k;\rho_k) \geq \frac{\rho_k}{2}\left(\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|^2\right)$$

(see step 2.6 in Algorithm 3.1 below).

3.2. Statement of the algorithm. This leads to the following class of trust-region SQP algorithms. They are the same as the trust-region SQP algorithms in [10], but are adapted to our problem context and to our notation.

ALGORITHM 3.1 (Trust-region SQP algorithm).

1. Choose x_0 and $\Delta_0 > 0$, and calculate λ_0. Set $\rho_{-1} \geq 1$ and $\epsilon_{tol} > 0$. Choose α_1, η_1, Δ_{min}, Δ_{max}, and $\bar{\rho}$ such that $0 < \alpha_1, \eta_1 < 1,\ 0 < \Delta_{min} \leq \Delta_{max}$, and $\bar{\rho} > 0$.
2. For $k = 0, 1, 2, \ldots$
   - 2.1 Compute s_k^{n} satisfying (3.13) and (3.4).
   - 2.2 Compute $W_k^T \nabla q_k(s_k^{\mathsf{n}})$.
   - 2.3 If $\|C_k\| + \|W_k^T \nabla q_k(s_k^{\mathsf{n}})\| \leq \epsilon_{tol}$, stop and return x_k as an approximate solution for problem (1.1).
   - 2.4 Compute $(s_u)_k$ satisfying (3.8).
   - 2.5 Compute λ_{k+1} and set $\Delta\lambda_k = \lambda_{k+1} - \lambda_k$.
   - 2.6 Update the penalty parameter.
     If $\text{pred}(s_k; \rho_{k-1}) \geq \frac{\rho_{k-1}}{2}\Big(\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|^2\Big)$, then set $\rho_k = \rho_{k-1}$.
     Otherwise set
     $$\rho_k = \frac{2\left(-\widehat{q}_k(0) + \widehat{q}_k((s_u)_k) - q_k(0) + q_k(s_k^{\mathsf{n}}) + \Delta\lambda_k^T(C_y(x_k)(s_y^{\mathsf{n}})_k + C_k)\right)}{\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|^2} + \bar{\rho}.$$
   - 2.7 Compute $(s_y^{\mathsf{t}})_k = -C_y(x_k)^{-1}C_u(x_k)(s_u)_k$ (if not already done in step 2.4).
   - 2.8 If $\text{ared}(s_k; \rho_k)/\text{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k) < \eta_1$, set
     $$\Delta_{k+1} = \alpha_1 \max\{\|s_k^{\mathsf{n}}\|, \|(s_u)_k\|\}$$
     and reject s_k.
     Otherwise accept s_k and choose Δ_{k+1} such that
     $$\max\{\Delta_{min}, \Delta_k\} \leq \Delta_{k+1} \leq \Delta_{max}.$$
   - 2.9 If s_k was rejected, set $x_{k+1} = x_k$ and $\lambda_{k+1} = \lambda_k$. Otherwise set $x_{k+1} = x_k + s_k$ and let λ_{k+1} be the vector computed in step 2.5.

Remark 3.2. In reduced SQP methods, one uses

$$H_k = \begin{pmatrix} 0 & 0 \\ 0 & \widehat{H}_k \end{pmatrix}.$$

In this case, $H_k s_k^{\mathsf{n}} = 0$ and steps 2.1 and 2.7 can be merged into a step inserted immediately after step 2.4. Instead of executing steps 2.1 and 2.7, one computes immediately after step 2.4 an approximate solution $(s_y)_k$ of

$$\begin{aligned} \min \quad & \|C_y(x_k)s_y + C(x_k)\| \\ \text{s.t.} \quad & \|s_y\| \leq \Delta_k, \end{aligned} \tag{3.12}$$

which satisfies (3.13) and (3.4). In this case, $(s_y^{\mathsf{n}})_k$ in steps 2.6 and 2.8 is replaced by $(s_y)_k$.

3.3. First-order global convergence of the algorithm. Dennis, El-Alem, and Maciel [10] have proved that the class of trust-region SQP algorithms 3.1 is globally convergent. Their convergence theory requires the set of assumptions given below. For all iterations k we assume that $x_k,\ x_k + s_k \in \Omega$, where Ω is an open subset of $\mathbb{R}^n$.

A.1. The functions f, c_i, $i = 1, \ldots, m$, are twice continuously differentiable functions in Ω. Here $c_i(x)$ represents the ith component of $C(x)$.

A.2. The partial Jacobian $C_y(x)$ is nonsingular for all $x \in \Omega$.

A.3. The functions f, ∇f, $\nabla^2 f$, C, J, $\nabla^2 c_i$, $i = 1, \ldots, m$, are bounded in Ω. The matrix $C_y(x)^{-1}$ is uniformly bounded in Ω.

A.4. The sequences $\{H_k\}$, $\{W_k\}$, and $\{\lambda_k\}$ are bounded.

Dennis, El-Alem, and Maciel [10] show that for a subsequence of the iterates the first-order necessary optimality conditions (3.1) of problem (1.1) are satisfied in the limit.

THEOREM 3.3. *Let assumptions* A.1–A.4 *hold. The sequences of iterates generated by the trust-region SQP algorithms* 3.1 *satisfy*

$$\liminf_{k\to\infty} \left(\|W_k^T \nabla f_k\| + \|C_k\| \right) = 0.$$

We remark that inequality (3.4) and A.3 imply the existence of $\kappa_3 > 0$, independent of k, such that

$$\|s_k^{\mathsf{n}}\| \leq \kappa_3 \|C_k\|. \tag{3.13}$$

In fact, using $\|C_y(x_k)(s_k^{\mathsf{n}})_y + C_k\| \leq \|C_k\|$ and the boundedness of $\{C_y(x_k)^{-1}\}$ we find that

$$\|s_k^{\mathsf{n}}\| \leq \|C_y(x_k)^{-1}\| \left(\|C_y(x_k)(s_k^{\mathsf{n}})_y + C_k\| + \|C_k\| \right) \leq 2\|C_y(x_k)^{-1}\|\, \|C_k\|\,.$$

In [10] the condition (3.13) is imposed as an additional condition on the quasi-normal step, because more general quasi-normal steps are allowed.

4. Trust-region SQP methods with inexactness. Now we allow f and C derivative information, as well as linear system (1.2) solutions to be inexact. We assume, however, that the user is able to adjust the level of inexactness. We will investigate how Algorithm 3.1 has to be modified to cope with this inexactness. Our aim is to devise conditions on the allowable level of inexactness that meet three criteria. First, we want our conditions to be as weak as possible to admit inexpensive problem information when the iterates (y_k, u_k) are far away from the solution. Second, we want our conditions to be comparable with the conditions on inexact function and gradient information for unconstrained trust-region methods applied to the black-box formulation (2.1), which have been reviewed in section 2. Third, while our conditions on the allowable level of inexactness will be general, we want them to be implementable. In particular, the conditions on the allowable level of inexactness should not depend on derivative bounds, Lipschitz constants, and other quantities that cannot be computed in practice.

4.1. The main components of our composite-step trust-region algorithms with inexact problem information.

4.1.1. The quasi-normal step. The assumption (3.4) on the quasi-normal step turns out to be rather weak and can be satisfied using several algorithms that fit into

our inexactness framework. This issue will be discussed in section 5.1. Notice also that assumption (3.4) is already expressed in terms of the right-hand side C_k and the residual $C_y(x_k)s_y^{\mathsf{n}} + C_k$ of the linear system $C_y(x_k)s_y^{\mathsf{n}} = -C_k$.

4.1.2. The *u*-component of the tangential step. The computation of the tangential step s_k^{t} allowing inexact information is more complicated. Among other things, we cannot assume that s_k^{t} is in the null-space of the linearized constraints. This condition, expressed as $s^{\mathsf{t}} = W_k s_u$, was used repeatedly in sections 3.1.2 and 3.1.3. It will be very useful to discuss the computation of the u-component and the computation of the y-component of the tangential step separately.

If exact derivative information and exact linearized system solves are available, then the u-component of the tangential step is the approximate solution of (3.7). Now only approximations of $W_k^T(H_k s_k^{\mathsf{n}} + \nabla_x \ell_k)$ and $W_k^T H_k W_k$ will be available, and we compute s_u as the approximate solution of

$$\begin{aligned} \min \quad & \widehat{m}_k(s_u) \stackrel{\text{def}}{=} q_k(s_k^{\mathsf{n}}) + \widehat{g}_k^T s_u + \tfrac{1}{2} s_u^T \widetilde{W_k^T H_k W_k} s_u \\ \text{s.t.} \quad & \|s_u\|_2 \le \Delta_k. \end{aligned} \tag{4.1}$$

In (4.1), the symbol $\widetilde{\ }$ over $W_k^T H_k W_k$ indicates that the reduced Hessian approximation may be inexact. What are the accuracy requirements on $\widehat{g}_k$ and on $\widetilde{W_k^T H_k W_k}$?

If (y_k, u_k) were feasible, i.e., if $C(y_k, u_k) = 0$, then $s_k^{\mathsf{n}} = 0$ (see (3.4)) and $\nabla \widehat{f}(u_k) = W_k^T(H_k s_k^{\mathsf{n}} + \nabla_x \ell_k)$ (see (2.2)). In this case the theory of [5, section 8.4] for the reduced problem (2.1), which was reviewed in section 2, suggests an accuracy requirement of the form

$$\|\widehat{g}_k - W_k^T(H_k s_k^{\mathsf{n}} + \nabla_x \ell_k)\| \le \xi_1 \|\widehat{g}_k\|, \tag{4.2}$$

with some $\xi_1 \in (0, 1)$ which is related to the parameters in the trust-region algorithm (cf. (2.4)). We need a slightly stronger condition, namely,

$$\|\widehat{g}_k - W_k^T(H_k s_k^{\mathsf{n}} + \nabla_x \ell_k)\| \le \xi_1 \min\left\{\|\widehat{g}_k\|, \Delta_k\right\}, \tag{4.3}$$

where $\xi_1 > 0$. In (4.3) the constant ξ_1 is not tied to the parameters in the trust-region algorithm—in particular, we do not need $\xi_1 < 1$—but the absolute error in the reduced gradient approximation must be less than $\|\widehat{g}_k\|$ and Δ_k.

In section 5.2 we show how (4.3) can be enforced in practice, if errors in the reduced gradient are due to inexact linear system solves. There we will see that while (4.3) is slightly stronger than (4.2), the fact that we can give up the restriction $\xi_1 < 1$ makes (4.3) preferable from an implementation point of view.

Remark 4.1. Imposing the inexactness condition

$$\|\widehat{g}_k - \nabla \widehat{f}(u_k)\| \le \xi_1 \min\left\{\|\widehat{g}_k\|, \Delta_k\right\}, \tag{4.4}$$

where $\xi_1 > 0$, instead of (2.4) also gives the standard lim inf global convergence result for the unconstrained problem (2.1). This may be seen using the proof in [27, Theorem 4.10] and applying (4.4) in the estimate for $|\psi_k(s_k) - \nabla f(x_k)^T s_k|$ from [27, p. 278].

The approximate reduced Hessian has to satisfy

$$(s_u)_k^T \widetilde{W_k^T H_k W_k} (s_u)_k \le \xi_2 \|(s_u)_k\|^2 \tag{4.5}$$

for some $\xi_2 > 0$ independent of k. If $W_k^T H_k W_k$ is evaluated exactly, then (4.5) is implied by assumption A.4.

The approximate solution $(s_u)_k$ of (4.1) computed in step 2.4 of Algorithm 3.1 must provide a fraction of the Cauchy decrease on this approximate model $\widehat{m}_k$, i.e.,

$$\widehat{m}_k(0) - \widehat{m}_k((s_u)_k) \;\geq\; \kappa_4 \|\widehat{g}_k\| \min\left\{\kappa_5 \|\widehat{g}_k\|,\; \kappa_6 \Delta_k\right\}, \tag{4.6}$$

where, as in (3.8), κ_4, κ_5, and κ_6 are positive constants independent of k. One method to actually compute s_u satisfying (4.6) will be discussed in section 5.3.

4.1.3. Measuring progress, updating the penalty parameter, and evaluating the trial step. The reformulation (3.11) of the predicted decrease $\mathrm{pred}(s_k;\rho_k)$ defined in (3.10) is only valid if s_k^{t} is in the null-space of the linearized constraints. If this is not the case, then

$$\begin{aligned}
\mathrm{pred}(s_k;\rho_k) \;=\;& \widehat{q}_k(0) - \widehat{q}_k((s_u)_k) \\
&+ q_k(0) - q_k(s_k^{\mathsf{n}}) - (\Delta\lambda_k)^T (C_y(x_k)(s_y^{\mathsf{n}})_k + C_k) \\
&+ \rho_k \left(\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|^2 \right) \\
&- (\Delta\lambda_k)^T (r_k^{\mathsf{t}}) - \rho_k \|r_k^{\mathsf{t}}\|^2 - 2\rho_k (r_k^{\mathsf{t}})^T \left(C_y(x_k)(s_y^{\mathsf{n}})_k + C_k \right),
\end{aligned}$$

where

$$r_k^{\mathsf{t}} = C_y(x_k)(s_y^{\mathsf{t}})_k + C_u(x_k)(s_u)_k. \tag{4.7}$$

Moreover, the reduced quadratic model $\widehat{q}_k$ defined in (3.2) is now replaced by $\widehat{m}_k$ defined in (4.1). We define

$$\begin{aligned}
\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k) = \widehat{m}_k(0) - \widehat{m}_k((s_u)_k) +\; & q_k(0) - q_k(s_k^{\mathsf{n}}) \\
&- (\Delta\lambda_k)^T (C_y(x_k)(s_y^{\mathsf{n}})_k + C_k) \\
&+\; \rho_k \left(\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|^2 \right)
\end{aligned} \tag{4.8}$$

and

$$\mathrm{rpred}(r_k^{\mathsf{t}};\rho_k) = -(\Delta\lambda_k)^T (r_k^{\mathsf{t}}) - \rho_k \|r_k^{\mathsf{t}}\|^2 - 2\rho_k (r_k^{\mathsf{t}})^T \left(C_y(x_k)(s_y^{\mathsf{n}})_k + C_k \right). \tag{4.9}$$

We now view

$$\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k) + \mathrm{rpred}(r_k^{\mathsf{t}};\rho_k)$$

as the quadratic model of the Lagrangian.

This predicted reduction $\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k)$ depends only on s_k^{n} and $(s_u)_k$ and can be readily computed. In fact, the quantities $\widehat{m}_k(0)$, $\widehat{m}_k((s_u)_k)$, and $C_y(x_k)(s_y^{\mathsf{n}})_k + C_k$ are typically already computed during the computation of the u-component of the tangential step and the computation of the quasi-normal step, respectively.

Because of the requirements (3.4) and (4.6) on s_k^{n} and $(s_u)_k$, respectively, we have that $\widehat{m}_k(0) - \widehat{m}_k((s_u)_k) + \rho_k \left(\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|^2 \right) > 0$, provided (y_k, u_k) does not satisfy the first-order necessary optimality conditions (3.1). We update the penalty parameter ρ_k, if necessary, to ensure sufficient positivity of $\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k)$. See step i2.6 in Algorithm 4.3 below.

The evaluation of the step $s_k = s_k^{\mathsf{n}} + s_k^{\mathsf{t}}$ (we will discuss the computation of $(s_y^{\mathsf{t}})_k$ in a moment) will be based on the ratio $\mathrm{ared}(s_k;\rho_k)/\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k)$.

4.1.4. The *y*-component of the tangential step. As we have noted in the previous section, the quadratic model of the Lagrangian is $\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k) + \mathrm{rpred}(r_k^{\mathsf{t}}; \rho_k)$. However, step evaluations are performed based on $\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k)$ only. To ensure that the inexactness in the tangential step $(s_y^{\mathsf{t}})_k$ does not dominate this quadratic model, we require that

$$\left|\mathrm{rpred}(r_k^{\mathsf{t}}; \rho_k)\right| \;\le\; \eta_0 \mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k), \tag{4.10}$$

where $\eta_0 \in (0, 1-\eta_1)$ is a given constant and η_1 is the parameter in step 2.8 of the trust-region algorithm, and that

$$\|r_k^{\mathsf{t}}\| \;\le\; \xi_3 \Delta_k \|(s_u)_k\| \tag{4.11}$$

for some constant $\xi_3 > 0$ independent of k. If we estimate $|\mathrm{rpred}(r_k^{\mathsf{t}}; \rho_k)| \le \rho_k \|r_k^{\mathsf{t}}\|^2 + (\|\Delta\lambda_k\| + 2\rho_k \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|)\|r_k^{\mathsf{t}}\|$ and insert this upper bound into (4.10), we see that inequality (4.10) is implied by

$$\|r_k^{\mathsf{t}}\| \le -\sigma + \sqrt{\sigma^2 + \eta_0 \mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k)/\rho_k}, \tag{4.12}$$

where $\sigma = \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\| + \|\Delta\lambda_k\|/(2\rho_k)$. Inequalities (4.10) and (4.11) are satisfied for the exact solution of $C_y(x_k)(s_y^{\mathsf{t}})_k = -C_u(x_k)(s_u)_k$. The quantity $\|r_k^{\mathsf{t}}\|$ is the residual accuracy of an inexact solution s_y^{t} of $C_y(x_k)s_y^{\mathsf{t}} = -C_u(x_k)(s_u)_k$. Since $s_k^{\mathsf{n}}, (s_u)_k$, and $\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k)$ are known, a step $(s_y^{\mathsf{t}})_k$ with (4.10) and (4.11) can be computed.

Remark 4.2. i. Condition (4.10) is motivated by (2.5). We need to control the accuracy of $\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k) + \mathrm{rpred}(r_k^{\mathsf{t}}; \rho_k)$, whereas (2.5) controls the accuracy of the actual reduction. However, the effects of both conditions on the ratio of actual and predicted reduction are similar.

ii. Notice that $(s_y^{\mathsf{t}})_k = -C_y(x_k)^{-1}C_u(x_k)(s_u)_k + C_y(x_k)^{-1}r_k^{\mathsf{t}}$ and that (4.11) implies

$$\|C_y(x_k)^{-1} r_k^{\mathsf{t}}\| \le \xi_4 \Delta_k \tag{4.13}$$

for some $\xi_4 > 0$. In other words, it implies that the norm of the residual (tangential) step $C_y(x_k)^{-1}r_k^{\mathsf{t}}$ is bounded by a constant times the trust-region radius. If we view $C_y(x_k)^{-1}r_k^{\mathsf{t}}$ as a second (tangential) step, or as a spacer (tangential) step, we then identify (4.13) as a condition that has already been imposed on steps of such types in the context of global convergence of trust-region algorithms for unconstrained optimization [5, section 10.4], [6].

We note that the amount of positivity in $\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k)$ is determined by the reductions $\widehat{m}_k(0) - \widehat{m}_k((s_u)_k)$ and $\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|^2$. Thus we can allow more inaccuracy in the $(s_y^{\mathsf{t}})_k$ computation, which typically translates into less expensive $(s_y^{\mathsf{t}})_k$ computation, the larger the linearized feasibility gain $\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|^2$ achieved by the quasi-normal step *and* the larger the optimality gain $\widehat{m}_k(0) - \widehat{m}_k((s_u)_k)$ achieved by the u-component of the tangential step. In particular, even if $\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|^2$ is small, but $\widehat{m}_k(0) - \widehat{m}_k((s_u)_k)$ is large (which is likely the case at a point $x_k = (y_k, u_k)$ that is almost feasible, but away from being optimal), the accuracy requirement on $(s_y^{\mathsf{t}})_k$ is rather weak. Our criterion also seems to be closely aligned with the SQP philosophy which allows one to trade gains in feasibility for gains in optimality. Another important point worth noting is that inaccuracy in $(s_y^{\mathsf{t}})_k$ does not enter the penalty parameter update. If it would, the

penalty parameter might increase faster. Since too-large penalty parameters ρ_k can slow down the convergence of SQP methods, this is another benefit of our accuracy requirement.

Our initial and somewhat straightforward approach [20, 36] to dealing with inaccuracy did not use the split $\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k) + \mathrm{rpred}(r_k^{\mathsf{t}}; \rho_k)$. Rather, the predicted decrease was defined by (3.10). After determination of s_k^{n} satisfying (3.4) we computed a tangential step that, among other conditions, satisfied

$$\|C_k\|^2 - \|J_k(s_k^{\mathsf{n}} + s_k^{\mathsf{t}}) + C_k\|^2 \leq \xi_5 \left(\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|^2\right) \tag{4.14}$$

with $\xi_5 \in (0,1)$. Thus accuracy of $(s_y^{\mathsf{t}})_k$ depended only on the linearized feasibility gain $\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|^2$ achieved by the quasi-normal step. Moreover, when

$$\mathrm{pred}(s_k; \rho_{k-1}) < \frac{\rho_{k-1}}{2}\Big(\|C_k\|^2 - \|J_k(s_k^{\mathsf{n}} + s_k^{\mathsf{t}}) + C_k\|^2\Big),$$

where $\mathrm{pred}(s_k; \rho_k)$ is given by (3.10), we used the update

$$\rho_k = \frac{2\left(-q_k(0) + q_k(s_k) + \Delta\lambda_k^T(J_k s_k + C_k)\right)}{\|C_k\|^2 - \|J_k(s_k^{\mathsf{n}} + s_k^{\mathsf{t}}) + C_k\|^2} + \bar{\rho}. \tag{4.15}$$

Condition (4.14) often leads to very stringent accuracy requirements for $(s_y^{\mathsf{t}})_k$, and the update (4.15) often leads to large increases in the penalty parameter, especially when the current iterate (y_k, u_k) happens to be almost feasible. The approach presented in this paper represents the quadratic model of the Lagrangian as $\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k) + \mathrm{rpred}(r_k^{\mathsf{t}}; \rho_k)$, separates the computation of the u- and the y-component of the tangential step, bases the accuracy requirement on $(s_y^{\mathsf{t}})_k$ on feasibility *and* optimality gains, and bases the penalty parameter update on quantities that are not contaminated by inaccuracies in $(s_y^{\mathsf{t}})_k$.

4.1.5. Computation of the Lagrange multiplier estimate. Finally, the computation of λ_{k+1} in step 2.5 of the exact trust-region SQP algorithms 3.1 is likely to involve inexact calculations. However, as we have seen in Theorem 3.3, global convergence to a stationary point requires only boundedness from the sequence of Lagrange multipliers $\{\lambda_k\}$. This requirement is not only fairly mild from a theoretical point of view, but, under assumptions A.1–A.4, also easy to impose computationally even when inexactness is present.

4.2. Statement of the algorithm. The inexact trust-region SQP algorithms are defined similarly to their exact counterpart, Algorithm 3.1, but with steps 2.1 to 2.8 modified to accommodate the inexact calculations discussed above.

ALGORITHM 4.3 (Inexact trust-region SQP algorithms).

1. The same initializations as in step 1 of Algorithm 3.1.
2. For $k = 0, 1, 2, \ldots$
   - i2.1 Compute s_k^{n} satisfying (3.13) and (3.4).
   - i2.2 Compute an approximation $\widehat{g}_k$ to $W_k^T \nabla q_k(s_k^{\mathsf{n}})$ satisfying (4.3).
   - i2.3 If $\|C_k\| + \|\widehat{g}_k\| \leq \epsilon_{tol}$, stop and return $x_k = (y_k, u_k)$ as an approximate solution for problem (1.1).
   - i2.4 Compute $(s_u)_k$ satisfying (4.6).
   - i2.5 Compute λ_{k+1} and set $\Delta\lambda_k = \lambda_{k+1} - \lambda_k$.
   - i2.6 Update the penalty parameter.
     If $\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_{k-1}) \geq \frac{\rho_{k-1}}{2}\Big(\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|^2\Big)$, then set

$\rho_k = \rho_{k-1}$.
Otherwise set

$$\rho_k = \frac{2\left(-\widehat{m}_k(0) + \widehat{m}_k((s_u)_k) - q_k(0) + q_k(s_k^{\mathsf{n}}) + \Delta\lambda_k^T(C_y(x_k)(s_y^{\mathsf{n}})_k + C_k)\right)}{\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\|^2} + \bar{\rho}.$$

i2.7 Compute $(s_y^{\mathsf{t}})_k$ so that the residual vector r_k^{t} satisfies (4.10) and (4.11).
i2.8 Compute $\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k)$ using (4.8).
If $\mathrm{ared}(s_k; \rho_k)/\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k) < \eta_1$, set

$$\Delta_{k+1} = \alpha_1 \max\{\|s_k^{\mathsf{n}}\|, \|(s_u)_k\|\}$$

and reject s_k.
Otherwise accept s_k and choose Δ_{k+1} such that

$$\max\{\Delta_{min}, \Delta_k\} \le \Delta_{k+1} \le \Delta_{max}.$$

i2.9 The same step and multiplier updates as in step 2.9 of Algorithm 3.1.

Remark 4.4. In reduced SQP methods where $H_k s_k^{\mathsf{n}} = 0$, the algorithm can be slightly reorganized to save one linear system solve with system matrix $(C_y)_k$. See also Remark 3.2. Steps 2.1 and 2.7 can be merged into a step inserted immediately after step 2.4. Instead of executing steps 2.1 and 2.7, one computes immediately after step 2.4 an approximate solution $(s_y)_k$ of (3.12) which satisfies (3.13) and (3.4). In this case $(s_y^{\mathsf{n}})_k$ is replaced by $(s_y)_k$ in the remaining steps of the algorithm, and $(s_y^{\mathsf{t}})_k = 0$.

4.3. First-order global convergence of the algorithm. The global convergence property of the inexact trust-region SQP algorithms 3.1 is stated in the following theorem.

THEOREM 4.5. *Let assumptions* A.1–A.4 *hold. The sequences of iterates generated by the inexact trust-region SQP algorithms* 4.3 *satisfy*

$$\liminf_{k\to\infty} (\|\widehat{g}_k\| + \|C_k\|) = 0. \tag{4.16}$$

Furthermore, we have

$$\liminf_{k\to\infty} \left(\|W_k^T \nabla f_k\| + \|C_k\|\right) = 0. \tag{4.17}$$

Proof. The proof of (4.16) follows the convergence analysis given in [10] with the predicted decrease used there always replaced by $\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k)$ as defined in (4.8). Only a very few steps in the convergence analysis change and we will review them in detail.

The first modification concerns the relationship between the size of the step s_k and the trust-region radius Δ_k. The convergence analysis requires that

$$\|s_k\| \le \kappa_7 \Delta_k$$

and, if s_k is rejected, that

$$\Delta_{k+1} \ge \kappa_8 \|s_k\|.$$

In our inexact trust-region SQP algorithms, the first inequality follows from the trust-region constraints in (3.3), (4.1), and from (4.11) and assumption A.3. The second inequality is a consequence of the update of the trust-region radius in i2.8.

The second modification is in the estimates of the difference between actual decrease and predicted decrease. Since $\mathrm{rpred}(r_k^{\mathsf{t}};\rho_k)$ is different from zero, the upper bounds on the difference between actual and predicted decreases given in [10, Lemmas 7.4, 7.5] are now different. We will be able to show

$$\begin{aligned}&|\mathrm{ared}(s_k;\rho_k)-\mathrm{pred}(s_k^{\mathsf{n}},(s_u)_k;\rho_k)-\mathrm{rpred}(r_k^{\mathsf{t}};\rho_k)|\\&\quad\le \kappa_9\Delta_k\|s_k\|+\kappa_{10}\rho_k\|s_k\|^3+\kappa_{11}\rho_k\|s_k\|^2\|C_k\|\end{aligned}\tag{4.18}$$

instead of [10, Lemma 7.4] and

$$\left|\mathrm{ared}(s_k;\rho_k)-\mathrm{pred}(s_k^{\mathsf{n}},(s_u)_k;\rho_k)-\mathrm{rpred}(r_k^{\mathsf{t}};\rho_k)\right|\le\kappa_{12}\rho_k\Delta_k\|s_k\|\tag{4.19}$$

instead of [10, Lemma 7.5].

The estimates (4.18) and (4.19) can be verified as follows. Using definitions (4.8) and (4.9), we can see that

$$\begin{aligned}&\mathrm{pred}(s_k^{\mathsf{n}},(s_u)_k;\rho_k)+\mathrm{rpred}(r_k^{\mathsf{t}};\rho_k)\\&=-\widehat{g}_k^T(s_u)_k-\frac{1}{2}(s_u)_k^T\widetilde{W_k^TH_kW_k}(s_u)_k-\nabla_x\ell_k^Ts_k^{\mathsf{n}}-\frac{1}{2}s_k^{\mathsf{n}\,T}H_ks_k^{\mathsf{n}}\\&\quad-\Delta\lambda_k^T\left(J_ks_k+C_k\right)+\rho_k\left(\|C_k\|^2-\|J_ks_k+C_k\|^2\right).\end{aligned}$$

With definition (3.9) of the actual decrease, the previous identity, and $W_k^T(H_ks_k^{\mathsf{n}}+\nabla_x\ell_k)=W_k^T\nabla q_k(s_k^{\mathsf{n}})$, we obtain

$$\begin{aligned}&\mathrm{ared}(s_k;\rho_k)-\left(\mathrm{pred}(s_k^{\mathsf{n}},(s_u)_k;\rho_k)+\mathrm{rpred}(r_k^{\mathsf{t}};\rho_k)\right)\\&\quad=\ell(x_k,\lambda_k)+\rho_k\|C_k\|^2-\ell(x_{k+1},\lambda_{k+1})-\rho_k\|C_{k+1}\|^2\\&\qquad-\mathrm{pred}(s_k^{\mathsf{n}},(s_u)_k;\rho_k)-\mathrm{rpred}(r_k^{\mathsf{t}};\rho_k)\\&\quad=\ell(x_k,\lambda_k)-\ell(x_{k+1},\lambda_k)+\ell(x_{k+1},\lambda_k)-\ell(x_{k+1},\lambda_{k+1})\\&\qquad+(H_ks_k^{\mathsf{n}}+\nabla_x\ell_k)^TW_k(s_u)_k+\tfrac{1}{2}(s_u)_k^TW_k^TH_kW_k(s_u)_k+\nabla_x\ell_k^Ts_k^{\mathsf{n}}+\tfrac{1}{2}s_k^{\mathsf{n}\,T}H_ks_k^{\mathsf{n}}\\&\qquad+(\widehat{g}_k-W_k^T\nabla q_k(s_k^{\mathsf{n}}))^T(s_u)_k+\tfrac{1}{2}(s_u)_k^T\widetilde{W_k^TH_kW_k}(s_u)_k-\tfrac{1}{2}(s_u)_k^TW_k^TH_kW_k(s_u)_k\\&\qquad+\Delta\lambda_k^T\left(J_ks_k+C_k\right)-\rho_k(\|C_{k+1}\|^2-\|J_ks_k+C_k\|^2)\\&\quad=-\ell(x_{k+1},\lambda_k)+q_k(s_k)-q_k(s_k)+\widehat{q}_k((s_u)_k)\\&\qquad+(\widehat{g}_k-W_k^T\nabla q_k(s_k^{\mathsf{n}}))^T(s_u)_k+\tfrac{1}{2}(s_u)_k^T\widetilde{W_k^TH_kW_k}(s_u)_k-\tfrac{1}{2}(s_u)_k^TW_k^TH_kW_k(s_u)_k\\&\qquad+\Delta\lambda_k^T\left(-C_{k+1}+J_ks_k+C_k\right)-\rho_k\left(\|C_{k+1}\|^2-\|J_ks_k+C_k\|^2\right).\end{aligned}\tag{4.20}$$

Using a Taylor expansion and definition (3.2) of q_k gives

$$|-\ell(x_{k+1},\lambda_k)+q_k(s_k)|\le\tfrac{1}{2}\|H_k-\nabla^2_{xx}\ell(x_k+t_k^1s_k,\lambda_k)\|\,\|s_k\|^2\tag{4.21}$$

with some $t_k^1\in(0,1)$. Using the definitions (3.2) and (3.7) of q_k and $\widehat{q}_k$, respectively, along with (3.6) and (4.7), we find that

$$\begin{aligned}&|-q_k(s_k)+\widehat{q}_k((s_u)_k)|\\&\quad\le\|H_ks_k^{\mathsf{n}}-\nabla_x\ell(x_k,\lambda_k)\|\,\|s_k^{\mathsf{t}}-W_k(s_u)_k\|+\tfrac{1}{2}\|H_k\|\,\|s_k^{\mathsf{t}}\|^2+\tfrac{1}{2}\|W_k^TH_kW_k\|\,\|(s_u)_k\|^2\\&\quad\le\|H_ks_k^{\mathsf{n}}-\nabla_x\ell(x_k,\lambda_k)\|\,\|C_y(x_k)^{-1}\|\,\|r_k^{\mathsf{t}}\|+\tfrac{1}{2}\|H_k\|\,\|s_k^{\mathsf{t}}\|^2+\tfrac{1}{2}\|W_k^TH_kW_k\|\,\|(s_u)_k\|^2.\end{aligned}\tag{4.22}$$

With

$$\|s_k^{\mathsf{t}}\| \le \|s_k^{\mathsf{t}} - W_k(s_u)_k\| + \|W_k(s_u)_k\| \le \|C_y(x_k)^{-1}\|\,\|r_k^{\mathsf{t}}\| + \|W_k\|\,\|(s_u)_k\|$$

and (4.11), equation (4.22) implies

$$\begin{aligned}
&|-q_k(s_k) + \widehat{q}_k((s_u)_k)| \\
&\le \xi_3\|H_k s_k^{\mathsf{n}} - \nabla_x \ell(x_k,\lambda_k)\|\,\|C_y(x_k)^{-1}\|\,\Delta_k\|(s_u)_k\| \\
&\quad + \tfrac{1}{2}\|H_k\|\,\left(\xi_3^2\|C_y(x_k)^{-1}\|^2\Delta_k^2 + 2\xi_3\|W_k\|\,\|C_y(x_k)^{-1}\|\Delta_k + \|W_k\|^2\right)\|(s_u)_k\|^2 \\
&\quad + \tfrac{1}{2}\|W_k^T H_k W_k\|\,\|(s_u)_k\|^2. \qquad (4.23)
\end{aligned}$$

The inequalities (4.3) and (4.5) give

$$\begin{aligned}
&(\widehat{g}_k - W_k^T\nabla q_k(s_k^{\mathsf{n}}))^T(s_u)_k + \tfrac{1}{2}(s_u)_k^T\widetilde{W_k^T H_k W_k}(s_u)_k - \tfrac{1}{2}(s_u)_k^T W_k^T H_k W_k(s_u)_k \\
&\le \xi_1\Delta_k\|(s_u)_k\| + \tfrac{1}{2}(\xi_2 + \|W_k^T H_k W_k\|)\|(s_u)_k\|^2. \qquad (4.24)
\end{aligned}$$

Using a Taylor expansion, we obtain

$$\begin{aligned}
&\Delta\lambda_k^T\,(-C_{k+1} + J_k s_k + C_k) - \rho_k\left(\|C_{k+1}\|^2 - \|J_k s_k + C_k\|^2\right) \\
&= -\tfrac{1}{2}\sum_{i=1}^m (\Delta\lambda_k)_i s_k^T\nabla^2 c_i(x_k + t_k^2 s_k)s_k \\
&\quad - \rho_k\left(\sum_{i=1}^m c_i(x_k + t_k^3 s_k)(s_k)^T\nabla^2 c_i(x_k + t_k^3 s_k)(s_k)\right. \\
&\qquad \left. + (s_k)^T J(x_k + t_k^3 s_k)^T J(x_k + t_k^3 s_k)(s_k) - (s_k)^T J(x_k)^T J(x_k)(s_k)\right),
\end{aligned}$$

where $t_k^2, t_k^3 \in (0,1)$. Now we expand $c_i(x_k + t_k^3 s_k)$ around $c_i(x_k)$. This expansion and assumptions A.1–A.4 give

$$\begin{aligned}
&\Delta\lambda_k^T\,(-C_{k+1} + J_k s_k + C_k) - \rho_k\left(\|C_{k+1}\|^2 - \|J_k s_k + C_k\|^2\right) \\
&\qquad \le \kappa_{10}\rho_k\|s_k\|^3 + \kappa_{11}\rho_k\|s_k\|^2\|C_k\|. \qquad (4.25)
\end{aligned}$$

If we insert (4.21)–(4.25) into (4.20) and use assumptions A.3, A.4, and (4.11), we arrive at the desired estimate (4.18) for some positive constants κ_9, κ_{10}, and κ_{11}. Inequality (4.19) is then a direct consequence of inequality (4.18) and the fact that $\rho_k \ge 1$.

We can now bound the difference between the actual and predicted decreases in the inexact context. Combining (4.18) with (4.10) yields

$$\begin{aligned}
&|\mathrm{ared}(s_k;\rho_k) - \mathrm{pred}(s_k^{\mathsf{n}},(s_u)_k;\rho_k)| \\
&\le \left|\mathrm{ared}(s_k;\rho_k) - \mathrm{pred}(s_k^{\mathsf{n}},(s_u)_k;\rho_k) - \mathrm{rpred}(r_k^{\mathsf{t}};\rho_k)\right| + \left|\mathrm{rpred}(r_k^{\mathsf{t}};\rho_k)\right| \\
&\le \kappa_9\Delta_k\|s_k\| + \kappa_{10}\rho_k\|s_k\|^3 + \kappa_{11}\rho_k\|s_k\|^2\|C_k\| + \eta_0\left|\mathrm{pred}(s_k^{\mathsf{n}},(s_u)_k;\rho_k)\right|. \qquad (4.26)
\end{aligned}$$

Similarly, combining (4.19) with (4.10) gives

$$|\mathrm{ared}(s_k;\rho_k) - \mathrm{pred}(s_k^{\mathsf{n}},(s_u)_k;\rho_k)| \le \kappa_{12}\rho_k\Delta_k\|s_k\| + \eta_0\left|\mathrm{pred}(s_k^{\mathsf{n}},(s_u)_k;\rho_k)\right|. \qquad (4.27)$$

The estimates (4.26) and (4.27) are used in the analysis only when rejection occurs in step i2.8. If s_k is rejected, we know that

$$0 < 1 - \eta_1 \leq \left| \frac{\mathrm{ared}(s_k; \rho_k)}{\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k)} - 1 \right|,$$

which in our inexact context implies

$$1 - \eta_1 \leq \left| \frac{\mathrm{ared}(s_k; \rho_k) - \mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k) - \mathrm{rpred}(r_k^{\mathsf{t}}; \rho_k)}{\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k)} \right| + \eta_0.$$

Thus, when the estimate (4.19) is required, we obtain

$$0 < 1 - \eta_0 - \eta_1 \leq \frac{\kappa_{12} \rho_k \Delta_k \|s_k\|}{\mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k)},$$

and the analysis in [10] remains unchanged except for the fact that a different lower bound $1 - \eta_0 - \eta_1 \in (0, 1)$ is used. A similar bound is obtained when the estimate is given by (4.18).

The proof of (4.17) follows from the conjunction of (4.16) with (4.3) and (3.13). □

5. Implementation in the presence of inexactness. In this section we discuss how the requirements on the approximate reduced gradient and on the step components introduced in section 4 can be satisfied in practice. Our discussion leads to an implementable version of Algorithm 4.3. However, other implementations are possible. This section is not meant to be comprehensive. Rather, it is meant to support our claim made in the introduction and at the beginning of section 4 that our conditions on the allowable level of inexactness are general but implementable.

5.1. Computation of the quasi-normal component. The quasi-normal component s_k^{n} is an approximate solution of the trust-region subproblem (3.3) and it is required to satisfy condition (3.4).

If $\|(s_y^{\mathsf{n}})_k\| \leq \Delta_k$ satisfies the fraction of the Cauchy decrease condition

$$\begin{aligned} &\tfrac{1}{2}\|C_y(x_k)(s_k^{\mathsf{n}})_y + C_k\|^2 \\ &\leq \min\left\{ \tfrac{1}{2}\|C_y(x_k)s + C_k\|^2 \ : \ s = -tC_y(x_k)^T C_k \,,\ \|s\| \leq \Delta_k \right\} , \end{aligned} \tag{5.1}$$

then a result due to Powell [29, Theorem 4] (see also [5, section 6.3], [27, Lemma 4.8]) shows that (3.4) is satisfied. The papers [17, 32] describe two iterative methods based on Krylov subspaces for the computation of steps $(s_y^{\mathsf{n}})_k$ satisfying

$$\|C_k\|^2 - \|C_y(x_k)(s_k^{\mathsf{n}})_y + C_k\|^2 \geq \beta\Big(\|C_k\|^2 - \|C_y(x_k)(s_y^{\mathsf{n}})_* + C_k\|^2 \Big),$$

where $(s_y^{\mathsf{n}})_*$ is the solution of (3.3). In particular, these steps also satisfy (3.4). The iterative method in [32] uses a restart technique that allows the specification of storage limitations by the user, which is important for large scale problems. The iterative methods in [17] and in [32] require the evaluation of $C_y(x_k)v$ and $C_y(x_k)^T u$ for given v and u.

For some applications, the evaluation of matrix-vector products $C_y(x_k)^T v$ is more expensive than the evaluation of $C_y(x_k)v$, and therefore it may be more efficient to use methods that avoid the use of $C_y(x_k)^T v$. In this case, one can apply nonsymmetric

Krylov subspace methods based on minimum residual approximations, such as the GMRES(l) algorithm [30]. In the context of nonlinear system solving, the use of such methods is described, e.g., in [2]. In that context, trust-region subproblems of the type (3.3) also have to be solved, and the solvers in [2] can be applied in our situation as well. If GMRES(l) is used to project the quasi-normal step problem (3.3) onto the l-dimensional Krylov subspace, and if

$$\frac{1}{2}C_k^T\Big(C_y(x_k)^T + C_y(x_k)\Big)C_k \geq \beta\|C_k\|^2 \tag{5.2}$$

holds with $\beta > 0$, then (3.4) is satisfied. Condition (5.2) is implied by the positive definiteness of the symmetric part of $C_y(x_k)$, a condition also important for the convergence of nonsymmetric Krylov subspace methods. A proof of this result and more details concerning the use of these methods can be found in [36].

Finally, we can also use the following simple procedure. Compute $\widetilde{s}_k^{\mathrm{n}}$ such that $\|C_y(x_k)\widetilde{s}_k^{\mathrm{n}} + C_k\| \leq \zeta\|C_k\|$, where $\zeta < 1$, and then scale this step back into the trust-region, i.e., set

$$s_k^{\mathrm{n}} = \begin{pmatrix} \xi_k\widetilde{s}_k^{\mathrm{n}} \\ 0 \end{pmatrix}, \quad \text{where } \xi_k = \begin{cases} 1 & \text{if } \|\widetilde{s}_k^{\mathrm{n}}\| \leq \Delta_k, \\ \Delta_k/\|\widetilde{s}_k^{\mathrm{n}}\| & \text{otherwise.} \end{cases}$$

The step s_k^{n} also satisfies (3.4) (see [36]).

5.2. Computation of an approximate reduced gradient. We show how (4.3) can be enforced, if errors in the reduced gradient are due to inexact linear system solves.

If we set $d = H_k s_k^{\mathrm{n}} + \nabla_x \ell_k$ and denote the y- and u-component of d by d_y and d_u, respectively, then $W_k^T(H_k s_k^{\mathrm{n}} + \nabla_x \ell_k) = -(C_u)_k^T (C_y)_k^{-T} d_y + d_u$. We suppose that the inexactness in the computation of $W_k^T(H_k s_k^{\mathrm{n}} + \nabla_x \ell_k)$ is due to the use of an iterative solver for the linear system $(C_y)_k^T z = -d_y$. More precisely, we assume that

$$\widehat{g}_k = (C_u)_k^T \widehat{z} + d_u, \tag{5.3}$$

where $\widehat{z}$ satisfies

$$(C_y)_k^T \widehat{z} = -d_y - e \tag{5.4}$$

with a residual error e. The following result is easy to prove.

LEMMA 5.1. *If $\widehat{g}_k$ is given by* (5.3), (5.4) *and if*

$$\|e\| \leq \min\{c_1\|(C_u)_k^T\widehat{z} + d_u\|, c_2\Delta_k\}, \tag{5.5}$$

where $c_1, c_2 > 0$ are given, then inequality (4.3) *is satisfied with $\xi_1 = \max\{c_1, c_2\}\|(C_u)_k^T (C_y)_k^{-T}\|$.*

Proof. Equations (5.3) and (5.4) imply $\widehat{g}_k = -(C_u)_k^T (C_y)_k^{-T}(d_y + e) + d_u$ and

$$\|\widehat{g}_k - W_k^T(H_k s_k^{\mathrm{n}} + \nabla_x \ell_k)\| = \|(C_u)_k^T (C_y)_k^{-T} e\| \leq \|(C_u)_k^T (C_y)_k^{-T}\|\ \|e\|.$$

Hence, using (5.3), (5.5),

$$\|\widehat{g}_k - W_k^T(H_k s_k^{\mathrm{n}} + \nabla_x \ell_k)\| \leq \|(C_u)_k^T (C_y)_k^{-T}\|\ \min\{c_1\|\widehat{g}_k\|, c_2\Delta_k\},$$

which yields the desired estimate. ∎

At first sight, the inequality (5.5) seems impractical since both e and $(C_u)_k^T\widehat{z}+d_u$ depend on $\widehat{z}$. However, (5.5) can be enforced if an iterative method for the solution of $(C_y)_k^T z = -d_y$ is used, and matrix-vector products of the form $(C_u)_k^T v$ for a given v can be easily computed. The latter is the case for many control problems. In fact, let $z^{(j)}$ be the jth iterate in the solution method for $(C_y)_k^T z = -d_y$ and let $e^{(j)} = -d_y - (C_y)_k^T z^{(j)}$ be the corresponding residual. If $(C_u)_k^T z^{(j)}$ can be easily computed, then we can monitor $\|(C_u)_k^T z^{(j)} + d_u\|$ and we can truncate the iterative linear system solver when

$$\|e^{(j)}\| \leq \min\{c_1\|(C_u)_k^T z^{(j)} + d_u\|, c_2\Delta_k\}.$$

Note that the truncation criterion (5.5) for the iterative linear system solver is only applicable because $\xi_1 > 0$ in (4.3) is not restricted. If it were required that $\xi_1 \in (0,1)$, say, then we would need an estimate for $\|(C_u)_k^T(C_y)_k^{-T}\|$. Thus, while (4.3) is slightly stronger than (4.2), the fact that we can give up the restriction $\xi_1 < 1$ makes (4.3) preferable from an implementation point of view.

5.3. Computation of the u-component of the tangential component. An approximate solution s_u of (4.1) that satisfies (4.6) can be computed, e.g., using the conjugate gradient (cg) method with a modification as suggested by Steihaug [33] and Toint [34]. Here the cg method with starting value $s_u = 0$ is applied to the minimization of $\widehat{m}_k$. The cg method is stopped if an approximate minimum of the quadratic model $\widehat{m}_k$ is reached, if negative curvature is detected, or if the iterates leave the trust-region bound. The first iterate in the Steihaug–Toint cg method is the Cauchy-step for $\widehat{m}_k$, and therefore (4.6) is satisfied for the first iterate of the Steihaug–Toint cg method. If $W_k^T H_k W_k$ can be applied exactly, which is the case in a reduced SQP method where $W_k^T H_k W_k = \widehat{H}_k$, then the cg method ensures that $\widehat{m}_k$ decreases monotonically, and (4.6) remains satisfied for all Steihaug–Toint cg iterates. If $W_k^T H_k W_k$ is applied inexactly, then one has to compare the function values $\widehat{m}_k$ at the first Steihaug–Toint cg iterate s_u^1 and at the final Steihaug–Toint cg iterate s_u^f. If $\widehat{m}_k(s_u^f) \leq \widehat{m}_k(s_u^1)$, then $(s_u)_k = s_u^f$; otherwise $(s_u)_k = s_u^1$.

5.4. Computation of the y-component of the tangential component. In section 4.1.4 we have already shown that (4.10), (4.11) are satisfied if $(s_y^{\mathsf{t}})_k$ satisfies $C_y(x_k)s_y^{\mathsf{t}} = -C_u(x_k)(s_u)_k + r_k^{\mathsf{t}}$ with residual

$$\|r_k^{\mathsf{t}}\| \;\leq\; \min\left\{\xi_3\Delta_k\|(s_u)_k\|, -\sigma + \sqrt{\sigma^2 + \eta_0 \mathrm{pred}(s_k^{\mathsf{n}}, (s_u)_k; \rho_k)/\rho_k}\right\}, \tag{5.6}$$

where $\sigma = \|C_y(x_k)(s_y^{\mathsf{n}})_k + C_k\| + \|\Delta\lambda_k\|/(2\rho_k)$. Note that all quantities on the right-hand side of (5.6) are known by the time $(s_y^{\mathsf{t}})_k$ needs to be computed.

6. Conclusions. In this paper we have extended the design of a class of composite-step trust-region SQP algorithms and their convergence theory to allow the use of inexact first-order derivative information or the use of inexact linearized constraint equation solves. The challenge was the formulation of accuracy requirements that are sufficient to guarantee global convergence to a point satisfying the first-order optimality conditions, but at the same time can be implemented in a practical algorithm without being overly stringent. Our accuracy requirements are based on the structure of the composite-step trust-region SQP algorithms, and they follow the SQP philosophy which allows one to trade gains in feasibility for gains in optimality. The main motivation of this paper is the control of inexactness arising from

iterative system solves (1.2) in trust-region SQP methods. This is important, e.g., for the solution of discretized optimal control problems governed by partial differential equations. However, our assumptions on the inexactness are not based on this particular source of inexactness and are applicable more broadly.

We focused on a specific class of problems (1.1) and on a limited class of algorithms to enhance the clarity of our presentation. An extension of our analysis of the influence of inexact first-order derivative information, or the use of inexact linearized constraint equation solves, to a broader range of problems and global SQP algorithms is useful. Some extensions are rather straightforward, although tedious. For example, we believe that our analysis can be generalized to the affine-scaling interior-point trust-region SQP algorithms in [11], which tackle problems (1.1) with additional simple bounds on u. In fact, the predecessor [20] of this paper contains many of the technical details of such an extension, although the assumptions on the inexactness made in [20] are stronger than those in this paper.

Acknowledgments. The authors would like to thank the two anonymous referees and the associate editor for their constructive comments on the first version of this paper, which lead to significant improvements in the presentation.

REFERENCES

[1] L. T. Biegler, J. Nocedal, and C. Schmid, *A reduced Hessian method for large-scale constrained optimization*, SIAM J. Optim., 5 (1995), pp. 314–347.

[2] P. N. Brown and Y. Saad, *Convergence theory of nonlinear Newton–Krylov algorithms*, SIAM J. Optim., 4 (1994), pp. 297–330.

[3] R. G. Carter, *On the global convergence of trust region algorithms using inexact gradient information*, SIAM J. Numer. Anal., 28 (1991), pp. 251–265.

[4] E. M. Cliff, M. Heinkenschloss, and A. Shenoy, *Airfoil design by an all-at-once method*, Int. J. Comput. Fluid Mech., 11 (1998), pp. 3–25.

[5] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Trust-Region Methods*, SIAM, Philadelphia, 2000.

[6] A. R. Conn, L. N. Vicente, and C. Visweswariah, *Two-step algorithms for nonlinear optimization with structured applications*, SIAM J. Optim., 9 (1999), pp. 924–947.

[7] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.

[8] R. S. Dembo and U. Tulowitzki, *Local Convergence Analysis for Successive Inexact Quadratic Programming Methods*, Technical Report SOM Series B # 78, Department of Computer Science, Yale University, New Haven, CT, 1984.

[9] R. S. Dembo and U. Tulowitzki, *Sequential truncated quadratic programming methods*, in Numerical Optimization 1984, P. T. Boggs, R. H. Byrd, and R. B. Schnabel, eds., SIAM, Philadelphia, 1985, pp. 83–101.

[10] J. E. Dennis, M. El-Alem, and M. C. Maciel, *A global convergence theory for general trust-region-based algorithms for equality constrained optimization*, SIAM J. Optim., 7 (1997), pp. 177–207.

[11] J. E. Dennis, M. Heinkenschloss, and L. N. Vicente, *Trust-region interior-point SQP algorithms for a class of nonlinear programming problems*, SIAM J. Control Optim., 36 (1998), pp. 1750–1794.

[12] J. E. Dennis and H. F. Walker, *Inaccuracy in quasi-Newton methods: Local improvement theorems*, in Mathematical Programming Study 22: Mathematical Programming at Oberwolfach II, North–Holland, Amsterdam, The Netherlands, 1984, pp. 70–85.

[13] P. Deuflhard, *Global inexact Newton methods for very large scale nonlinear problems*, Impact of Computing in Science and Engineering, 4 (1991), pp. 366–393.

[14] S. C. Eisenstat and H. F. Walker, *Globally convergent inexact Newton methods*, SIAM J. Optim., 4 (1994), pp. 393–422.

[15] R. Fontecilla, *On inexact quasi-Newton methods for constrained optimization*, in Numerical Optimization 1984, P. T. Boggs, R. H. Byrd, and R. B. Schnabel, eds., SIAM, Philadelphia, 1985, pp. 102–118.

[16] O. Ghattas and J.-H. Bark, *Optimal control of two- and three-dimensional Navier–Stokes flow*, J. Comput. Phys., (1997), pp. 231–244.
[17] G. H. Golub and U. von Matt, *Quadratically constrained least squares and quadratic problems*, Numer. Math., 59 (1991), pp. 561–580.
[18] M. D. Gunzburger, L. S. Hou, and T. P. Svobotny, *Optimal control and optimization of viscous, incompressible flows*, in Incompressible Computational Fluid Dynamics, M. D. Gunzburger and R. A. Nicolaides, eds., Cambridge University Press, Cambridge, New York, 1993, pp. 109–150.
[19] M. Heinkenschloss, *Projected sequential quadratic programming methods*, SIAM J. Optim., 6 (1996), pp. 373–417.
[20] M. Heinkenschloss and L. N. Vicente, *Analysis of Inexact Trust-Region Interior-Point SQP Algorithms*, Technical Report TR95–18, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1995, also available online from http://www.caam.rice.edu/ ∼heinken/papers/Papers.html.
[21] K. Ito and K. Kunisch, *Augmented Lagrangian–SQP methods for nonlinear optimal control problems of tracking type*, SIAM J. Control Optim., 34 (1996), pp. 874–891.
[22] S. Ito, C. T. Kelley, and E. W. Sachs, *Inexact primal-dual interior-point iteration for linear programs in function spaces*, Comput. Optim. Appl., 4 (1995), pp. 189–201.
[23] H. Jäger and E. W. Sachs, *Global convergence of inexact reduced SQP-methods*, Optim. Methods Softw., 7 (1997), pp. 83–110.
[24] A. Jameson, L. Martinelli, and N. A. Pierce, *Optimum aerodynamic design using the Navier-Stokes equations*, Theoretical and Computational Fluid Dynamics, 10 (1998), pp. 213–237.
[25] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, SIAM, Philadelphia, 1995.
[26] F. Leibfritz and E. W. Sachs, *Inexact SQP interior point methods and large scale optimal control problems*, SIAM J. Control Optim., 38 (2000), pp. 272–293.
[27] 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, Heidelberg, New York, 1983, pp. 258–287.
[28] S. Omatu and J. H. Seinfeld, *Distributed Parameter Systems: Theory and Applications*, Oxford University Press, Oxford, New York, Toronto, 1989.
[29] M. J. D. Powell, *Convergence properties of a class of minimization algorithms*, in Nonlinear Programming 2, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, eds., Academic Press, New York, 1975, pp. 1–27.
[30] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comp., 7 (1986), pp. 856–869.
[31] V. Schulz and G. Wittum, *Multigrid optimization methods for stationary parameter identification problems in groundwater flow*, in Multigrid Methods V, Springer-Verlag, Berlin, Heidelberg, New York, 1997; also available online from http://dom.ica3.uni-stuttgart.de/∼volker/papers.html.
[32] D. C. Sorensen, *Minimization of a large-scale quadratic function subject to spherical constraint*, SIAM J. Optim., 7 (1997), pp. 141–161.
[33] T. Steihaug, *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., 20 (1983), pp. 626–637.
[34] P. L. Toint, *Towards an efficient sparsity exploiting Newton method for minimization*, in Sparse Matrices and Their Uses, I. S. Duff, ed., Academic Press, New York, 1981, pp. 57–87.
[35] P. L. Toint, *Global convergence of a class of trust-region methods for nonconvex minimization in Hilbert space*, IMA J. Numer. Anal., 8 (1988), pp. 231–252.
[36] L. N. Vicente, *Trust–Region Interior–Point Algorithms for a Class of Nonlinear Programming Problems*, Ph.D. thesis, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1996.

SIAM J. OPTIM.
Vol. 12, No. 2, pp. 303–321

ON THE RESOLUTION OF THE GENERALIZED NONLINEAR COMPLEMENTARITY PROBLEM*

ROBERTO ANDREANI†, ANA FRIEDLANDER‡, AND SANDRA A. SANTOS‡

Abstract. Minimization of a differentiable function subject to box constraints is proposed as a strategy to solve the generalized nonlinear complementarity problem (GNCP) defined on a polyhedral cone. It is not necessary to calculate projections that complicate and sometimes even disable the implementation of algorithms for solving these kinds of problems. Theoretical results that relate stationary points of the function that is minimized to the solutions of the GNCP are presented. Perturbations of the GNCP are also considered, and results are obtained related to the resolution of GNCPs with very general assumptions on the data. These theoretical results show that local methods for box-constrained optimization applied to the associated problem are efficient tools for solving the GNCP. Numerical experiments are presented that encourage the use of this approach.

Key words. box-constrained optimization, complementarity

AMS subject classifications. 65H10, 90C33, 90C30

PII. S1052623400377591

1. Introduction.

The generalized nonlinear complementarity problem (GNCP) is to find $x \in \mathbb{R}^m$ such that

$$F(x) \in \mathcal{K}, \quad G(x) \in \mathcal{K}^\circ, \quad F(x)^T G(x) = 0, \tag{1}$$

where F and G are continuous functions from $\mathbb{R}^m$ to $\mathbb{R}^n$, $\mathcal{K}$ is a nonempty closed convex cone in $\mathbb{R}^n$, and $\mathcal{K}^\circ$ denotes the polar cone of $\mathcal{K}$.

We consider the case $n = m$, $F, G \in C^1$, and $\mathcal{K}$ a polyhedral cone in R^n; that is, given $A \in \mathbb{R}^{q\times n}$ and $B \in \mathbb{R}^{s\times n}$, we have

$$\mathcal{K} = \{v \in \mathbb{R}^n \,|\, Av \geq 0,\ Bv = 0\}$$

and

$$\mathcal{K}^\circ = \{u \in \mathbb{R}^n \,|\, u = A^T\lambda_1 + B^T\lambda_2,\ \lambda_1 \geq 0\}.$$

This problem has many interesting applications, and its solution using special techniques has been considered extensively in the literature. See [16, 17, 24] among others. If $\mathcal{K} = \mathbb{R}^m_+ \equiv \{x \in \mathbb{R}^m \,|\, x \geq 0\}$, $G(x) = x - F(x)$, and $F : \mathbb{R}^m \to \mathbb{R}^m$, the GNCP$(F, G, \mathcal{K})$ reduces to the so-called *implicit complementarity problem* [20, 21]. In particular, if $G(x) = x$, the GNCP reduces to the *nonlinear complementarity problem*, denoted by NCP.

Our approach in this paper is to formulate the GNCP as an equivalent bound-constrained smooth optimization problem. Differentiable bound-constrained minimization is a well-developed area of practical optimization, and many methods and reliable software are available for large-scale problems. See, for example, [7, 8, 12, 26].

*Received by the editors September 5, 2000; accepted for publication (in revised form) May 11, 2001; published electronically November 7, 2001.

http://www.siam.org/journals/siopt/12-2/37759.html

†Department of Computer Science and Statistics, University of the State of São Paulo (UNESP), CP 136, CEP 15054-000, São José do Rio Preto SP, Brazil (andreani@nimitz.dcce.ibilce.unesp.br). This author was supported by FAPESP and CNPq.

‡Department of Applied Mathematics, IMECC-UNICAMP, University of Campinas, CP 6065, 13081-970 Campinas SP, Brazil (friedlan@ime.unicamp.br, sandra@ime.unicamp.br). The second and third authors were supported by FAPESP, CNPq, FINEP, and FAEP-UNICAMP.

This motivated the authors to find equivalences between variational and complementarity problems and smooth box-constrained minimization problems (see [1, 13, 14, 15]).

We prove here that the GNCP is equivalent to a bound-constrained optimization problem in the sense that a global minimizer with zero objective function value is a solution of the GNCP. We also establish conditions for proving that stationary points of the minimization problems are global minimizers and, consequently, solutions of the GNCP. The GNCP (or GCP in other references) is a problem related to the *variational inequality problem* (VIP). The VIP and other related problems were reformulated by many authors as different minimization problems and systems of equations. See [18, 22, 24, 25]. The reformulations of related problems as bound-constrained problems in [1, 14, 15] that use the same approach as the one presented here cannot be extended to obtain a merit function with the properties of the reformulation proposed in this paper. As pointed out by one of the referees, the GNCP can be reformulated as a *mixed complementarity problem* (MCP). In [4], Andreani and Martínez prove results for the MCP based on their work on the bounded VIP [5]. These results applied to the GNCP lead to sufficient conditions on the functions F and G stronger than the ones obtained in this paper. The sufficient conditions given in this paper on the functions F and G that guarantee that stationary points of the merit function solve the GNCP cannot be obtained from any of the previous results.

The objective functions of the minimization problems have a very simple structure that consists of a sum of terms that are polynomials in the original problem data plus an additional term of the type $(x^T z)^p$, with $p > 1$. This term plays a fundamental role in the proof of the equivalence results, and $p = 2$ is especially interesting for linear programming and linear complementarity problems, because in these cases the objective function to be minimized is just a polynomial of fourth degree. It is important to remark that no penalty parameters are needed in these problem formulations, which we call the *quartic approach*. In [1, 13, 14, 15] some very simple counterexamples show that when $p = 1$ the existence of stationary points that are not global minimizers is possible. For the complementarity problem, [1, Theorem 2.4] shows that if F' is positive definite, the merit function with $p = 1$ is such that its stationary points are solutions of the original problem.

These merit functions preserve all the derivatives of the functions that define the GNCP. Consequently, the global and local convergence properties depend on the algorithm used for box-constrained minimization. This is a very important feature, since it makes viable the use of algorithms that need high-order derivatives or their approximations, such as the tensor methods of [23]. Any efficient algorithm for smooth box-constrained minimization can be used, in particular, algorithms that do not rest upon matrix factorizations at all, allowing us to deal with large-scale problems.

Complementarity and related problems have also been solved using algorithms based on the projection equation. See [10] and references therein. These methods are very efficient; however, their behavior is strongly dependent on the monotonocity of the function that defines the problem. Failure of this condition results in divergence of the sequences generated by these algorithms. Unlike the formulations in [22, 25], the computation of the objective function of the equivalent minimization problem considered here is straightforward, and projections on convex sets are not necessary to compute either the objective function or the derivatives. Therefore, special algorithms for dealing with nonsmoothness do not need to be devised. In [24], to obtain the fundamental equivalence result for a cone that is not necessarily polyhedral, the

authors assume the same conditions on the problem as we do here. However, even for polyhedral cones, the implementation of the algorithm proposed there requires projections that, in general, are very expensive to compute.

Using the same merit function of [17], a stronger result is obtained in [16] where the GNCP is reformulated as a system of semismooth equations, and an unconstrained differentiable formulation is given if $\mathcal{K}$ is the positive orthant. The conditions established to ensure that a stationary point x_* of the unconstrained minimization problem is a solution of the GNCP are essentially that the Jacobian of F at x_* (denoted by $F'(x_*)$) is invertible and that $D(G'(x_*)[F'(x_*)]^{-1})_{RR}D$ is an S_0-matrix, where D is a convenient nonsingular diagonal matrix and R is the set of indices for which (1) does not hold at x_*. ($B \in \mathbb{R}^{n\times n}$ is an S_0-*matrix* if there exists $v \in \mathbb{R}^n$ such that $v \geq 0$, $v \neq 0$, and $Bv \geq 0$.) A trust-region method is proposed in [16] for solving the GNCP based on these reformulations. This algorithm was implemented by the authors and tested for some problems.

In [17] an unconstrained minimization reformulation of the GNCP is considered such that the merit function is differentiable when $\mathcal{K} = \mathbb{R}^n_+$. The sufficient conditions for a stationary point x_* of the merit function to be a global minimizer are that $F'(x_*)$ is nonsingular and the product $G'(x_*)[F'(x_*)]^{-1}$ is a P_0-matrix. ($B \in \mathbb{R}^{n\times n}$ is a P_0-*matrix* if its principal minors are all nonnegative.) The authors suggest the use of a first-order method for minimizing the merit function due to the fact that it is once but not twice continuously differentiable.

The case of a general cone $\mathcal{K}$ was considered in [24], using an unconstrained reformulation for the GNCP. It is proved there that x_* is a solution of the GNCP if $F'(x_*)$ is nonsingular and $G'(x_*)[F'(x_*)]^{-1}$ is positive definite. The evaluation of the corresponding objective function is rather complicated and requires projections that in general are not easy to compute.

Here we require, essentially, the same conditions as in [24] to guarantee that stationary points of the minimizing problems are solutions of the GNCP. These assumptions cannot be relaxed for a general cone $\mathcal{K}$ as we show with an example in section 3. If $\mathcal{K} = \mathbb{R}^n_+$, we require a weaker condition on matrix $G'(x_*)F'(x_*)^{-1}$. If F and G are affine functions with $\mathcal{K}$ polyhedral, the conditions are that $G'F'^{-1}$ is positive semidefinite in the null space of B and the GNCP is feasible. Finally, an even weaker condition is needed if F and G are affine and $\mathcal{K} = \mathbb{R}^n_+$.

If $\mathcal{K}$ is a general cone and it is not possible to ensure that $G'F'^{-1}$ is positive definite at a stationary point of the merit function, a sequence of perturbed problems can be constructed for which the strict monotonicity property holds and such that the sequence of solutions of these perturbed problems converges to a solution of the original one. The results related to this construction are valid for a general cone and may be applied also to the results in [24].

The paper is organized as follows: In section 2 we associate with (1) a box-constrained minimization problem, and we prove that assuming a local strict monotonicity condition, stationary points of this problem are solutions of (1). In section 3 we consider perturbations of the original problem that allow us to deal with monotone (not necessarily strict) functions. Numerical experiments are presented in section 4. Finally, conclusions and lines for future research are discussed in section 5.

Notation. We denote by $\langle\cdot,\cdot\rangle$ the Euclidean inner product on $\mathbb{R}^n$ and by $\|\cdot\|$ the norm induced by this inner product and its corresponding matricial norm. If B is a real $n \times n$ matrix, $B \geq 0$ ($B > 0$) means that B is positive semidefinite (positive definite).

2. Equivalence results. The following minimization problem with simple bounds is associated with the GNCP$(F, G, \mathcal{K})$ defined in (1):

$$\begin{aligned} &\min \quad f(x,z,\lambda) \\ &\text{subject to} \quad \begin{cases} z^1 \geq 0, \\ \lambda^1 \geq 0, \end{cases} \end{aligned} \tag{2}$$

where

$$f(x,z,\lambda) = \|RF(x) - z\|^2 + \|G(x) - R^T\lambda\|^2 + \rho\langle z^1, \lambda^1\rangle^2$$

and

$$R = \begin{pmatrix} A \\ B \end{pmatrix}, \quad z = \begin{pmatrix} z^1 \\ 0 \end{pmatrix} \in \mathbb{R}^q \times \mathbb{R}^s, \quad \lambda = \begin{pmatrix} \lambda^1 \\ \lambda^2 \end{pmatrix} \in \mathbb{R}^q \times \mathbb{R}^s.$$

The next theorem states that solving problem GNCP$(F, G, \mathcal{K})$ is equivalent to finding the global minimizer of the optimization problem (2).

THEOREM 1. *If (x_*, z_*, λ_*) is a global minimizer of problem (2) with $f(x_*, z_*, \lambda_*) = 0$, then x_* is a solution of the* GNCP$(F, G, \mathcal{K})$. *Conversely, if x_* is a solution of the* GNCP$(F, G, \mathcal{K})$, *then there exist z_*, λ_* such that (x_*, z_*, λ_*) is a global minimizer of (2) with $f(x_*, z_*, \lambda_*) = 0$.*

Proof. If $f(x_*, z_*, \lambda_*) = 0$, then

$$\begin{aligned} AF(x_*) = z^1 \geq 0, \quad BF(x_*) = 0, \quad &\text{implying that} \quad F(x_*) \in \mathcal{K}, \\ G(x_*) = A^T\lambda_*^1 + B^T\lambda_*^2, \quad &\text{with} \quad \lambda_*^1 \geq 0, \quad \text{so} \quad G(x_*) \in \mathcal{K}^\circ, \end{aligned}$$

and

$$\langle F(x_*), G(x_*)\rangle = \langle F(x_*), R^T\lambda_*\rangle = \langle z_*, \lambda_*\rangle = \langle z_*^1, \lambda_*^1\rangle = 0.$$

Conversely, if x_* is a solution of the GNCP$(F, G, \mathcal{K})$ then there exists $\lambda_* = (\lambda_*^1, \lambda_*^2)$ with $\lambda_*^1 \geq 0$ such that $G(x_*) = A^T\lambda_*^1 + B^T\lambda_*^2$, $z_*^1 = AF(x_*) \geq 0$, and $0 = F(x_*)^T G(x_*) = F(x_*)^T(A^T\lambda_*^1 + B^T\lambda_*^2) = (z_*^1)^T\lambda_*^1 + (BF(x_*))^T\lambda_*^2 = (z_*^1)^T\lambda_*^1$.

Therefore, calling $z_* = (z_*^1, 0)^T$, we have that $f(x_*, z_*, \lambda_*) = 0$. ∎

Global minimizers are very hard to find, especially in large-scale problems. Most efficient large-scale algorithms for box-constrained optimization are guaranteed to converge only to stationary points of the problem. Therefore, it is desirable to relate stationary points of (2) to solutions of the GNCP.

THEOREM 2. *Let $F(x), G(x) \in C^1$. If (x_*, z_*, λ_*) is a stationary point of (2) and $G'(x_*)[F'(x_*)]^{-1}$ is positive definite in the null space of B, then x_* is a solution of the* GNCP$(F, G, \mathcal{K})$.

Proof. Let

$$\begin{aligned} H_* &= G'(x_*)[F'(x_*)]^{-1}, \\ w_* &= AF(x_*) - z_*^1, \\ u_* &= BF(x_*), \\ v_* &= G(x_*) - R^T\lambda_*, \\ \theta_* &= \langle z_*^1, \lambda_*^1\rangle. \end{aligned}$$

If (x_*, z_*, λ_*) is a stationary point of (2), then there exist $\mu \in \mathbb{R}^p_+$ and $\gamma \in \mathbb{R}^s_+$ such that

$$2G'(x_*)^T v_* + 2F'(x_*)^T(A^T w_* + B^T u_*) = 0, \tag{3}$$

$$-2Av_* + 2\rho\theta_* z^1_* - \mu = 0, \tag{4}$$

$$Bv_* = 0, \tag{5}$$

$$-2w_* + 2\rho\theta_*\lambda^1_* - \gamma = 0, \tag{6}$$

$$\langle \mu, \lambda^1_* \rangle = 0, \quad \langle \gamma, z^1_* \rangle = 0, \tag{7}$$

$$\lambda^1_* \geq 0, \quad \mu \geq 0, \quad \gamma \geq 0, \quad z^1_* \geq 0. \tag{8}$$

By (3) we have

$$H^T_* v_* + A^T w_* + B^T u_* = 0. \tag{9}$$

Now, by (4), (6), and (7), we obtain

$$4\langle Av_*, w_* \rangle = 4\rho^2\theta^2_* + \langle \mu, \gamma \rangle, \tag{10}$$

and (5), (9), and (10) imply that

$$\langle v_*, H^T_* v_* \rangle + \langle Av_*, w_* \rangle = \langle v_*, H^T_* v_* \rangle + \rho^2\theta^3_* + \frac{\langle \mu, \gamma \rangle}{4} = 0. \tag{11}$$

Therefore, by (5) and the fact that $\langle v_*, H^T_* v_* \rangle > 0$ in the null space of B, (11) implies

$$\theta_* = 0, \quad \langle v_*, H^T_* v_* \rangle = 0. \tag{12}$$

Since H^T_* is positive definite in the null space of B, by (12), necessarily,

$$v_* = 0. \tag{13}$$

Thus, by (12) and (6),

$$2w_* = -\gamma. \tag{14}$$

If a_i denotes the ith row of matrix A, using (13) and replacing w_* and v_* in (9), we get

$$A^T w_* + B^T u_* = \sum_{i=1}^{q} a_i(\langle a_i, F(x_*) \rangle - (z^1_*)_i) + B^T BF(x_*) = 0. \tag{15}$$

Let

$$\mathcal{I} = \{i \in \{1, \ldots, q\} \mid (z^1_*)_i = 0\};$$

then, if $i \notin \mathcal{I}$, we have that $(z^1_*)_i > 0$. But, by (7), we also have $\gamma_i = 0$. So, by (14),

$$(w_*)_i = \langle a_i, F(x_*) \rangle - (z^1_*)_i = 0 \quad \forall\, i \notin \mathcal{I}. \tag{16}$$

Now, by (15) and (16)

$$A^T w_* + B^T u_* = \sum_{i \in \mathcal{I}} a_i \langle a_i, F(x_*) \rangle + B^T BF(x_*) = 0. \tag{17}$$

Premultiplying (17) by $F(x_*)^T$, we obtain

$$\sum_{i\in\mathcal{I}}\langle a_i, F(x_*)\rangle^2 + \|BF(x_*)\|^2 = 0, \tag{18}$$

and by (18)

$$u_* = BF(x_*) = 0, \quad (w_*)_i = \langle a_i, F(x_*)\rangle = 0 \quad \forall\, i \in \mathcal{I}. \tag{19}$$

Finally, (12), (13), (16), and (19) imply that $f(x_*, z_*, \lambda_*) = 0$. ∎

In the following theorem we show that the hypothesis of Theorem 2 can be relaxed if the functions F and G are affine.

THEOREM 3. *Let $F(x), G(x)$ be affine, $G'F'^{-1}$ positive semidefinite in the null space of B and* GNCP$(F, G, \mathcal{K})$ *feasible. If (x_*, z_*, λ_*) is a stationary point of* (2)*, then x_* is a solution of* GNCP$(F, G, \mathcal{K})$.

Proof. As in Theorem 2, we obtain (3)–(12). Since $\theta_* = 0$, the optimality conditions read as

$$2G'^T v_* + 2F'^T(A^T w_* + B^T u_*) = 0, \tag{20}$$

$$-2Av_* - \mu = 0, \tag{21}$$

$$Bv_* = 0, \tag{22}$$

$$-2w_* - \gamma = 0, \tag{23}$$

$$\langle \mu, \lambda_*^1\rangle = 0, \quad \langle \gamma, z_*^1\rangle = 0, \tag{24}$$

$$\lambda_*^1 \geq 0, \quad \mu \geq 0, \quad \gamma \geq 0, \quad z_*^1 \geq 0. \tag{25}$$

Relations (20)–(25) are the necessary and sufficient conditions for a global minimizer of the following convex quadratic minimization problem:

$$\begin{aligned} &\min \quad f(x, z, \lambda) = \|RF(x) - z\|^2 + \|G(x) - R^T\lambda\|^2 \\ &\text{subject to} \quad \begin{cases} z^1 \geq 0, \\ \lambda^1 \geq 0. \end{cases} \end{aligned} \tag{26}$$

Since, by hypothesis, the GNCP$(F, G, \mathcal{K})$ is feasible, it turns out that (x_*, z_*, λ_*) is a global solution of (26) with objective function value zero, and as $\theta_* = 0$, we get $f(x_*, z_*, \lambda_*) = 0$. ∎

The hypotheses of Theorem 2 can also be weakened if $\mathcal{K} = \mathbb{R}^n_+$, as we show in the following theorem.

DEFINITION 1. *A matrix $B \in \mathbb{R}^{n\times n}$ is* column-sufficient *if for $v \in \mathbb{R}^n$, $v_i(Bv)_i \leq 0\ \forall\, i$ implies $v_i(Bv)_i = 0\ \forall\, i$. A matrix B is called* row-sufficient *if B^T is column-sufficient.*

DEFINITION 2. *A matrix $B \in \mathbb{R}^{n\times n}$ is called an S-*matrix *if there exists $v \in \mathbb{R}^n$ such that $v \geq 0$ and $Bv > 0$.*

THEOREM 4. *Let $\mathcal{K} = \mathbb{R}^n_+$ and $F(x), G(x) \in C^1$. If (x_*, z_*, λ_*) is a stationary point of* (2) *and $G'(x_*)[F'(x_*)]^{-1}$ is a row-sufficient S-matrix, then x_* is a solution of the GNCP$(F, G, \mathcal{K})$.*

Proof. In this case the optimization problem is

$$\begin{aligned} &\min \quad \|F(x) - z\|^2 + \|G(x) - \lambda\|^2 + \rho\langle z, \lambda\rangle^2 \\ &\text{subject to} \quad \begin{cases} z \geq 0, \\ \lambda \geq 0. \end{cases} \end{aligned} \tag{27}$$

Defining

$$w_* = F(x_*) - z_*,$$
$$v_* = G(x_*) - \lambda_*,$$
$$H_* = G'(x_*)[F'(x_*)]^{-1},$$

and

$$\theta_* = z_*^T \lambda_*,$$

the optimality conditions read as

$$2G'(x_*)^T v_* + 2F'(x_*)^T w_* = 0, \tag{28}$$
$$-2v_* + 2\rho\theta_* z_* - \mu = 0, \tag{29}$$
$$-2w_* + 2\rho\theta_* \lambda_* - \gamma = 0, \tag{30}$$
$$\langle \mu, \lambda_* \rangle = 0, \quad \langle \gamma, z_* \rangle = 0, \tag{31}$$
$$\lambda_* \geq 0, \quad \mu \geq 0, \quad \gamma \geq 0, \quad z_* \geq 0. \tag{32}$$

By (29) and (30),

$$4(w_*)_i(v_*)_i = 4\rho^2\theta_*^2(\lambda_*)_i(z_*)_i + \mu_i\gamma_i \tag{33}$$

for $i \in \{1, \ldots, n\}$. We can write (28) as

$$H_*^T v_* + w_* = 0. \tag{34}$$

Therefore, by (33) and (34),

$$4(v_*)_i(H_*^T v_*)_i + 4\rho^2\theta_*^2(\lambda_*)_i(z_*)_i + \mu_i\gamma_i = 0 \tag{35}$$

for $i \in \{1, \ldots, n\}$. Since H_* is row-sufficient, (35) implies that

$$(v_*)_i(H_*^T v_*)i = 0 \quad \text{for } i \in \{1, \ldots, n\} \quad \text{and} \quad \theta_* = 0. \tag{36}$$

Using (29), (30), (33), (34), and (36), we have that

$$H_*^T v_* = -w_* = \frac{\gamma}{2} \geq 0 \tag{37}$$

and

$$v_* = -\frac{\mu}{2} \leq 0. \tag{38}$$

If $v_* \neq 0$, (37) and (38) contradict the fact that H_* is an S-matrix (see [11]), and therefore

$$v_* = 0, \quad w_* = 0. \tag{39}$$

Finally, by (36) and (39), $f(x_*, z_*, \lambda_*) = 0$. ∎

If F and G are affine functions and $\mathcal{K}$ is the positive orthant, then the following result holds.

THEOREM 5. *Let $\mathcal{K} = \mathbb{R}^n_+$, $F(x)$, $G(x)$ be affine such that $G'F'^{-1}$ is a row-sufficient matrix. If* GNCP$(F, G, \mathcal{K})$ *is feasible and (x_*, z_*, λ_*) is a stationary point of* (2)*, then x_* is a solution of* GNCP$(F, G, \mathcal{K})$.

Proof. As in Theorem 4, we obtain (28) and (36). The rest of the proof mimics that of Theorem 3. ∎

Remark. The results of Theorems 4 and 5 are valid with the following hypothesis: There exists a partition of $I = \{1, \ldots, n\}$, $I = [I_0, I_1]$, where

$$\widetilde{F}^T = (F^T_{i\in I_0}, G^T_{i\in I_1}) \quad \text{and} \quad \widetilde{G}^T = (G^T_{i\in I_0}, F^T_{i\in I_1})$$

such that $\widetilde{G}'(x_*)[\widetilde{F}'(x_*)]^{-1}$ is a row-sufficient S-matrix or just row-sufficient if F and G are affine.

3. Perturbed problems. The finite *variational inequality problem* VIP$(\widehat{F}, \Omega)$, where $\widehat{F} : \mathbb{R}^n \to \mathbb{R}^n$ and $\Omega \subseteq \mathbb{R}^n$ is a closed convex set, is to find $x \in \Omega$ such that $\langle \widehat{F}(x), y - x\rangle \geq 0 \ \forall\, y \in \Omega$.

In [1], for $\Omega = \{x \in \mathbb{R}^n \,|\, g(x) \leq 0, Bx = c, x \geq 0\}$, where $g = (g_1, \ldots, g_m)^T$, $g_i \in C^1(\mathbb{R}^n)$ is convex $\forall\, i = 1, \ldots, m$, $B \in \mathbb{R}^{q\times n}$, and $c \in \mathbb{R}^q$, the authors reformulated the VIP$(\widehat{F}, \Omega)$ as an equivalent box-constrained smooth optimization problem. The properties of the merit function proposed there are similar to the one considered in section 2 of this paper for the GNCP.

We relate now the GNCP$(F, G, \mathcal{K})$ with the VIP$(G\circ F^{-1}, \mathcal{K})$ whenever F^{-1} exists.

LEMMA 6. *If F^{-1} exists, then x_* is a solution of the* GNCP$(F, G, \mathcal{K})$ *if and only if $F(x_*)$ is a solution of the* VIP$(G\circ F^{-1}, \mathcal{K})$.

Proof. If x_* is a solution of GNCP$(F, G, \mathcal{K})$, then

$$F(x_*) \in \mathcal{K}, \quad G(x_*) \in \mathcal{K}^\circ, \quad \langle F(x_*), G(x_*)\rangle = 0. \tag{40}$$

Since F^{-1} exists,

$$\langle G(x_*), F(x_*)\rangle = \langle G \circ F^{-1}(F(x_*)), F(x_*)\rangle = 0 \tag{41}$$

and, as $G(x_*) \in \mathcal{K}^\circ$,

$$\langle G(x_*), y\rangle \geq 0 \quad \forall\, y \in \mathcal{K}. \tag{42}$$

By (40)–(42), $F(x_*) \in \mathcal{K}$ and

$$\langle G \circ F^{-1}(F(x_*)), y - F(x_*)\rangle \geq 0 \quad \forall\, y \in \mathcal{K}. \tag{43}$$

This implies that $F(x_*)$ is a solution of VIP$(G\circ F^{-1}, \mathcal{K})$.

Conversely, if $F(x_*)$ is a solution of VIP$(G\circ F^{-1}, \mathcal{K})$, then

$$F(x_*) \in \mathcal{K}. \tag{44}$$

So, for $0 \leq \varepsilon \leq 1$,

$$(1 + \varepsilon)F(x_*) \in \mathcal{K} \quad \text{and} \quad (1 - \varepsilon)F(x_*) \in \mathcal{K} \tag{45}$$

and, since (43) holds for any $y \in \mathcal{K}$, we obtain

$$\langle G \circ F^{-1}(F(x_*)), F(x_*)\rangle = \langle G(x_*), F(x_*)\rangle = 0. \tag{46}$$

By (43) and (46), $\langle G(x_*), y\rangle \geq 0 \ \forall\, y \in \mathcal{K}$, so $G(x_*) \in \mathcal{K}^\circ$. Then, by (44) and (46), x_* is a solution of GNCP$(F, G, \mathcal{K})$. ∎

If F and G are affine functions we can guarantee that, if $G\circ F^{-1}$ is (not necessarily strictly) monotone, stationary points of the merit function are solutions of the GNCP.

In general, we can have stationary points of the associated problem that are not solutions of the original problem. Consider, for instance, the following example.

Example. Let $F:\mathbb{R}\to\mathbb{R}$ be defined as $F(x)=x$, $G(x)=-1$ if $x\le 1$ and $G(x)=(x-1)^2-1$ if $x\ge 1$, and $\mathcal{K}=\mathbb{R}_+$. Observe that $G\circ F^{-1}$ is monotone and convex. The GNCP$(F,G,\mathcal{K})$ has the unique solution $x_*=2$. The merit function is given in this case by

$$f(x,v,\lambda)=\begin{cases}(-1-v)^2+(x-\lambda)^2+(\lambda v)^2 & \text{if } x\le 1,\\ ((x-1)^2-1-v)^2+(x-\lambda)^2+(\lambda v)^2 & \text{if } x\ge 1,\end{cases}$$

and $(0,0,0)^T$ is a stationary point of reformulation (2) that corresponds to this problem.

In [1, Theorem 3.2] the authors proved that if $\widehat{F}$ is (not necessarily strictly) monotone, the sequence of solutions of the perturbed problems $\widehat{F}+\varepsilon_k I$, where I is the identity matrix, converges to the unique solution of minimum norm of the VIP$(\widehat{F},\mathcal{K})$.

In a similar way, given a sequence of strictly positive ε_k such that $\varepsilon_k\downarrow 0$, we can associate with the GNCP$(F,G,\mathcal{K})$ a family of perturbed problems, as follows. For all $k\in\mathbb{N}$ and $x\in\mathbb{R}^n$ we define

$$G_k(x)=G(x)+\varepsilon_k F(x).$$

In the following theorems we relate the solutions of the perturbed problems to the solution of GNCP$(F,G,\mathcal{K})$, where $\mathcal{K}$ is not necessarily a polyhedral cone. Thus, these results may be used with the formulation proposed in [24].

THEOREM 7. *If* GNCP$(F,G_k,\mathcal{K})$ *admits a solution* $x_k\ \forall k\in\mathbb{N}$ *and the sequence of solutions* $\{x_k\}$ *is bounded, then every limit point of* $\{x_k\}$ *is a solution of the* GNCP$(F,G,\mathcal{K})$.

Proof. Since $\{x_k\}$ is bounded, it admits a convergent subsequence. Let K_1 be an infinite subset of $\mathbb{N}$, and x_* be such that

$$\lim_{k\in K_1} x_k = x_*.$$

If x_k is a solution of GNCP$(F,G_k,\mathcal{K})$, then

$$F(x_k)\in\mathcal{K},\quad G(x_k)+\varepsilon_k F(x_k)\in\mathcal{K}^\circ,\quad \langle F(x_k),G(x_k)+\varepsilon_k F(x_k)\rangle=0. \tag{47}$$

By the continuity of F and G and the closedness of $\mathcal{K}$, $\lim_{k\in K_1}F(x_k)=F(x_*)\in\mathcal{K}$, $\lim_{k\in K_1}G(x_k)=G(x_*)\in\mathcal{K}^\circ$, and $F(x_*)^TG(x_*)=0$. ∎

Remark. In Theorem 7 there is no assumption of monotonicity on either the original problem or the perturbed ones.

The result of [1, Theorem 3.2] is used next to characterize x_* in the set of solutions of GNCP$(F,G,\mathcal{K})$, denoted by SOL(GNCP$(F,G,\mathcal{K})$). Also, SOL(VIP) denotes the set of solutions of a VIP.

THEOREM 8. *Assume that* $G\circ F^{-1}$ *is monotone and that the set of solutions of the* GNCP$(F,G,\mathcal{K})$ *is not empty. Then the sequence* $\{x_k\}$ *of solutions of the GNCP*$(F,G_k,\mathcal{K})$ *converges to a solution* x_* *of the* GNCP$(F,G,\mathcal{K})$ *that is the unique solution of the problem*

$$\min\|F(x)\| \text{ subject to } x\in \mathrm{SOL}(\mathrm{GNCP}(F,G,\mathcal{K})). \tag{48}$$

Proof. If x_k is a solution of GNCP$(F, G_k, \mathcal{K})$, then, by Lemma 6, $F(x_k)$ is a solution of the VIP$(G_k \circ F^{-1}, \mathcal{K})$.

Since $G \circ F^{-1}(x)$ is monotone and

$$G_k(x) \circ F^{-1}(x) = (G + \varepsilon_k F) \circ F^{-1}(x) = G \circ F^{-1}(x) + \varepsilon_k x, \tag{49}$$

we have that $G_k \circ F^1(x)$ is strictly monotone. As F is an homeomorphism, [1, Theorem 3.2] implies that $\lim_{k\to\infty} F(x_k) = F(x_*)$, where $F(x_*)$ is the unique minimum norm solution of VIP$(G \circ F^{-1}, \mathcal{K})$ and solves the problem

$$\min \|F(x)\| \text{ subject to } F(x) \in \text{SOL(VIP)}.$$

Then, by Lemma 6, x_* is a solution of GNCP$(F, G, \mathcal{K})$ and is the unique solution of

$$\min \|F(x)\| \text{ subject to } x \in \text{SOL(GNCP}(F, G, \mathcal{K})). \qquad \square$$

The results obtained in this section allow us to solve GNCPs such that $G \circ F^{-1}$ is monotone using the approach developed in section 2 for the perturbed problems.

4. Computational experiments. Our set of experiments contains four families: randomly generated problems in the positive orthant, implicit complementarity problems from Outrata and Zowe [19], problems with general cones in $\mathbb{R}^n$, and problems in three-dimensional cones with control of generated faces.

For the first family of problems, functions F and G are affine and both cones are the positive orthant. Although quite simple, these problems contain essential elements to start the investigation. By varying dimensions and features of the matrices that define F and G, we have produced an extensive set of tests for which the theoretical hypothesis of equivalence might hold or not.

In the second family our main objective was to solve problems already addressed in the literature. We also extended the family of implicit complementarity problems proposed in [19] to variable dimension, producing large-scale tests. For such problems, however, the cones are the positive orthant as well.

General polyhedral cones were treated in the third and fourth families of problems. In the third one, functions F and G are affine and the matrices A and B that define the cones are generated to accomplish well defined problems, but without any specific control. In the fourth family, we produced three-dimensional tests, so that geometrical features of the cone, like control of edges and number of faces, were exploited to a great extent.

The equivalent minimization problems (2), with simple bounded variables, were solved using `BOX-QUACAN`, software developed by our research group at the State University of Campinas. It is based on the trust-region approach for solving large-scale bound-constrained minimization and uses the infinity norm to define the trust-region, so that the quadratic subproblems also have simple bounded variables. The subproblems are solved by combining conjugate gradients with projected gradients and a mild active set strategy (see [6, 12] or [9, p. 459]).

The code was developed in Fortran 77 double precision (Microsoft PowerStation) and run on a Pentium 64MB RAM. The stopping criteria used are tolerance for the objective function value $\varepsilon_f = 10^{-10}$ and tolerance for the norm of the continuous projected gradient $\varepsilon_g = 10^{-6}$. We set $\rho = 1$ for all the tests.

4.1. Randomly generated problems in the positive orthant. In our first set of experiments we considered the problem of finding $x \in \mathbb{R}^n$ such that $Mx+c \geq 0$, $Px + d \geq 0$, and $(Mx + c)^T(Px + d) = 0$, where matrices $M, P \in \mathbb{R}^{n\times n}$ and vectors $c, d \in \mathbb{R}^n$ are given.

The problems were randomly generated to exploit specific features of matrices M and P in a total of fourteen families as follows: M and P may be identical (families 1 to 6) or not (families 7 to 14); M and P may be symmetric (families 1 to 3 and 7 to 10) or not (4–6, 11–14); and matrices M and P may be regular (1, 2, 4, 5, 7, 8, 11, and 12) or singular (3, 6, 9, 10, 13, and 14). For each family, four values for the dimension n were used (5, 50, 500, and 5000). For each dimension, three problems were solved, with different seeds. For details on the generation, see [2].

Whenever M or P is invertible, the theoretical hypotheses of the equivalence results of section 2 can be verified by analyzing properties of matrices PM^{-1} or MP^{-1}. There were some problems, from families 8, 12, and 13, that converged to local nonglobal minimizers of (2), with merit function value greater than 10^{-1}. For problems from the first, second, fourth, and fifth sets, the theoretical hypotheses hold, representing 28.5% of the total number of tests. For families 1, 2, 4, 5, and 7, the algorithm computed the same solution that was generated for assembling the problem data. For families 3, 6, 10, and 14, since both matrices M and P are singular, the theoretical hypotheses fail, representing 28.5% of tests. For these tests, however, the global solution of (2) was always obtained. There is no guarantee that the theoretical hypotheses are valid for the test problems of sets 7, 8, 9, 11, 12, and 13, which represent 43% of tests. In fact, in 18 out of the 60 problems of these last six sets, at least one of the values $u^T PM^{-1}u$ or $v^T MP^{-1}v$, where $u = Mx + c - z$ and $v = Px + d - \lambda$, was negative. In the total of 168 problems solved, the hypotheses fail for 66 (39%), but only 16 converged to local solutions of (2), which correspond to 24% of the candidates for failure, and to 9.5% of the total of tests.

4.2. Implicit complementarity problems from Outrata and Zowe. In the second set of experiments we solved implicit complementarity problems (see [19]) of the following form:

Find $y \in \mathbb{R}^n$ such that

$$y - m(y) \geq 0, \quad F(y) \geq 0, \quad \text{and} \quad \langle F(y), y - m(y)\rangle = 0,$$

where $m_i : \mathbb{R}^n \to \mathbb{R},\ i = 1, \ldots, n,$

$$F(y) = Ay + b = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix} y + \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \tag{50}$$

and $m(y) = \varphi(Ay + b)$, with $\varphi : \mathbb{R}^n \to \mathbb{R}^n$ twice continuously differentiable.

As in [19, Examples 4.3 and 4.4], the following choices for function φ defined our test problems:

$$\text{POZ1:} \quad \varphi_i(\lambda) = -0.5 - \lambda_i, \quad i = 1, 2, 3, 4, \quad \text{and}$$

$$\text{POZ2:} \quad \varphi_i(\lambda) = -1.5\lambda_i + 0.25\lambda_i^2, \quad i = 1, 2, 3, 4.$$

For each problem, three starting vectors were used, namely,

(a) $(0.0, 0.0, 0.0, 0.0)^T$,
(b) $(-0.5, -0.5, -0.5, -0.5)^T$,
(c) $(-1.0, -1.0, -1.0, -1.0)^T$.

In [19], Newtonian strategies were adopted to solve problems POZ1 and POZ2. In the *first approach*, the iterative scheme to compute fixed points of an operator S was

$$y_{k+1} = y_k - (E - V^k)^{-1}(y_k - S(y_k)),$$

where $V^k \in \partial S(y_k)$. In the *second approach*, a Newton variant scheme was applied to the semismooth operator

$$H(y) := \min\{y - m(y), F(y)\} = 0,$$

where min denotes the componentwise minimum of the two vectors in brackets.

Problems POZ1 and POZ2 were also solved in [16], with a trust-region approach for solving the GNCP$(F, G, \mathbb{R}^n_+)$ using the merit function $\Phi : \mathbb{R}^n \to R$ defined by

$$\Phi(x) := \frac{1}{2}\sum_{i=1}^{n} \phi(F_i(x), G_i(x))^2.$$

The function $\phi(a, b) = \sqrt{a^2 + b^2} - a - b$ is the Fischer–Burmeister one, with the property $\phi(a, b) = 0 \Leftrightarrow a \geq 0,\ b \geq 0,\ ab = 0$.

In Tables 1 and 2 we present, for comparative purposes, numerical results of [19] and [16] for problems POZ1 and POZ2, respectively. Our results are reported in Table 3, where the notation `INNER`, `MVP`, `OUTER`, and `FE` is used to indicate the number of iterations and matrix-vector products performed by the inner (quadratic) solver, and the number of iterations and functional evaluations performed by the outer (trust-region) algorithm. We also included the final value of our merit function $f(x, z, \lambda)$, together with the norm of the projected gradient $\|g_p\|$ at the final approximation.

TABLE 1
Previous results: Problem 1 (POZ1: $n = 4$).

OZ95			JFQS98		
Start	First approach `ITER`	Second approach `ITER`	`ITER`	`FE`	Φ
(a)	2	14	5	17	7.65D−18
(b)	2	41	4	16	9.71D−15
(c)	V^2 singular	56	5	11	3.43D−24

TABLE 2
Previous results: Problem 2 (POZ2: $n = 4$).

OZ95			JFQS98		
Start	First approach `ITER`	Second approach `ITER`	`ITER`	`FE`	Φ
(a)	3	15	5	17	1.05D−18
(b)	V^2 singular	15	4	16	4.89D−15
(c)	V^2 singular	No convergence	5	11	7.05D−22

The results of our approach compared quite well with [16] and were, by far, superior to the results of [19]. For problem POZ1, starting points (a) and (b) provide similar results in terms of computational effort, although point (b) generates a solution with slightly better quality. For this problem, starting with point (c), on the other hand, requires twice as many inner iterations and matrix-vector products as starting

TABLE 3
Results using our approach ($n = 4$).

Problem	Start	OUTER	FE	INNER	MVP	$f(x, z, \lambda)$	$\|g_p\|$
POZ1	(a)	4	5	24	30	2.31D−10	8.61D−06
	(b)	4	5	22	39	1.55D−14	7.03D−08
	(c)	4	5	45	68	6.63D−11	7.77D−06
POZ2	(a)	5	6	48	74	4.25D−12	2.33D−06
	(b)	6	8	104	171	1.15D−14	8.25D−08
	(c)	3	4	31	60	9.43D−11	2.25D−05

TABLE 4
Additional tests with larger dimensions.

Problem	Start	OUTER	FE	INNER	MVP	$f(x, z, \lambda)$	$\|g_p\|$
POZ1	(a)	7	10	125	236	1.26D−11	5.48D−06
$n = 40$	(b)	6	8	102	313	3.16D−13	4.52D−07
	(c)	5	7	84	176	1.11D−10	6.62D−06
POZ1	(a)	8	12	146	205	2.42D−12	8.69D−07
$n = 400$	(b)	7	10	126	201	5.66D−12	1.59D−06
	(c)	6	8	94	206	1.44D−11	2.32D−06
POZ1	(a)	9	14	143	311	1.59D−12	7.79D−07
$n = 4000$	(b)	8	12	123	377	7.43D−12	2.26D−06
	(c)	7	9	99	289	1.96D−11	2.91D−06
POZ2	(a)	7	11	127	248	4.36D−12	2.45D−06
$n = 40$	(b)	6	9	116	201	1.89D−11	2.56D−06
	(c)	6	8	104	176	6.90D−13	7.44D−07
POZ2	(a)	9	14	143	227	6.64D−13	5.35D−07
$n = 400$	(b)	7	11	135	367	1.75D−11	2.74D−06
	(c)	7	10	120	203	6.30D−13	4.93D−07
POZ2	(a)	10	15	157	394	2.98D−12	9.12D−07
$n = 4000$	(b)	9	14	161	385	7.84D−11	5.18D−06
	(c)	8	12	161	309	1.21D−12	4.94D−07

TABLE 5
Average results of our approach.

Problem	n	OUTER	FE	INNER	MVP
POZ1	4	4.0	5.0	30.3	45.7
	40	6.0	8.3	103.7	241.7
	400	7.0	10.0	122.0	204.0
	4000	8.0	11.7	121.7	325.7
POZ2	4	4.7	6.0	61.0	101.7
	40	6.3	9.3	115.7	208.3
	400	7.7	11.7	132.7	265.7
	4000	9.0	13.7	159.7	362.7

with (a) or (b). For problem POZ2, the starting point that generated the highest cost was (b).

To assess the reliability of our approach, we enlarged the dimension n of problems POZ1 and POZ2, allowing $n = 40$, $n = 400$, and $n = 4000$. Matrix $A \in \mathbb{R}^{n\times n}$ and vector $b \in \mathbb{R}^n$ are the natural extensions of (50), as are the starting vectors (a), (b), and (c). Results are presented in Table 4, where one can see that the computational effort grows very slowly as n increases. The greatest difference happens between $n = 4$ and $n = 40$, but from 40 to 400 and from 400 to 4000 the cost does not grow as much as in the first case. Such differences in the increasing factors can be better appreciated by the average values shown in Table 5.

4.3. Problems with general polyhedral cones in $\mathbb{R}^n$. In this third set of experiments we address the problem of finding $x \in \mathbb{R}^n$ such that $Mx + c \in \mathcal{K}$, $Px + d \in \mathcal{K}^\circ$, and $(Mx + c)^T(Px + d) = 0$, where the sets $\mathcal{K}$, $\mathcal{K}^\circ$ are defined by

$$\mathcal{K} = \{v \in \mathbb{R}^n \,|\, Av \geq 0,\ Bv = 0\},$$
$$\mathcal{K}^\circ = \{u \in \mathbb{R}^n \,|\, u = A^T\lambda_1 + B^T\lambda_2,\ \lambda_1 \geq 0\},$$

with $A \in \mathbb{R}^{q\times n}$, $B \in \mathbb{R}^{s\times n}$ given. Matrices $M, P \in \mathbb{R}^{n\times n}$ and vectors $c, d \in \mathbb{R}^n$ are also given.

The problems were randomly generated quite similarly to our first set of experiments. For details, see [2]. According to the features of matrices M and P, we divided the set of tests into three families: (1) $M = P$, indefinite and nonsymmetric; (2) $M = P$, indefinite and symmetric; (3) $M \neq P$, indefinite, nonsymmetric, and singular. For families (1) and (2) the theoretical hypotheses of the equivalence results hold since $PM^{-1} = I$.

For each family, six sets for the dimensions (n, q, s) were considered: $(10, 5, 1)$, $(10, 10, 1)$, $(10, 15, 1)$, $(100, 50, 5)$, $(100, 100, 5)$, and $(100, 150, 5)$. For each set of dimensions, three problems were generated, with different seeds. The arithmetic means of the results are reported in Tables 6 and 7, where we present the number of iterations (`INNER`) and matrix-vector products (`MVP`) performed by the inner (quadratic) solver, and the number of iterations (`OUTER`) and functional evaluations (`FE`) performed by the outer (trust-region) algorithm.

TABLE 6
Average results: Problems with $n = 10$, $s = 1$.

q	Family	INNER	MVP	OUTER	FE
5	1	136.7	170.3	9.0	10.0
10		184.0	257.0	11.3	12.3
15		309.0	436.7	18.0	19.0
5	2	168.0	213.0	11.3	12.3
10		168.7	232.3	11.8	12.8
15		208.3	282.3	12.7	13.7
5	3	208.7	253.3	10.0	11.0
10		278.7	371.7	13.0	14.0
15		485.7	640.7	19.7	20.7

TABLE 7
Average results: Problems with $n = 100$, $s = 5$.

q	Family	INNER	MVP	OUTER	FE
50	1	1021.0	1373.0	35.7	36.7
100		2199.3	2971.3	72.0	73.0
150		3946.3	5103.0	113.3	114.0
50	2	1064.7	1421.0	37.0	38.0
100		2167.7	2833.0	67.3	68.3
150		4291.3	5720.3	124.3	125.3
50	3	7397.7	7922.0	101.0	102.0
100		160724.0	166259.0	1856.0	1857.0
150		102189.0	112886.0	957.3	963.0

We denote the figures of Tables 6 and 7 by T^k_{ij}, where $k \in \{1, 2, 3\}$ represents each family, $i \in \{1, 2, 3\}$ corresponds to rows with $q = 5, 10, 15$ (Table 6), $i \in \{4, 5, 6\}$ corresponds to rows with $q = 50, 100, 150$ (Table 7), and $j \in \{1, 2, 3, 4\}$ is the corre-

sponding column with the values `INNER`, `MVP`, `OUTER`, and `FE`. Based on these values, we define cost measures to guide our analysis.

Concerning the effort spent by the algorithm, there are two aspects we would like to address: How is such effort related to the problem dimension, and how is it related to the problem features? Considering each dimension separately, we started by defining two cost measures, per inner iteration (`MVP`/`INNER`) and global (`INNER`/`OUTER`), as follows:

$$me_1(i) = \frac{1}{3}\sum_k \frac{T_{i2}^k}{T_{i1}^k} \quad \text{and} \quad me_2(i) = \frac{1}{3}\sum_k \frac{T_{i1}^k}{T_{i3}^k}$$

for $i = 1, 2, 3, 4, 5, 6$.

For a better understanding of the average values represented by these two measures, we also computed the minimum and maximum values:

$$m_1(i) = \min_k \frac{T_{i2}^k}{T_{i1}^k}, \quad M_1(i) = \max_k \frac{T_{i2}^k}{T_{i1}^k}, \quad m_2(i) = \min_k \frac{T_{i1}^k}{T_{i3}^k}, \quad \text{and} \quad M_2(i) = \max_k \frac{T_{i1}^k}{T_{i3}^k}.$$

Results are reported in Table 8, where the triples contain

$$(m_1(i), me_1(i), M_1(i)) \quad \text{and} \quad (m_2(i), me_2(i), M_2(i))$$

for $i = 1, \ldots, 6$.

TABLE 8
Measures of effort per problem dimension.

Dimension (q)	(m_1, me_1, M_1)	(m_2, me_2, M_2)
5	(1.22, 1.24, 1.26)	(14.69, 16.54, 20.03)
10	(1.34, 1.36, 1.38)	(14.79, 17.50, 21.34)
15	(1.33, 1.37, 1.42)	(16.49, 19.28, 24.11)
50	(1.09, 1.25, 1.34)	(28.24, 41.04, 66.12)
100	(1.06, 1.24, 1.35)	(30.49, 48.74, 83.55)
150	(1.09, 1.24, 1.34)	(34.24, 54.99, 95.91)

With the aim of analyzing results according to the family of generated problems, we define two additional measures for each one of sets 1 to 3. The weights $\ln(n+2q+s)$ and $\sqrt{\ln(n+2q+s)}$ were introduced to filter dependence of dimension and somehow make uniform the computed values:

$$me_3(k) = \frac{1}{6}\left(\sum_{i=1}^{3} \frac{T_{i2}^k}{\ln(11+10i)T_{i1}^k} + \sum_{i=4}^{6} \frac{T_{i2}^k}{\ln(100i-195)T_{i1}^k}\right)$$

and

$$me_4(k) = \frac{1}{6}\left(\sum_{i=1}^{3} \frac{T_{i2}^k}{\sqrt{\ln(11+10i)}T_{i1}^k} + \sum_{i=4}^{6} \frac{T_{i2}^k}{\sqrt{\ln(100i-195)}T_{i1}^k}\right)$$

for $k = 1, 2, 3$. We stress that the values $11+10i$, $i = 1, 2, 3$, and $100i-195, i = 4, 5, 6$ are, respectively, the dimensions 21, 31, 41 and 205, 305, 405 used in the tests. Results are shown in Table 9, where we also include minimum (m_3, m_4) and maximum values (M_3, M_4).

TABLE 9
Measures of effort per problem family.

Family	(m_3, me_3, M_3)	(m_4, me_4, M_4)
1	(0.50, 0.54, 0.58)	(23.08, 37.16, 54.70)
2	(0.51, 0.54, 0.56)	(23.19, 37.02, 53.86)
3	(0.43, 0.48, 0.54)	(31.34, 81.58, 151.38)

Observing Table 8, one can see that the effort of the inner solver is always inferior to 1.5 matrix-vector products per iteration. Moreover, it is slightly larger for smaller problems (dimensions $n + 2q + s \in \{21, 31, 41\}$) than for larger ones ($n + 2q + s \in \{205, 305, 405\}$), although the dispersion between minimum and maximum values grows with increasing q. This last comment also applies to the global effort measure me_2, that grows as q increases, together with the length of intervals $[m_2, M_2]$. Although dimension differs by a factor of ten for the two sets of problems, figures of (m_2, me_2, M_2) are about twice as large when the two sets are compared.

Concerning Table 9, the main conclusions are that symmetry of matrices M and P does not seem to interfere in the performance of our approach, since families 1 and 2 produced quite similar results for both triples (m_3, me_3, M_3) and (m_4, me_4, M_4). The singularity of matrices M and P, on the other hand, showed significant effects, especially as far as the global performance is concerned.

This set of experiments consists of 54 tests. For the 27 problems of smaller dimension, the final objective function value was always inferior to 10^{-5}. Considering the 27 large ones, for 8 problems of the third family the final objective function values were greater than 10^{-2}, indicating convergence to a local nonglobal solution. This amounts to 55.6% success among problems for which the theoretical condition of equivalence does not hold. We stress, however, that whenever the hypothesis is valid, a global solution was reached.

4.4. Problems in three-dimensional cones with control of generated faces. In the fourth set of experiments we addressed the problem of finding $x \in \mathcal{K} = \{v \in \mathbb{R}^n \,|\, Av \geq 0\}$ such that $Tx + c \in \mathcal{K}^\circ = \{v \in \mathbb{R}^n \,|\, A^T\lambda = v, \lambda \geq 0\}$ and $x^T(Tx + c) = 0$. We generated the polyhedral cones $\mathcal{K}$ with q faces, such that their edges were the lines

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} r\cos\left(\frac{2\pi}{q}k\right) \\ r\sin\left(\frac{2\pi}{q}k\right) \\ 1 \end{pmatrix} t, \quad t \in \mathbb{R}, \quad k = 1, \ldots, q.$$

Therefore, $\mathcal{K}$ was defined by computing the rows of matrix A as the normal vectors to the support planes of the cone faces. In other words, the vector that defines the ith row of matrix A $(i = 1, \ldots, q)$ is given by the cross-product

$$\begin{pmatrix} \cos\left(\frac{2\pi}{q}(i-1)\right) \\ \sin\left(\frac{2\pi}{q}(i-1)\right) \\ \frac{1}{r} \end{pmatrix} \times \begin{pmatrix} r\cos\left(\frac{2\pi}{q}i\right) \\ r\sin\left(\frac{2\pi}{q}i\right) \\ 1 \end{pmatrix} = \begin{pmatrix} \sin\left(\frac{2\pi}{q}i\right)\left(\cos\frac{2\pi}{q} - 1\right) - \cos\left(\frac{2\pi}{q}i\right)\sin\frac{2\pi}{q} \\ \cos\left(\frac{2\pi}{q}i\right)\left(1 - \cos\frac{2\pi}{q}\right) - \sin\left(\frac{2\pi}{q}i\right)\sin\frac{2\pi}{q} \\ r\sin\frac{2\pi}{q} \end{pmatrix}.$$

The problems were generated as follows. Given the values of the radius r and of the dimension q (number of faces of cone $\mathcal{K}$), we built matrix A and created two

types of solutions x_*, at the boundary and in the interior of $\mathcal{K}$, respectively. Next we randomly generated matrix T, keeping it symmetric, and produced four families of problems, namely, (1) T indefinite, (2) T positive definite, (3) T positive semidefinite, and (4) T negative semidefinite. For more details, see [2].

The tests were produced by varying $r \in \{0.1, 1, 10\}$, $q \in \{3, 4, 5, 6, 9, 12\}$, the four families of matrices T, and the two kinds of generated solution x_*, which amounted to 144 problems. Three distinct seeds were chosen to generate problems for each selection of r, q, T, and x_*.

To analyze the robustness of the proposed approach, since half of the generated problems do not satisfy the hypothesis of the equivalence result (families 1 and 4, with matrices T indefinite and negative semidefinite, respectively), we observed that for the 72 problems with x_* generated at the boundary of the cone, 29 out of the 72×3 tests stopped at local nonglobal solutions. This corresponds to success for 86.6% of the total and 73.2% of the candidates for failure. For problems with x_* generated in the interior of the cone, six problems converged to local nonglobal solutions, in a total of 72×3 problems. In this case, the measures of success are 97.2% of the total and 94.4% of the problems without theoretical guarantee of convergence. Summing up the two blocks of tests, there were 35 failures, representing success in 91.9% of total and 83.8% of the universe of problems that do not satisfy the hypothesis of equivalence result.

There are some salient features that emerge from the results. First, the computational cost of the inner solver grows with the problem dimension, reaching its maximum for $q = 9$ and $q = 5$ if x_* is generated at the boundary and in the interior of $\mathcal{K}$, respectively.

It is also evident that the degree of difficulty of the generated problems grows as the radius r decreases: $r = 10$ produces the easiest problems whereas $r = 0.1$ generates the most difficult ones. Recall that in this set of experiments our problem is to find $x \in \mathcal{K} = \{v \in \mathbb{R}^n \mid Av \geq 0\}$ such that $Tx + q \in \mathcal{K}^\circ = \{v \in \mathbb{R}^n \mid A^T\lambda = v, \lambda \geq 0\}$, so the requirements for $\mathcal{K}$ and $\mathcal{K}^\circ$ are different.

Grouping problems according to the features of matrix T, there are 36 problems for each family (6 dimensions q, 3 values for r, and 2 types of generated x_*). We have computed the ratios $\texttt{INNER}/n_t$ and $\texttt{OUTER}/n_t$, where $n_t = n + 2q$ is the dimension of problem (2), and calculated average values, presented in Table 10, together with minimum and maximum values.

TABLE 10
Measures of effort per problem features.

Family	$\texttt{INNER}/n_t$			$\texttt{OUTER}/n_t$		
	Minimum	Average	Maximum	Minimum	Average	Maximum
1	6.3	15.6	51.1	0.3	0.6	1.4
2	1.6	12.2	33.4	0.2	0.6	1.2
3	3.9	13.6	35.8	0.3	0.6	1.1
4	1.7	16.4	153.3	0.1	0.6	2.1

Observing the figures of Table 10, we see that solving problems from families 1 and 4 (T indefinite and negative semidefinite, respectively) demands more effort than solving those from families 2 and 3 (T positive definite and positive semidefinite, respectively). The largest dispersion, that is, the largest interval (minimum, maximum), occurs for the fourth family, because of an outlier. Removing this discrepant value, the triples become $(1.7, 14.5, 46.2)$ and $(0.1, 0.7, 1.2)$, with dispersions similar to those of the first family.

5. Conclusions. We proposed a smooth box-constrained minimization reformulation of the GNCP$(F, G, \mathcal{K})$, assuming that $\mathcal{K}$ is a polyhedral cone. Any efficient minimization algorithm for solving this kind of problems may be used. The study of perturbed problems gives information about the solutions of a GNCP$(F, G, \mathcal{K})$ for a general cone $\mathcal{K}$ with very mild assumptions on the problem data.

Computational experiments are presented which encourage the use of our approach. Four groups of problems were addressed: randomly generated problems in the positive orthant, implicit complementarity problems from Outrata and Zowe, problems with general cones in $\mathbb{R}^n$, and problems in three-dimensional cones with control of generated faces.

The numerical results showed that the solution of the GNCP using (2) was found in the majority of the tests, even without accomplishment of theoretical hypothesis, meaning that the behavior of the method does not depend strongly on the sufficient conditions that guarantee the equivalence. Quantifying this robustness, considering only the universe of problems without theoretical support for convergence, for the first set of experiments the amount of failure was 24%. In the third and fourth sets, local nonglobal solutions were reached in 44% and 16% of the tests, respectively. No doubt, in the absence of theoretical support, the convergence to global solutions is more frequent for problems of smaller dimensions. The second set of problems, included for comparative purposes, formed by implicit complementarity problems, contained large-scale experiments (dimension up to $3 \times 4000 = 12000$) for which our approach had a very good performance. The third set of experiments revealed that general polyhedral cones might produce quite difficult problems, especially as the dimension increases. The fourth group of tests was created to investigate geometrical features of the cone $\mathcal{K}$. Besides noticing that, for the generated three-dimensional problems, thinner cones need more effort than wider ones, we observed that the increasing number of edges and faces did not substantially augment the amount of effort needed to solve the problems. As a natural extension of this work, we would like to investigate the possibility of approximating a general cone by a polyhedral one. This leads us to look for further connections between theory and practice concerning geometrical and algebraic properties of general cones and their relationship with GNCP defined in these sets. We are also interested in studying the behavior of our approach applied to problems with nonlinear functions F and G and polyhedral cones.

An important question that arises concerns whether limit points of the sequences generated by the minimization algorithm exist. The boundedness of the level sets of the merit function is a sufficient condition for the existence of these limit points, and results in this sense are given in [3, 17]. This matter deserves future research.

REFERENCES

[1] R. Andreani, A. Friedlander, and J. M. Martínez, *On the solution of finite-dimensional variational inequalities, using smooth optimization with simple bounds*, J. Optim. Theory Appl., 94 (1997), pp. 635–657.

[2] R. Andreani, A. Friedlander, and S. A. Santos, *Solving Generalized Nonlinear Complementarity Problems: Numerical Experiments on Polyhedral Cones*, Technical Report, IMECC, State University of Campinas, Campinas, Brazil, 2001; also available online from http://www.dcce.ibilce.unesp.br/∼andreani.

[3] R. Andreani and J. M. Martínez, *On the reformulation of nonlinear complementarity problems using the Fischer-Burmeister function*, Appl. Math. Lett., 12 (1999), pp. 7–12.

[4] R. Andreani and J. M. Martínez, *Reformulation of variational inequalities on a simplex and compactification of complementarity problems*, SIAM J. Optim., 10 (2000), pp. 878–895.

[5] R. Andreani and J. M. Martínez, *On the solution of bounded and unbounded mixed complementarity problems*, Optimization, 50 (2001), pp. 265–278.

[6] R. H. Bielschowsky, A. Friedlander, F. A. M. Gomes, J. M. Martínez, and M. Raydan, *An adaptive algorithm for bound constrained quadratic minimization*, Investigación Operativa, 7 (1997), pp. 67–102.

[7] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, *A limited memory algorithm for bound constrained optimization*, SIAM J. Sci. Comput., 16 (1995), pp. 1190–1208.

[8] A. R. Conn, N. I. M. Gould, and Ph. 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; see also SIAM J. Numer. Anal., 26 (1989), pp. 764–767.

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

[10] M. C. Ferris and C. Kanzow, *Complementarity and related problems*, in Handbook of Applied Optimization, P. M. Pardalos and M. G. C. Resende, eds., to appear.

[11] M. Fiedler and V. Pták, *Some generalizations of positive definiteness and monotonicity*, Numer. Math., 9 (1966), pp. 163–172.

[12] A. Friedlander, J. M. Martínez, and S. A. Santos, *A new trust region algorithm for bound constrained minimization*, Appl. Math. Optim., 30 (1994), pp. 235–266.

[13] A. Friedlander, J. M. Martínez, and S. A. Santos, *On the resolution of linearly constrained convex minimization problems*, SIAM J. Optim., 4 (1994), pp. 331–339.

[14] A. Friedlander, J. M. Martínez, and S. A. Santos, *Solution of linear complementarity problems using minimization with simple bounds*, J. Global Optim., 6 (1995), pp. 1–15.

[15] A. Friedlander, J. M. Martínez, and S. A. Santos, *A new strategy for solving variational inequalities on bounded polytopes*, Numer. Funct. Anal. Optim., 6 (1995), pp. 653–668.

[16] H. Jiang, M. Fukushima, L. Qi, and D. Sun, *A trust region method for solving generalized complementarity problems*, SIAM J. Optim., 8 (1998), pp. 140–157.

[17] C. Kanzow and M. Fukushima, *Equivalence of the generalized complementarity problem to differentiable unconstrained minimization*, J. Optim. Theory Appl., 90 (1996), pp. 581–603.

[18] O. L. Mangasarian and M. V. Solodov, *Nonlinear complementarity problem as unconstrained and constrained minimization*, Math. Program., 62 (1993), pp. 277–297.

[19] J. V. Outrata and J. Zowe, *A Newton method for a class of quasi-variational inequalities*, Comput. Optim. Appl., 4 (1995), pp. 5–21.

[20] J.-S. Pang, *The implicit complementarity problem*, in Nonlinear Programming 4, O. Mangasarian, M. Robinson, and R. Meyer, eds., Academic Press, New York, 1981, pp. 487–518.

[21] J.-S. Pang, *Complementarity problem*, in Handbook of Global Optimization, R. Horst and P. Pardalos, eds., Kluwer Academic Publishers, Boston, 1995, pp. 271–338.

[22] J. M. Peng, *Equivalence of variational inequality problems to unconstrained optimization*, Math. Programming, 78 (1997), pp. 347–355.

[23] R. B. Schnabel and T.-T. Chow, *Tensor methods for unconstrained optimization using second derivatives*, SIAM J. Optim., 1 (1991), pp. 293–315.

[24] P. Tseng, N. Yamashita, and M. Fukushima, *Equivalence of complementarity problems to differentiable minimization: A unified approach*, SIAM J. Optim., 6 (1996), pp. 446–460.

[25] N. Yamashita, K. Taji, and M. Fukushima, *Unconstrained optimization reformulations of variational inequality problems*, J. Optim. Theor. Appl., 92 (1997), pp. 439–456.

[26] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, *L-BFGS-B Fortran Subroutines for Large-Scale Bound Constrained Optimization*, Technical report, Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL, 1995.

SIAM J. OPTIM.
Vol. 12, No. 2, pp. 322–344

MEASURING THE GOODNESS OF ORTHOGONAL ARRAY DISCRETIZATIONS FOR STOCHASTIC PROGRAMMING AND STOCHASTIC DYNAMIC PROGRAMMING*

VICTORIA C. P. CHEN†

Abstract. This paper describes a continuous space discretization scheme based on statistical experimental designs generated from orthogonal arrays (OAs) of strength three with index unity. Chen, Ruppert, and Shoemaker [*Oper. Res.*, 47 (1999), pp. 38–53] employed this efficient discretization scheme in a numerical solution method for high-dimensional continuous-state stochastic dynamic programming (SDP). These OAs may be instrumental in reducing the dimensionality of event spaces, SDP state spaces, and first-stage decision spaces in two-stage stochastic programming. In particular, computationally efficient space-filling measures for these OAs are derived for evaluating how well a specific OA discretization fills the state space. Comparisons were made with two types of common measures: ones which maximize the average (or minimum) distance between discretization points within the OA and ones which minimize the average (or maximum) distance between discretization points and nondiscretization points lying on a full grid (i.e., points lying on the full grid that are not contained in the OA discretization). OAs of strength three were tested by fitting multivariate adaptive regression splines to data from an inventory-forecasting continuous-state stochastic dynamic program.

Key words. continuous state space, event space, space-filling design, finite projective geometry

AMS subject classifications. 05B15, 05B25, 60J25, 62K05, 62K99, 90C39, 90C15

PII. S1052623498332403

1. Introduction.

Stochastic programming (SP) and stochastic dynamic programming (SDP) have been used to solve problems in manufacturing systems, environmental engineering, revenue management, and many other fields (see King (1988), Birge and Louveaux (1997); White (1988), Puterman (1994)). Neither SP nor SDP is new (e.g., Dantzig (1955), Bellman (1957), Nemhauser (1966)), but they can both require computationally demanding solutions. For both, the objective is to minimize expected "cost," which represents any measure that one would like to minimize. Equivalently, one could maximize "benefit." SDP minimizes the cost to operate a stochastic system over several time periods by controlling *decision* variables in each time period. The *state* variables represent the current state of the system (e.g., at the beginning of a specific time period). Similarly, SP models decisions in discrete and ordered *stages*, where decisions in subsequent stages depend on decisions in prior stages. SDP is restricted to stochastic systems whose states can be modeled by a Markov decision process. SP is typically practical for only a small number of stages (most commonly two).

In both SP and SDP, computational complications arise when the random vector representing the stochastic nature of the system has a continuous distribution, i.e., a continuous event space. Both require a finite sample of *scenarios* to represent this distribution. The sample of scenarios is not only used to estimate the expected cost, but is needed to demonstrate "almost sure" feasibility of solutions.

*Received by the editors July 1, 1998; accepted for publication (in revised form) April 4, 2001; published electronically November 7, 2001.
http://www.siam.org/journals/siopt/12-2/33240.html
†Department of Information Systems, University of Texas at Arlington, Campus Box 19437, Arlington, TX 76019-0437 (vchen@exchange.uta.edu).

More interesting computational issues arise in the context of (nearly) continuous decision variables in the first (or prior) SP stages and (nearly) continuous SDP state variables. In a two-stage SP formulation, the second-stage *value function* is the optimal expected cost in the second stage, subject to constraints that are directly dependent on the decision in the first stage. Thus, when the first-stage decision variables are continuous, the second-stage value function cannot be calculated for every possible first-stage decision. In practice, even if the first-stage decisions are discrete, there will be too many to calculate all possibilities. Instead, iterative approximation methods which exploit the structure of SP problems are employed (see Birge and Louveaux (1997)).

In SDP, the *future value function* provides the minimum (cumulative) expected cost through the end of the (finite) time horizon. Continuous-state SDP assumes that the state variables are continuous. In deterministic form, large continuous-state dynamic programs may be efficiently solved using differential dynamic programming (see Caffey, Liao, and Shoemaker (1993)). Solutions to large continuous-state SDP models are much more difficult due to the stochasticity. However, the orthogonal array/multivariate adaptive regression splines (OA/MARS) method introduced by Chen, Ruppert, and Shoemaker (1999) shows great promise. In SDP with continuous state spaces, an approximate solution is found by discretizing the state space to a finite set of points. In Chen, Ruppert, and Shoemaker (1999), a discretization based on an OA was instrumental in reducing the dimensionality of high-dimensional continuous-state SDP.

The OA discretization is useful for representing any continuous space and is commonly used to design experiments for efficient statistical analysis. For SP and SDP, OA discretizations have the potential to significantly affect computational effort. This paper provides details on the construction of the OAs used by Chen, Ruppert, and Shoemaker (1999) and describes measures for assessing the "goodness" of these OA discretizations from a space-filling perspective. In particular, new computationally efficient space-filling measures for these OAs are introduced. These measures were used to differentiate the space-filling quality of OAs in a study by Chen (1999).

The next section briefly describes a two-stage SP formulation and a continuous-state SDP problem, highlighting the discretization issues. Section 3 presents discretization from a statistical perspective. Section 4.1 describes the construction of the OA designs employed by Chen, Ruppert, and Shoemaker (1999), followed by the derivation of properties in section 4.2 that lead to new measures of goodness in section 5.2. Section 5.1 introduces some common measures for space-filling designs, section 5.3 derives the computational requirements of all the measures, and section 5.4 presents correlation results between all the measures for OAs of strength three with four, six, and nine continuous variables. Finally, section 5.5 illustrates the difference in accuracy between "good" and "poor" state space discretizations on data from the last period of four-, six-, and nine-dimensional inventory-forecasting continuous-state SDP problems (see Chen (1999)).

2. Motivation for discretization.

2.1. A two-stage SP formulation. The basic two-stage stochastic linear program solves the following problem:

$$\begin{array}{ll} \min\limits_{\boldsymbol{x}} & \boldsymbol{c}^T\boldsymbol{x} + E\left\{\min\limits_{\boldsymbol{y}} \boldsymbol{g}^T\boldsymbol{y}\right\} \\ \text{s.t.} & A\boldsymbol{x} = \boldsymbol{b}, \\ & T\boldsymbol{x} + W\boldsymbol{y} = \boldsymbol{h}, \\ & \boldsymbol{x} \geq 0, \boldsymbol{y} \geq 0, \end{array}$$

where $\boldsymbol{x} \in R^{n_1}$ is the first-stage decision vector with linear costs $\boldsymbol{c} \in R^{n_1}$, $\boldsymbol{y} \in R^{n_2}$ is the second-stage decision vector with linear costs $\boldsymbol{g} \in R^{n_2}$, A is the $m_1 \times n_1$ first-stage linear constraint matrix with right-hand-side $\boldsymbol{b} \in R^{m_1}$, and T and W are, respectively, $m_2 \times n_1$ and $m_2 \times n_2$ matrices specifying the second-stage linear constraints on $\boldsymbol{x}$ and $\boldsymbol{y}$ with right-hand-side $\boldsymbol{h} \in R^{m_2}$. The expectation is taken over stochastic variables that may appear in $\boldsymbol{g}$, T, W, or $\boldsymbol{h}$. For a given realization of the stochastic variables, call it $\boldsymbol{\omega}$, we can write the second-stage value function as

$$Q(\boldsymbol{x}, \boldsymbol{\omega}) = \min_{\boldsymbol{y}} \left\{\boldsymbol{g}(\boldsymbol{\omega})^T\boldsymbol{y} \mid W(\boldsymbol{\omega})\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{\omega}) - T(\boldsymbol{\omega})\boldsymbol{x}, \boldsymbol{y} \geq 0\right\}. \tag{2.1}$$

Here we can see that the second-stage decision depends directly on the first-stage decision. Then the expected second-stage value function is

$$Q(\boldsymbol{x}) = E[Q(\boldsymbol{x}, \boldsymbol{\omega})],$$

where the expectation is taken over scenario realizations $\boldsymbol{\omega}$. Finally, the first-stage decision is found by solving the deterministic linear program:

$$\begin{array}{ll} \min\limits_{\boldsymbol{x}} & \boldsymbol{c}^T\boldsymbol{x} + Q(\boldsymbol{x}) \\ \text{s.t.} & A\boldsymbol{x} = \boldsymbol{g}, \\ & \boldsymbol{x} \geq 0. \end{array} \tag{2.2}$$

The difficulty lies in determining $Q(\boldsymbol{x})$. If the stochastic variables are continuous, then a large number of scenarios may be needed to estimate the expectation. The second-stage value function in (2.1) must be solved individually for each scenario $\boldsymbol{\omega}$. Thus there is a high computational cost for "evaluating" $Q(\boldsymbol{x})$ at just one $\boldsymbol{x}$. In solving the minimization in (2.2), each evaluation of $Q(\cdot)$ is computationally expensive. Consequently, the iterative approximation methods described by Birge and Louveaux (1997) can be very slow to converge.

Discretization may be useful in two ways for SP: (i) for generating the set of scenarios representing the continuous event space of the stochastic variables, and (ii) for constructing an approximation of $Q(\cdot)$ over the $\boldsymbol{x}$-space. The first item is a common use of discretization in SP. However, the second item is yet to be explored. To control the computational requirements of a solution method, we would need to control the number of times we evaluate $Q(\cdot)$. If we discretize the $\boldsymbol{x}$-space to a finite set of points and solve for $Q(\cdot)$ only at those points, then we can employ a function approximation technique to estimate the entire surface of $Q(\cdot)$. This approximation, call it $\hat{Q}(\cdot)$, will be computationally trivial to evaluate in the minimization of (2.2). In selecting a discretization, it will be important to select only those $\boldsymbol{x}$ vectors that result in a feasible solution in the second stage. The state space discretization problem for SDP, discussed next, is more straightforward.

2.2. Solving continuous-state SDP. The objective of SDP is to minimize expected costs over T time periods, i.e., to solve

$$\min_{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_T} E\left\{\sum_{t=1}^{T} c_t(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\epsilon}_t)\right\}$$
$$\text{s.t. } \boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\epsilon}_t) \quad \text{for } t = 1, \ldots, T-1, \text{ and}$$
$$(\boldsymbol{x}_t, \boldsymbol{u}_t) \in \Gamma_t \quad \text{for } t = 1, \ldots, T,$$

where $\boldsymbol{x}_t \in R^n$ is the state vector, $\boldsymbol{u}_t \in R^m$ is the decision vector, $c_t : R^{n+m+l} \to R^1$ is a known cost function for period t, $\Gamma_t \subset R^{n+m}$ is the set of constraints on $\boldsymbol{u}_t$ which depend on $\boldsymbol{x}_t$, and the expectation is taken over the random vector $\boldsymbol{\epsilon}_t \in R^l$, with known probability distribution. The known function f defines the transition from $\boldsymbol{x}_t$ to $\boldsymbol{x}_{t+1}$ by $\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\epsilon}_t)$. The *future value function* at time t is

$$F_t(\boldsymbol{x}_t) = \min_{\boldsymbol{u}_t,\ldots,\boldsymbol{u}_T} E\left\{\sum_{\tau=t}^{T} c_\tau(\boldsymbol{x}_\tau, \boldsymbol{u}_\tau, \boldsymbol{\epsilon}_\tau)\right\}$$
$$\text{s.t. } \boldsymbol{x}_{\tau+1} = f(\boldsymbol{x}_\tau, \boldsymbol{u}_\tau, \boldsymbol{\epsilon}_\tau) \quad \text{for } \tau = t, \ldots, T-1 \text{ and}$$
$$(\boldsymbol{x}_\tau, \boldsymbol{u}_\tau) \in \Gamma_\tau \quad \text{for } \tau = t, \ldots, T-1,$$

for $t = 1, \ldots, T$, and can be written recursively as

$$F_t(\boldsymbol{x}_t) = \min_{\boldsymbol{u}_t} E\{c_t(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\epsilon}_t) + F_{t+1}(\boldsymbol{x}_{t+1})\}, \quad t = 1, \ldots, T, \tag{2.3}$$

where we define $F_{T+1} \equiv 0$. It is assumed that the expected value can be calculated exactly (i.e., that the stochastic variable has a finite number of realizations). If the event space is continuous, then a set of scenarios, as discussed for SP, may be used to estimate the expected value.

Since there are assumed to be no costs incurred after the last period, the basic SDP solution algorithm solves backwards in time. In theory, the SDP solution is found by exhaustively solving, for each period, the minimization in (2.3) for every possible $\boldsymbol{x}_t$ state. This would result in a complete description of the future value function for each period (given the future value functions, we can re-solve for the optimal decisions). Of course, in continuous-state SDP this exhaustive solution method is impossible, since there are an infinite number of states. Instead, an approximate solution is found by discretizing the n-dimensional state space to a finite set of points, then using a function approximation technique to estimate the future value function over the continuous state space.

The most popular technique for discretizing a continuous SDP state space is to form a finite grid of discretization points in the state space. This approach is subject to exponential growth in the number of discretization points as the number of state variables increases. From a statistical design of experiments perspective, the discretization of the n-dimensional state space is analogous to constructing an experimental design for n predictor variables. This is discussed in the next section.

3. Design of experiments. The task of finding a good discretization parallels the task of finding a good statistical experimental design. The primary purpose of the field of experimental design is to create designs such that the relationship between a univariate *response* from an experiment and several *predictor variables* may be accurately modeled with an efficient number of design points. When a specific parametric model can be assumed for this relationship, then a design may be chosen

which satisfies special criteria, such as minimizing maximum variance, known as G-optimality in the statistical literature (Johnson, Moore, and Ylvisaker (1990)). When a model cannot be assumed, then a space-filling design, which spreads design points evenly over the region of interest, may be employed. A space-filling discretization would attempt to fill the continuous state, event, or decision space uniformly.

Consider an experiment with n predictor variables, $\boldsymbol{x} = (x_1, \ldots, x_n)$. In particular, these correspond to the n state variables in the SDP model. Alternatively, these could correspond to n stochastic variables over which an expectation will be calculated or, in SP, n first-stage decision variables. Assume each predictor variable can be set to q different levels. One *trial* of an actual experiment entails setting the predictor variables to specific levels and measuring the response produced by this combination of levels. For a particular trial, the setting of levels of the predictor variables corresponds to a discretization point in the $\boldsymbol{x}$-space. The set of trials for an experiment constitutes an *experimental design*, which corresponds to a discretization of the $\boldsymbol{x}$-space. Let N be the total number of trials in an experimental design (i.e., the number of discretization points). The N trials correspond to the rows of the $N \times n$ experimental *design matrix* D.

A full *factorial* design for an experiment with n predictors, each at q levels, contains all possible trials (combinations of levels of the predictor variables) and corresponds to the grid of $N = q^n$ discretization points in the n-dimensional continuous space, where each dimension represents a predictor variable and each variable takes on q levels. Clearly, N increases exponentially with n. A *fractional* factorial design is any proper subset of a full factorial design.

The *main effect* due to one predictor variable consists of the effect of that predictor on the response, averaged over the values of the other predictors. An *interaction* exists between two or more predictor variables when the combined effect of these predictors on the response is different from the sum of their main effects. A full factorial design permits estimation of all possible interactions between any number of variables. In practice, most relationships can be approximated using only main effects and low-order interactions. By assuming that all high-order interactions (e.g., all interactions involving four or more predictor variables) are negligible, fractional factorial designs or other smaller designs may be utilized. OA designs are special fractional factorial designs.

In addition to the usual fractional factorial designs (see Montgomery (1997)), there is a growing literature on experimental designs for meta-modeling. In meta-modeling, the goal is to build a response surface model based on a deterministic simulation of a complex system, so as to facilitate optimization of the system. This is very similar to (but not quite the same as) use of the discretization problems for approximating the value functions of SP and SDP. In a recent meta-modeling study using a flexible "kriging" model (see Sacks et al. (1989)), Palmer (1998) evaluated several types of experimental designs, including Latin hypercube sampling (McKay, Conover, and Bechman (1979)), OA-based Latin hypercube sampling (Owen (1992), Tang (1993)), (t, m, s)-based b nets (Owen (1995)), the Hammersley sequence (Kalagnanam and Diwekar (1997)), and a minimum-bias Latin hypercube design (Palmer and Tsui (2001)). Although the results were somewhat mixed, those that performed best overall in estimating one seven-dimensional and three four-dimensional chemical process systems were minimum-bias Latin hypercubes, OA-based designs, and Latin hypercube sampling. It should be noted that Latin hypercubes are a special case of OAs. In this paper, optimal OA designs (such as minimum-bias) are not

specifically generated because finding them is computationally grueling. However, assessing the "goodness" of our OA design in a computationally efficient manner is the focus of sections 5.1 through 5.3.

For discretization of the SDP state space, Chen, Ruppert, and Shoemaker (1999) selected OA designs because of their inherent relationship with factorial designs and their ease of construction. An OA design of *strength* d for n predictor variables and q levels is a fractional factorial design with the property that for any subset of d predictors, all possible factor-level combinations appear with the same frequency (λ). OA designs are *balanced* when looking at the variables d at a time. Spatially, when the points of an OA design of strength d in n dimensions are projected onto any d-dimensional subspace, each point of the d-dimensional full factorial with q levels in each dimension will be represented λ times. The notation for this array is $\mathrm{OA}(N, n, q, d)$ with *index* λ, where n, q, d, and λ are as above and $N = \lambda q^d$ is the total number of points in the design. An array $\mathrm{OA}(N = q^k, n, q, d)$ is identical to a hypercube denoted by (n, q, k, d). OAs of strength two are equivalent to hyper-Græco-Latin squares, and OAs of strength one with index unity are equivalent to Latin hypercubes. In order to minimize confusion, we will use only the OA notation.

The discretization used by Chen, Ruppert, and Shoemaker (1999) was chosen according to OA designs of strength three ($d = 3$) with index unity ($\lambda = 1$). Thus the total number of discretization points is $N = q^3$. For this OA design, the number of predictor variables must satisfy (see Bush (1952))

$$n \leq \begin{cases} q+1 & \text{when } q \text{ is odd,} \\ q+2 & \text{when } q \text{ is even.} \end{cases}$$

Thus q must increase linearly with n, implying that N increases polynomially with n, compared to exponentially for the full factorial design. MATLAB functions which generate the strength three OA designs when q is prime, and instructions on their use, may be obtained from the author.

4. OA discretization.

4.1. Construction. Following the notation used in the previous section, Chen's group (1999) employed an OA of strength three with index unity, which is denoted as $\mathrm{OA}(N = q^3, n, q, 3)$. Suppose that the number of levels $q = p^k$, where p is prime and k is a positive integer. Then Bose and Bush (1952) proved we can construct an array $\mathrm{OA}(q^r, n, q, d)$, where $r \geq d$ is an integer, if we can find an $n \times r$ matrix C, whose elements belong to the p^k-element Galois field $\mathrm{GF}(p^k)$ such that every $d \times r$ submatrix has rank d. They state that this is equivalent to finding n points in a $(r-1)$-dimensional finite projective geometry based on the p^k-element Galois field such that no d are *conjoint*. A set of d points is said to be *conjoint* if they all lie on a flat space with dimension $d-2$ or lower. In Chen, Ruppert, and Shoemaker (1999), $r = 3$ and $d = 3$, so a strength three OA discretization may be generated by finding n points in the projective plane such that no three points lie in a line (i.e., *conjoint* translates to collinear in the projective plane). A brief background on the projective plane is provided in Appendix A.

Although the theory for generating these OAs appears in the literature, an actual algorithm does not. In this section we present our algorithm. As stated above, our goal is to find n points such that no three points are collinear. Batten (1986) shows that any line in the projective plane intersects a nondegenerate *point conic* (described in Appendix B and briefly illustrated in the next paragraph) in either 0, 1, or 2 points.

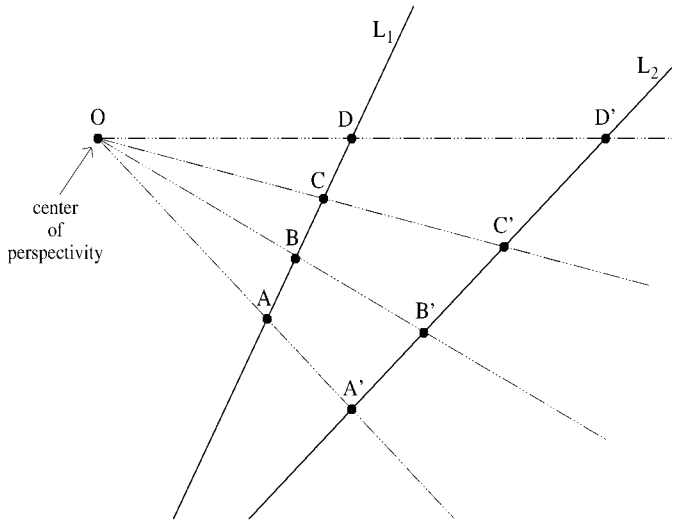


FIG. 4.1. *This central perspectivity transforms the points on line* L_1 *onto line* L_2 *such that the corresponding pairs of points on* L_1 *and* L_2 *are collinear with the point* O. *In particular, we see that the four points* A, B, C, *and* D *on* L_1 *become the points* A', B', C', *and* D' *on* L_2, *respectively.*

In other words, no *three* points of a nondegenerate point conic are collinear. Thus we can achieve our goal by constructing a nondegenerate point conic. For strength $d = 3$ and $n = q+1$ points, and specifying $r = 3$, we can generate the array $\mathrm{OA}(q^3, q+1, q, 3)$ from the coordinates of the $q + 1$ points on a nondegenerate point conic of the finite projective plane $\mathrm{PG}(2, q)$. The maximum allowable number of predictor variables for a strength three OA with q odd (e.g., prime greater than two) is $q + 1$ (Bush (1952)). A similar approach in projective space may be used to generate strength four OAs; however, strength four OAs are generally too large to be practical.

To follow the algorithm, the reader must have, at minimum, a graphical understanding of a point conic. Figures 4.1 and 4.2 illustrate two basic concepts, *perspectivity* and *projectivity*, leading up to the point conic illustrated in Figure 4.3. As shown in Figure 4.1, a *central perspectivity* transforms the points of one line onto another line via a point called the *center of perspectivity*. A product of perspectivities is called a *projectivity*. The projectivity illustrated in Figure 4.2 is the product of two central perspectivities.

Figure 4.3 attempts to illustrate a point conic. The construction of a point conic begins with two points, say P and Q. In the finite projective plane $\mathrm{PG}(2, q)$, a point P has exactly $q+1$ lines going through it. This set of lines is called the *pencil* through point P. The point conic is formed by a *projectivity* of the lines in the pencil through P onto the lines in the pencil through Q. In Figure 4.3, the lines are labeled such that line u_i in the pencil through P is projected onto line v_i in the pencil through Q. The dots are the points in the resulting point conic. Given the same two pencils, a different projectivity would result in a different point conic.

For the algorithm, we adopt the notation P for points, $\boldsymbol{u}$ and $\boldsymbol{v}$ for lines, and use subscripts $\{1, 2\}$ to denote the two pencils. The algorithm for generating an $\mathrm{OA}(q^3, n, q, 3)$ design from a nondegenerate point conic is given below. In step 2(a), the selection of two lines from each pencil defines the specific projectivity that will be used to generate the point conic. In step 2(c), the rows of the matrix C correspond

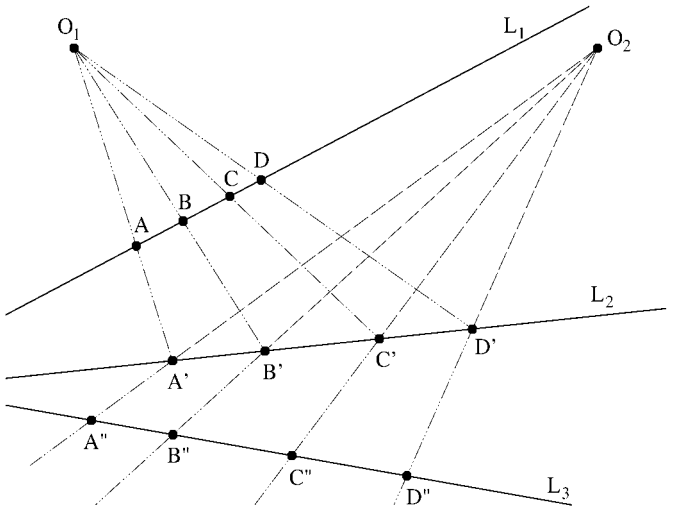


FIG. 4.2. *This projectivity transforms the points on line* L_1 *onto line* L_3 *and is a product of two central perspectivities. The first central perspectivity transforms the points on line* L_1 *onto line* L_2 *via the center of perspectivity* O_1. *The second transforms the points on line* L_2 *onto line* L_3 *via the center of perspectivity* O_2. *In particular, we see that the four points* A, B, C, D *on* L_1 *become the points* A″, B″, C″, D″ *on* L_3, *respectively.*

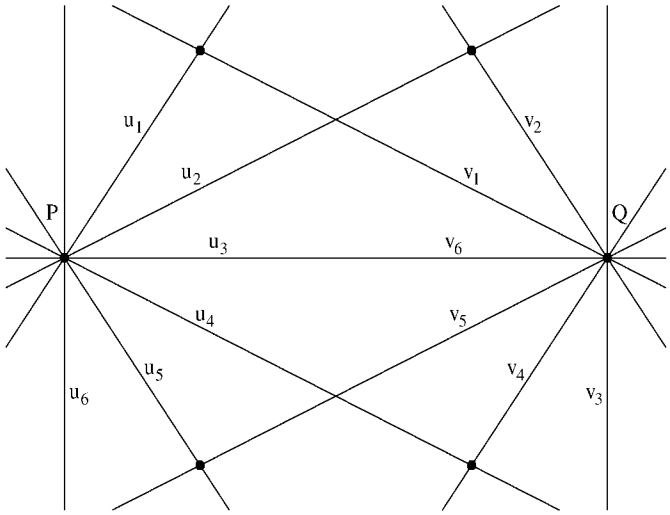


FIG. 4.3. *This point conic in* PG(2,5) *is the result of a projectivity of the lines through point* P *into lines through point* Q. *The lines* u_i *in the pencil through* P *are projected into the lines* v_i *in the pencil through* Q, *and the intersections of the pairs of lines* (u_i, v_i), $i = 1, \ldots, 6$, *form the points of the conic. This is a nondegenerate point conic since there are six lines in both pencils and six points in the point conic itself.*

to the points of the point conic.

ALGORITHM OA3.

1. Select two distinct points P_1 and P_2 from the projective plane.
2. Identify the points of the point conic generated by P_1 and P_2 as follows:

(a) Select two distinct lines, $\boldsymbol{u}_1 = [u_{11}, u_{12}, u_{13}]^T$ and $\boldsymbol{v}_1$, from the pencil through P_1 and any two distinct lines, $\boldsymbol{u}_2$ and $\boldsymbol{v}_2$, from the pencil through P_2;
(b) Calculate the coefficients

$$a_{ij} = a_{ji} = [(u_{1i}v_{2j} - u_{2j}v_{1i}) + (u_{1j}v_{2i} - u_{2i}v_{1j})]/2 \tag{4.1}$$

for the symmetric homogeneous second-degree equation

$$\sum_{i=1}^{3}\sum_{j=1}^{3} a_{ij}x_i x_j = \boldsymbol{x}^T A\boldsymbol{x} = 0; \tag{4.2}$$

(c) Construct the $(q+1)\times 3$ matrix C with rows corresponding to the points $\boldsymbol{x}$ that satisfy (4.2).

3. If C is not full rank, then the point conic is degenerate → return to step 1.
4. Otherwise the point conic is nondegenerate and the OA may be constructed as follows:
   (a) Construct the $q^3 \times 3$ matrix E with rows consisting of all q^3 possible triples with components belonging to GF(p^k);
   (b) Calculate $D = EC^T$, which is the $q^3 \times (q+1)$ experimental design matrix for the array OA$(q^3, q+1, q, 3)$. For $n < q+1$ variables, select n of the $q+1$ columns of the matrix D.

The rows of D are the points in the OA discretization. Details on the derivation of (4.1) and (4.2) are given in Appendix B. Permutations of the columns $1, 2, \ldots, n$ of the design matrix D and permutations of the elements $0, 1, \ldots, q-1$ within any column generate other OA$(q^3, n, q, 3)$ discretizations.

For illustration in the later sections of this paper, two OA$(5^3, 6, 5, 3)$ discretizations, two OA$(7^3, 8, 7, 3)$ discretizations, one OA$(11^3, 12, 11, 3)$ discretization, and one OA$(13^3, 14, 13, 3)$ discretization were generated in MATLAB. Denote the two OAs with $q = 5$ levels by OA$(5)_1$ and OA$(5)_2$, the two OAs with $q = 7$ by OA$(7)_1$ and OA$(7)_2$, and the $q = 11$ and $q = 13$ OAs by OA$(11)_1$ and OA$(13)_1$. All our OA$(q^3, n, q, 3)$ discretizations were constructed by choosing a subset of n columns from an OA$(q^3, q+1, q, 3)$ discretization. Details on creating the six discretizations above are given in Appendix C.

4.2. Properties. In this section we will focus on the properties of the OA$(N = q^d, n, q, d)$ discretizations generated by the Bose and Bush (1952) theorem used in section 4.1. This will lead to new space-filling measures specifically for these OAs. As described in section 3, these OAs have the special property that, spatially, when the n-dimensional points of the OA are projected onto any d-dimensional subspace, each point of the d-dimensional full factorial with q levels in each dimension will be represented once. Equivalently, in the $N \times n$ design matrix, where each column corresponds to a variable, two rows may coincide in at most $d-1$ columns. This property gives OA discretizations balance, but does not guarantee an evenly spaced discretization.

For OAs with index unity, we can consider the configurations of points on $(n-d+1)$-dimensional slices with $d-1$ variables fixed. For example, in a four-dimensional discretization, Figure 4.4 pictures slices with x_1 and x_4 fixed. Specifically, if we pull out all the points in our OA for which $x_1 = 3$ and $x_4 = 0$, that would result in exactly the $p = 7$ points plotted in the first square of Figure 4.4. The figure does

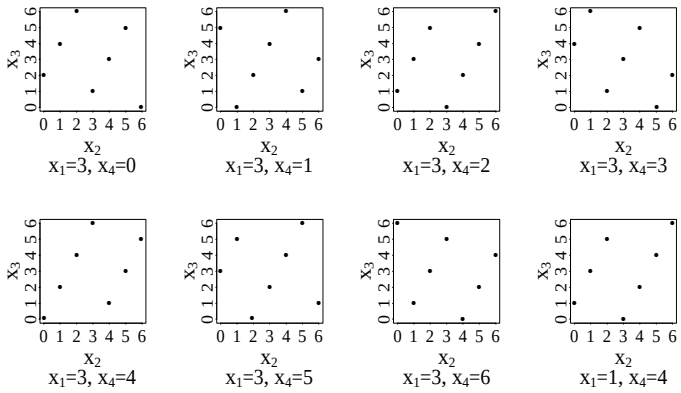


FIG. 4.4. *Eight configurations from the same q-family in the* $\mathrm{OA}(7^3, 4, 7, 3)$ *array which consists of the columns* 1, 4, 5, *and* 7 *from the array* $\mathrm{OA}(7)_1$. *The plot for one square was generated by pulling all points out of the* $\mathrm{OA}(7^3, 4, 7, 3)$ *array for which* x_1 *and* x_4 *were fixed at the designated levels, then plotting the resulting set of* $p = 7$ *points in the two-dimensional subspace of* x_2 *and* x_3. *This subspace is referred to as a "slice" of the array. Note that, for all the configurations, the points lie on lines with slope* 2.

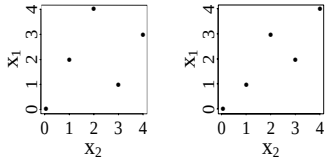


FIG. 4.5. *Two types of configurations. In the configuration on the left, the points lie on parallel lines with slope* 2 *(*x_2 *jumps* $z = 2$ *levels for each* x_1 *unit jump). In the configuration on the right,* x_2 *jumps in steps* $1, 2, 4, 2, 1$ *as* x_1 *jumps levels* 0–1–2–3–4–0. *The second type of configuration does not occur with our OAs.*

not show all possible slices with x_1 and x_4 fixed since this would vary x_1 and x_4 in all possible combinations, resulting in 49 squares. Looking at the spatial configurations pictured in these slices, we obtain graphical evidence of the space-filling "goodness" of the OA. If the seven points in the slice appear to be evenly spread over the square (as they basically do in Figure 4.4), then this indicates good space-filling quality. By contrast, if all seven points lie on one diagonal line, then this would clearly indicate poor space-filling quality. By considering the configurations of *all* possible slices, we can obtain a measure of the space-filling "goodness" of a particular OA discretization. The new space-filling measures presented in section 5.2 are based on this premise.

The reality of generating all possible slices would result in a prohibitively large computation. Instead, we present properties of our OAs that significantly reduce the number of slices we must consider and thus reduce the computational requirement for calculating the new measures (see section 5.3). The first property is illustrated in Figure 4.5. For OAs generated by the Bose and Bush (1952) theorem, only certain configurations are possible. For example, Figure 4.5 illustrates configurations from two $\mathrm{OA}(5^3, 4, 5, 3)$ discretizations. The first one has a constant step size in x_2 for each unit of x_1, and the second one has variable step sizes. This second type never occurs in the OAs generated by the Bose and Bush (1952) theorem. The second property is illustrated in Figure 4.6 and is proven below. This property states that for a particular set of $d - 1$ fixed variables (e.g., x_1 and x_4 fixed in Figure 4.4), the

resulting configurations will all have the same step size. Thus, for each possible set of $d-1$ fixed variables, only one configuration must be identified. For example, note that all the configurations in Figure 4.4 show x_3 with a step size of two for each unit of x_2.

In an OA discretization, if variables $(x_{i_1}, x_{i_2}, \ldots, x_{i_{d-1}})$ are fixed at specific values, then there will be exactly q rows of D that contain these variables at these values (due to the balance property of an OA with index unity). Call this set of rows a *q-set*. Since for any two rows $d-1$ is the maximum number of coinciding columns, each of the $n-d+1$ nonfixed columns in a q-set contains each of the q elements in GF(q) exactly once. If two q-sets have the same variables fixed, then they are in the same *q-family*. There are exactly q^{d-1} q-sets in a q-family, and $\binom{n}{d-1}$ types of q-families in D. Also note that the rows of the q-sets of one q-family comprise D.

THEOREM 4.1. *Let E be the $N \times d$ matrix of all possible d-tuples with elements belonging to* GF(q). *Then $D = EC^T$ is the corresponding experimental design matrix, where C satisfies the theorem from Bose and Bush* (1952). *Since* GF(q) *is closed under the operations addition, subtraction, multiplication, and division, assume all operations are taken modulo q. Consider two rows, D_1 and D_2, of D contained in a q-set with variables $(x_{i_1}, x_{i_2}, \ldots, x_{i_{d-1}})$ fixed at the values $(K_1, K_2, \ldots, K_{d-1})$. For any* nonfixed column v, *define $D_i(v)$ to be the value in the vth column of D_i, and $\Delta v = D_1(v) - D_2(v)$. Choose a nonfixed column u as a* reference column *and let $(v_1, v_2, \ldots, v_{n-d})$ denote the remaining $n-d$ nonfixed columns. Then for each v_i, $i = 1, 2, \ldots, n-d$, a* slope $z(v_i)$ *can be computed such that*

$$\Delta v_i = \Delta u \cdot z(v_i), \tag{4.3}$$

where $z(v_i)$ is identical for any two rows in the q-set.

Furthermore, the slopes $z(v_1), \ldots, z(v_{n-d})$ only depend on the choice of columns $(i_1, \ldots, i_{d-1}, u)$ and do not depend on the values of those columns.

Proof. Without loss of generality, consider only column v_1. Sort the rows of the q-set so that the uth column is $(0, 1, \ldots, p-1)^T$. Let C_v denote the row of C corresponding to the vth column of D, and let $\tilde{C}$ be the $d \times d$ submatrix of C corresponding to the columns $i_1, i_2, \ldots, i_{d-1}, u$. Let $\tilde{E}$ denote the $q \times d$ submatrix of E corresponding to the ordered q-set, and let $\tilde{E}_j$ denote the jth row of $\tilde{E}$.

Since the q-set is ordered according to the uth column, the difference between values of column u for any two points is simply the difference in the indices,

$$\tilde{E}_{j+\Delta u} C_u^T - \tilde{E}_j C_u^T = (\tilde{E}_{j+\Delta u} - \tilde{E}_j) C_u^T = \Delta u,$$

where j and Δu are both elements of GF(q). In matrix form, we have

$$\begin{aligned} (\tilde{E}_{j+\Delta u} - \tilde{E}_j)\tilde{C}^T &= (K_1 - K_1, K_2 - K_2, \ldots, K_{d-1} - K_{d-1}, \Delta u) \\ &= (0, 0, \ldots, 0, \Delta u). \end{aligned} \tag{4.4}$$

Since any $d \times d$ submatrix of C has full rank, $\tilde{C}$ is nonsingular. Thus we can solve for $(\tilde{E}_{j+\Delta u} - \tilde{E}_j)$ in (4.4) to get

$$\tilde{E}_{j+\Delta u} - \tilde{E}_j = (0, 0, \ldots, 0, \Delta u)\tilde{C}^{-T}. \tag{4.5}$$

For column v_1, we need to show

$$\Delta v = \tilde{E}_{j+\Delta u} C_{v_1}^T - \tilde{E}_j C_{v_1}^T = (\tilde{E}_{j+\Delta u} - \tilde{E}_j) C_{v_1}^T = \Delta u \cdot z(v_1) \tag{4.6}$$

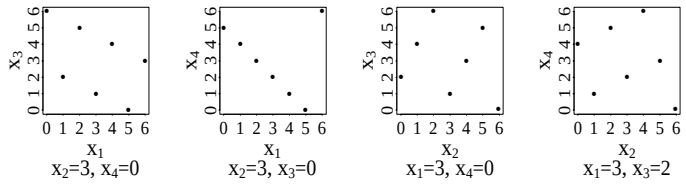


FIG. 4.6. *Four configurations from different q-families. The third one is from the family shown in Figure* 4.4. *The equivalent slope vectors* s *that have minimizing length* $\|\mathsf{s}\|$ *for these q families are (respectively):* $(2,-1)$, $(1,-1)$, $(1,2)$, *and* $(2,1)$. *Thus, the first, third, and fourth q-families have* $\delta_s^m = 2$, *while the second has* $\delta_s^m = 1$.

for any j in GF(q). Plugging (4.5) into (4.6), we have

$$\Delta v = (0,0,\ldots,0,\Delta u)\tilde{C}^{-T}C_{v_1}^T = \Delta u \cdot z(v_1),$$

where $z(v_1) = (0,0,\ldots,0,1)\tilde{C}^{-T}C_{v_1}^T$. □

The main consequence of Theorem 4.1 is that all q-sets in the same q-family have exactly the same slopes $z(v_1),\ldots,z(v_{n-d})$. Thus, it is sufficient to compute the slopes for each q-family. For an OA($7^3, 4, 7, 3$) discretization generated from the array OA$(7)_1$, Figure 4.4 illustrates configurations for the q-family with variables x_1 and x_4 fixed. Figure 4.6 illustrates configurations from four different q-families.

COROLLARY 4.2. *For the situation in Theorem* 4.1, *define the* slope vector *to be* $\boldsymbol{s} = (\Delta u, \Delta v_1, \ldots, \Delta v_{n-d})^T$. *Then the slope vector for a different choice of rows* D_1 *and* D_2 *is* equivalent *to* $\boldsymbol{s}$ *in the sense that it is a multiple (modulo q) of* $\boldsymbol{s}$.

Proof. From (4.3), we deduce that the slope vector for any two rows in a q-set is equivalent to the vector $(1, \Delta v_1/\Delta u, \ldots, \Delta v_{n-d}/\Delta u)^T$, which is equivalent to $\boldsymbol{s}$. □

As a consequence of Corollary 4.2, all q-sets in the same q-family have equivalent slope vectors. Since the rows of D correspond to the points in the discretization, the length of the slope vector for two rows translates to the distance between the corresponding two points. The minimum distance between two points in a q-set can be determined by choosing the equivalent slope vector $\boldsymbol{s}$ which minimizes the length $\|\boldsymbol{s}\|$. For example, in Figure 4.4, if we take x_2 as the reference column ($u = 2$) and x_3 as the remaining nonfixed column ($v = 3$), then the two topmost points in the first configuration produce the slope vector, modulo $q = 7$, ($\Delta u = 4, \Delta v = 1$). (Note: in GF($q$), $-3 = 4$.) The equivalent slope vectors are $(1,2), (5,3), (2,4), (6,5)$, and $(3,6)$. The minimum-length slope vector is $(1,2)$ with a Euclidean norm of $\sqrt{5}$. The minimum distance between two points in any of the configurations is easily seen to be $\sqrt{5}$. This leads to special space-filling measures for OAs generated by the Bose and Bush (1952) method.

5. Measuring goodness.

5.1. Space-filling measures. A good space-filling discretization should spread points evenly over the continuous space of interest. Define a *nondiscretization point* to be a point in the continuous space not contained in the discretization. Reasonable objectives include (1) minimizing the maximum distance from any nondiscretization point to the nearest discretization point and (2) maximizing the minimum distance from any discretization point to its nearest neighbor in the discretization. Related objectives include (1) minimizing the average distance from a nondiscretization point to the nearest discretization point and (2) maximizing the average distance from a discretization point to its nearest neighbor in the discretization. Let $d(x,y)$ be a

distance measure (commonly Euclidean distance) over the region of interest. Let X_D be the set of discretization points and X_N be the set of nondiscretization points. Define the measures

$$\delta_{\text{nd}}^M = \max_{x \in X_N} \left\{ \min_{y \in X_D} \; d(x,y) \right\}, \tag{5.1}$$

$$\delta_{\text{dd}}^m = \min_{x \in X_D} \left\{ \min_{\substack{y \in X_D \\ y \neq x}} \; d(x,y) \right\}, \tag{5.2}$$

$$\overline{\delta}_{\text{nd}} = \frac{1}{\mid X_N \mid} \sum_{x \in X_N} \left\{ \min_{y \in X_D} \; d(x,y) \right\}, \text{ and} \tag{5.3}$$

$$\overline{\delta}_{\text{dd}} = \frac{1}{\mid X_D \mid} \sum_{x \in X_D} \left\{ \min_{\substack{y \in X_D \\ y \neq x}} \; d(x,y) \right\}, \tag{5.4}$$

where $\mid X \mid$ is the cardinality of the set X. For the above to be computationally practical we must assume $\mid X_N \mid$ and $\mid X_D \mid$ are finite. Our objectives translate to minimizing δ_{nd}^M or $\overline{\delta}_{\text{nd}}$ (minimax) and maximizing δ_{dd}^m or $\overline{\delta}_{\text{dd}}$ (maximin).

5.2. A new measure. For an OA($N = q^d, n, q, d$) discretization satisfying the properties in the previous section, let S be the set of slope vectors corresponding to the $\binom{n}{d-1}$ q-families. For each slope vector in S, determine the equivalent slope vector $\boldsymbol{s}$ that minimizes the length, $\|\boldsymbol{s}\|$, and let S^* be the set of minimum-length slope vectors. This minimum length for a q-family is the minimum distance between points within a q-set from that family. Define the "slicing" measures

$$\delta_{\text{s}}^m = \min_{\boldsymbol{s} \in S^*} \{\|\boldsymbol{s}\|\} \quad \text{and} \tag{5.5}$$

$$\overline{\delta}_{\text{s}} = \frac{1}{\binom{n}{d-1}} \sum_{\boldsymbol{s} \in S^*} \{\|\boldsymbol{s}\|\}, \tag{5.6}$$

which represent the minimum and average length of the minimum-length slope vectors in S^*. Intuitively, our space-filling objective is to maximize both δ_{s}^m and $\overline{\delta}_{\text{s}}$. In Figure 4.6, the second configuration, which is the least desirable of the four, has $\|\boldsymbol{s}\| = 1$ while the other three have a minimum length of $\|\boldsymbol{s}\| = \sqrt{5}$. Thus, one main objective would be to avoid discretizations containing q-families with $\|\boldsymbol{s}\| = 1$.

5.3. Computational considerations. In this section, the computational times of the measures given in (5.1)–(5.6) are compared. The minimax measures δ_{nd}^M and $\overline{\delta}_{\text{nd}}$ compute distances for all pairs of nondiscretization and discretization points. The maximin measures δ_{dd}^m and $\overline{\delta}_{\text{dd}}$ compute distances for all pairs of discretization points. The slicing measures δ_{s}^m and $\overline{\delta}_{\text{s}}$ compute distances (lengths) for equivalent slopes corresponding to each q-family. As before, let n be the dimension of the continuous space of interest (e.g., the SDP state space), q be the number of levels for each variable, and d be the strength of the OA. The time required to compute each distance is $\text{O}(n)$. For OAs with index unity, the number of discretization points is q^d. Restricting all points to the full grid, the number of nondiscretization points is $q^n - q^d$. Finally, the number of q-families is $\binom{n}{d-1}$, each having q equivalent slope vectors. Thus we have

TABLE 5.1

Computational times (in hours, minutes, or seconds) on a Sun SPARCstation 10. Estimated times are given in parentheses. Notation: n is the number of state variables and q is the number of discretization levels in each dimension; an entry of "$\cdots$" means that computational demands were too high for times to be estimated.

n	q	δ_{nd}^M and $\overline{\delta}_{\text{nd}}$	δ_{dd}^m and $\overline{\delta}_{\text{dd}}$	δ_{s}^m and $\overline{\delta}_{\text{s}}$
4	5	0.29 s	0.07 s	< 0.001 s
	7	3.21 s	0.52 s	0.0014 s
	11	1.34 m	7.92 s	0.0022 s
	13	4.36 m	22.06 s	0.0033 s
6	7	3.74 m	0.73 s	0.0036 s
	11	3.95 h	11.24 s	0.0051 s
	13	(18 h)	31.34 s	0.0082 s
9	11	(7,850 h)	16.75 s	0.0141 s
	13	(58,250 h)	44.72 s	0.0213 s
15	17	$\cdots$	(6 m)	(0.1 s)
40	41	$\cdots$	(50 h)	3 s

the computational times:

$$C_{\text{nd}} = \text{O}\left(n\left[q^n - q^d\right]q^d\right) = \text{O}\left(nq^{nd}\right),$$

$$C_{\text{dd}} = \text{O}\left(n\binom{q^d}{2}\right) = \text{O}\left(nq^{2d}\right),$$

$$C_{\text{s}} = \text{O}\left(pn\binom{n}{d-1}\right) = \text{O}\left(pn^d\right).$$

Actual and estimated (based on C_{nd}, C_{dd}, C_{s}) computational times are shown in Table 5.1. Our new measures δ_{s}^m and $\overline{\delta}_{\text{s}}$ compute the fastest, with measures δ_{nd}^M and $\overline{\delta}_{\text{nd}}$ becoming impractical very quickly.

5.4. Correlation between measures. The three types of measures were compared on OA discretizations generated from the arrays $\text{OA}(5)_1$, $\text{OA}(5)_2$, $\text{OA}(7)_1$, $\text{OA}(7)_2$, $\text{OA}(11)_1$, and $\text{OA}(13)_1$. For a specific OA, the six computed measures δ_{nd}^M, $\overline{\delta}_{\text{nd}}$, δ_{dd}^m, $\overline{\delta}_{\text{dd}}$, δ_{s}^m, and $\overline{\delta}_{\text{s}}$ provide the space-filling quality of the OA. In conducting the computational study, it was discovered that different OAs would frequently have the exact same six computed measures. This indicated that the configurations of these OAs were spatially equivalent. Thus, it was only necessary to identify all the different sets of measures, which are referred to below as different "OA patterns."

First consider the case with $n = 4$ continuous variables. For $q = 5$ levels, four possible sets of measures were identified, i.e., $m = 4$ different OA patterns. For $q = 7$, 11, and 13, OA patterns were identified with $m = 11$, 8, and 22, respectively. Recall that the objectives are to minimize $(\overline{\delta}_{\text{nd}}, \delta_{\text{nd}}^M)$, maximize $(\overline{\delta}_{\text{dd}}, \delta_{\text{dd}}^m)$, and maximize $(\overline{\delta}_{\text{s}}, \delta_{\text{s}}^m)$. In Figure 5.1(a) one can see that the best $\overline{\delta}_{\text{s}}$ values correspond to the best $\overline{\delta}_{\text{nd}}$ values, and, in particular, the worst values for each q correspond exactly. Figure 5.1(b) indicates that $\overline{\delta}_{\text{dd}}$ cannot distinguish between OAs with $q = 5$ and $q = 7$, and can only distinguish the top values for $q = 11$ and $q = 13$. Correlations between our new slicing measures and the others are shown in Table 5.2(a). Note that δ_{s}^m is perfectly correlated with δ_{dd}^m and is perfectly correlated with $\overline{\delta}_{\text{dd}}$ when $q = 11$ and $q = 13$.

Next consider the case with $n = 6$ continuous variables. For $q = 7$, 11, and 13, OA patterns were identified with $m = 14$, 50, and 581, respectively. Correlations are

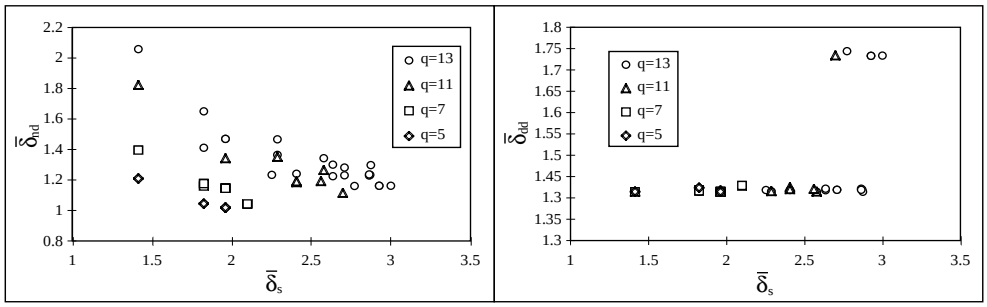


FIG. 5.1. *Measures for four-dimensional* OA *discretizations consisting of columns from the arrays* $\mathrm{OA}(5)_1$, $\mathrm{OA}(5)_2$, $\mathrm{OA}(7)_1$, $\mathrm{OA}(7)_2$, $\mathrm{OA}(11)_1$, *and* $\mathrm{OA}(13)_1$. *In plot* (a), *the average minimum-length slope,* $\overline{\delta}_s$, *is negatively correlated with the average distance from a nondiscretization point to the nearest discretization point,* $\overline{\delta}_{nd}$. *In plot* (b), *the average distance from a discretization point to its nearest neighbor in the discretization,* $\overline{\delta}_{dd}$, *distinguishes only a few discretizations with high* $\overline{\delta}_s$.

TABLE 5.2

Correlations between $(\overline{\delta}_s, \delta_s^m)$ *and* $(\overline{\delta}_{nd}, \delta_{nd}^M, \overline{\delta}_{dd}, \delta_{dd}^m)$. *Notation:* n *is the number of state variables,* q *is the number of discretization levels in each dimension, and* m *is the number of different OA patterns; N/A indicates data not available.*

	n	q	m		$\overline{\delta}_{\mathrm{nd}}$	δ_{nd}^M	$\overline{\delta}_{\mathrm{dd}}$	δ_{dd}^m
(a)	4	5	4	$\overline{\delta}_{\mathrm{s}}$	−0.99	−0.97	0.22	N/A
				δ_{s}^m	N/A	N/A	N/A	1.00
		7	11	$\overline{\delta}_{\mathrm{s}}$	−0.98	−0.83	0.64	N/A
				δ_{s}^m	N/A	N/A	N/A	1.00
		11	8	$\overline{\delta}_{\mathrm{s}}$	−0.94	−0.91	0.41	0.40
				δ_{s}^m	−0.35	−0.35	1.00	1.00
		13	22	$\overline{\delta}_{\mathrm{s}}$	−0.86	−0.84	0.47	0.46
				δ_{s}^m	−0.41	−0.38	1.00	1.00
(b)	6	7	14	$\overline{\delta}_{\mathrm{s}}$	−0.73	−0.10	0.23	0.14
				δ_{s}^m	−0.33	−0.36	0.91	0.96
		11	50	$\overline{\delta}_{\mathrm{s}}$	−0.88	−0.81	0.51	0.36
				δ_{s}^m	−0.75	−0.64	0.87	0.70
		13	581	$\overline{\delta}_{\mathrm{s}}$	N/A	N/A	0.32	0.20
				δ_{s}^m	N/A	N/A	0.82	0.73
(c)	9	11	12	$\overline{\delta}_{\mathrm{s}}$	N/A	N/A	0.84	N/A
				δ_{s}^m	N/A	N/A	0.98	N/A
		13	475	$\overline{\delta}_{\mathrm{s}}$	N/A	N/A	0.14	−0.13
				δ_{s}^m	N/A	N/A	0.93	0.92

shown in Table 5.2(b). The measures δ_{nd}^M and $\overline{\delta}_{\mathrm{nd}}$ were not computed for $q = 13$ due to excessive computational time. Figure 5.2(a) illustrates strong decreasing linear relationships between $\overline{\delta}_{\mathrm{s}}$ and $\overline{\delta}_{\mathrm{nd}}$ for $q = 7$ and $q = 11$. Both Figures 5.1(a) and 5.2(a) imply that we can use $\overline{\delta}_{\mathrm{s}}$ when $\overline{\delta}_{\mathrm{nd}}$ cannot be computed. In Figure 5.2(b), the values of $\overline{\delta}_{\mathrm{dd}}$ separate into three regions. The worst δ_{s}^m values correspond to the worst $\overline{\delta}_{\mathrm{dd}}$ values for $q = 7$ and $q = 11$, but δ_{s}^m does not distinguish between the two other regions. However, the middle level completely consists of $q = 13$ points, except for one $q = 7$ point, and the top level predominantly consists of $q = 11$ points and the "good" $q = 7$ points. Within each region, one can see a slightly increasing linear relationship, indicating that δ_{s}^m is able to distinguish the "good" $\overline{\delta}_{\mathrm{dd}}$ values from the

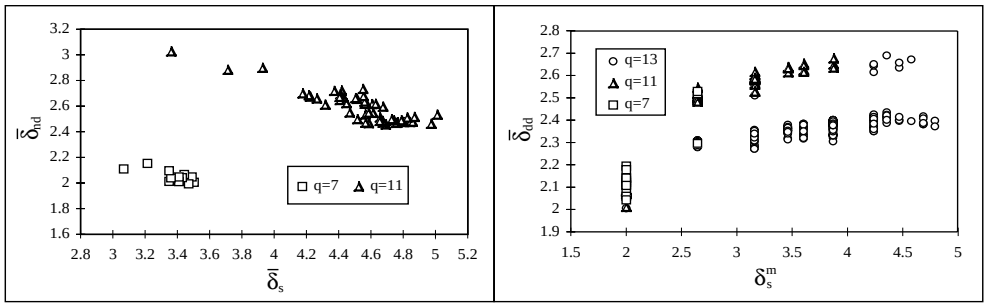


FIG. 5.2. *Measures for six-dimensional OA discretizations consisting of columns from the arrays* $\mathrm{OA}(7)_1$, $\mathrm{OA}(7)_2$, $\mathrm{OA}(11)_1$, *and* $\mathrm{OA}(13)_1$ *(plot* (b) *only). In plot* (a), *the average minimum-length slope,* $\overline{\delta}_s$, *is negatively correlated with the average distance from a nondiscretization point to the nearest discretization point,* $\overline{\delta}_{nd}$. *In plot* (b), *the smallest minimum-length slope,* δ_s^m, *is positively correlated with the average distance from a discretization point to its nearest neighbor in the discretization,* $\overline{\delta}_{dd}$.

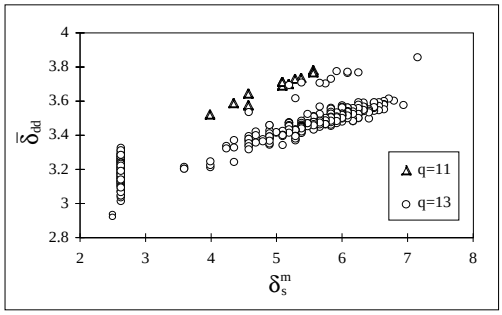


FIG. 5.3. *Measures for nine-dimensional OA discretizations consisting of columns from the arrays* $\mathrm{OA}(11)_1$ *and* $\mathrm{OA}(13)_1$. *The smallest minimum-length slope,* δ_s^m, *is positively correlated with the average distance from a discretization point to its nearest neighbor in the discretization,* $\overline{\delta}_{dd}$.

"poor" ones.

Finally, consider the case with $n = 9$ continuous variables. For $q = 11$ and 13, OA patterns were identified with $m = 12$ and 475, respectively. The measures δ_{nd}^M and $\overline{\delta}_{\mathrm{nd}}$ were not computed. Correlations are shown in Table 5.2(c). In Figure 5.3, an increasing linear relationship is visible among the $q = 11$ points, which only appear at the top of the plot. The $q = 13$ points are separated into two regions, one which parallels the $q = 11$ relationship, and one which contains the worst $\overline{\delta}_{\mathrm{dd}}$ values. There is overlap in the $\overline{\delta}_{\mathrm{dd}}$ values between the two regions, but, overall, a "poor" discretization can be avoided by choosing one with a high δ_{s}^m value.

5.5. Fitting the last period future value function in SDP. As an empirical evaluation of how space-filling goodness affects accuracy in function approximation, we consider the inventory-forecasting SDP problems of Chen (1999). These involve discrete state variables (inventory and demand forecasts) that have a wide enough range so that we may consider them to be near-continuous. SDP models a stochastic system over several, say T, time periods. The future value function in period $t \leq T$ provides the minimum cost to operate the system from time period t through T as a function of the state of the system at the beginning of period t. For the inventory-

TABLE 5.3

Correlations between mean absolute deviation (MAD) *or standard error* (SE) *and* $\overline{\delta}_{nd}$, δ^M_{nd}, $\overline{\delta}_{dd}$, δ^m_{dd}, $\overline{\delta}_s$, δ^m_s, *based on* SDP *data results. Notation:* n *is the number of state variables,* q *is the number of discretization levels in each dimension, and* m *is the number of* OAs *tested.*

	n	q	m		$\overline{\delta}_{\mathrm{nd}}$	δ^M_{nd}	$\overline{\delta}_{\mathrm{dd}}$	δ^m_{dd}	$\overline{\delta}_{\mathrm{s}}$	δ^m_{s}
(a)	4	5	8	MAD	0.78	0.78	−0.27	N/A	−0.78	N/A
				SE	0.78	0.77	−0.26	N/A	−0.77	N/A
		7	22	MAD	0.68	0.73	−0.55	N/A	−0.68	N/A
				SE	0.75	0.82	−0.62	N/A	−0.77	N/A
		11	8	MAD	0.95	0.96	−0.29	−0.27	−0.92	−0.27
				SE	0.93	0.95	−0.27	−0.26	−0.87	−0.27
		13	22	MAD	0.83	0.85	−0.34	−0.33	−0.67	−0.33
				SE	0.86	0.86	−0.40	−0.39	−0.71	−0.34
(b)	6	7	14	MAD	0.41	0.41	−0.41	−0.35	−0.29	−0.41
				SE	0.32	0.28	−0.31	−0.27	−0.26	−0.35
		11	50	MAD	0.66	0.64	−0.41	−0.28	−0.60	−0.47
				SE	0.72	0.65	−0.53	−0.44	−0.69	−0.50
		13	9	MAD	N/A	N/A	−0.80	−0.72	−0.90	−0.76
				SE	N/A	N/A	−0.85	−0.80	−0.86	−0.80
(c)	9	11	7	MAD	N/A	N/A	−0.73	N/A	−0.61	−0.80
				SE	N/A	N/A	−0.72	N/A	−0.61	−0.80
		13	8	MAD	N/A	N/A	−0.86	−0.76	−0.31	−0.85
				SE	N/A	N/A	−0.94	−0.83	−0.25	−0.93

forecasting SDPs, the true future value functions are unknown, but given the specific state of the system the true value of this function can be computed for the last period alone. Using this last period future value function, the curve fitting ability of "good" versus "poor" OA discretizations generated from the arrays $\mathrm{OA}(5)_1$, $\mathrm{OA}(5)_2$, $\mathrm{OA}(7)_1$, $\mathrm{OA}(7)_2$, $\mathrm{OA}(11)_1$, and $\mathrm{OA}(13)_1$ is considered with respect to the measures discussed in this paper.

For each OA discretization tested, a MARS approximation (Friedman (1991)) was evaluated, the true value of the last period future value function was computed at 10,000 randomly chosen initial states, and then the mean absolute deviation was computed over the 10,000 points. Correlations between mean absolute deviation, its standard error, and the six measures are shown in Table 5.3. We hope to see a negative relationship between mean absolute deviation and the minimax measures $\overline{\delta}_{\mathrm{nd}}$ and δ^M_{nd}, while a positive relationship is desired with the maximin and slicing measures, $\overline{\delta}_{\mathrm{dd}}$, δ^m_{dd}, $\overline{\delta}_{\mathrm{s}}$, and δ^m_{s}. The maximin measures $\overline{\delta}_{\mathrm{dd}}$ and δ^m_{dd} only achieved good correlation results for the nine-dimensional SDP in Table 5.3(c), and the six-dimensional SDP with $q = 11$ in Table 5.3(b). The smallest correlations for $\overline{\delta}_{\mathrm{nd}}$, δ^M_{nd}, $\overline{\delta}_{\mathrm{s}}$, and δ^m_{s} were found in Table 5.3(b) with the arrays from $\mathrm{OA}(7)_1$ and $\mathrm{OA}(7)_2$, implying that $q = 7$ was not sufficient for the six-dimensional SDP. As expected from the results in Table 5.2, our measure $\overline{\delta}_{\mathrm{s}}$ performed similarly to the minimax measures $\overline{\delta}_{\mathrm{nd}}$ and δ^M_{nd}, and our measure δ^m_{s} performed similarly to the maximin measures $\overline{\delta}_{\mathrm{dd}}$ and δ^m_{dd}.

In Figure 5.4 accuracy increases as $\overline{\delta}_{\mathrm{s}}$ increases, indicating that high values of this measure yield better OA discretizations. The plot of mean absolute deviation versus $\overline{\delta}_{\mathrm{nd}}$ is very similar to Figure 5.4, except with accuracy increasing as $\overline{\delta}_{\mathrm{nd}}$ decreases. The plot of mean absolute deviation versus $\overline{\delta}_{\mathrm{dd}}$ was not informative.

In Figure 5.5(a) the relationship is very different for the different values of q. For both $q = 11$ and $q = 13$, the "worst" OA discretizations have the worst accuracy, which is motivation to avoid OA discretizations with very low $\overline{\delta}_{\mathrm{s}}$ values. MARS was

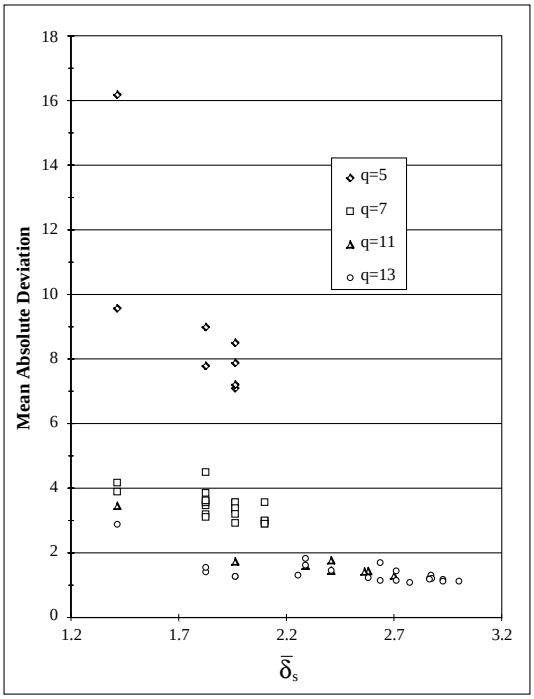
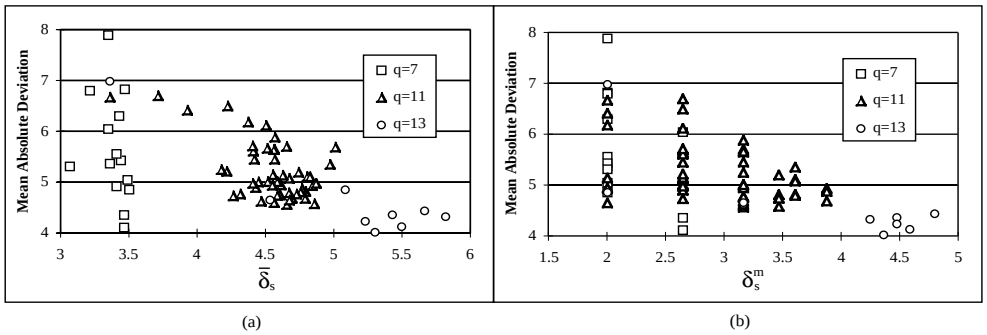


FIG. 5.4. *The four-dimensional* SDP *data. The average minimum-length slope,* $\overline{\delta}_s$*, is negatively correlated with mean absolute deviation.*



(a) (b)

FIG. 5.5. *The six-dimensional* SDP *data. In plot* (a)*, the average minimum-length slope,* $\overline{\delta}_s$*, is plotted versus mean absolute deviation. There is a visible negative trend, although for* $q = 7$ *it is very slight and for* $q = 11$ *there is a lack of data at low values of* $\overline{\delta}_s$*. In plot* (b)*, the smallest minimum-length slope,* δ_s^m*, is negatively correlated with mean absolute deviation. In addition, accuracy is more variable for smaller values of* δ_s^m*.*

only fit to a few $q = 13$ OAs because significant CPU was required by these OA discretizations. Again, the plot of mean absolute deviation versus $\overline{\delta}_{\rm nd}$ is similar. In Figure 5.5(b), the "best" OA discretizations are associated with the best accuracy, which is motivation to choose OA discretizations with high $\delta_{\rm s}^m$ values. The plot of mean absolute deviation versus $\overline{\delta}_{\rm dd}$ was messy and unclear.

For Figure 5.6 results were available for only a few OAs due to the size of these OA discretizations. Including the omitted data point at $\delta_{\rm s}^m = 4.0$, the "worst" OA

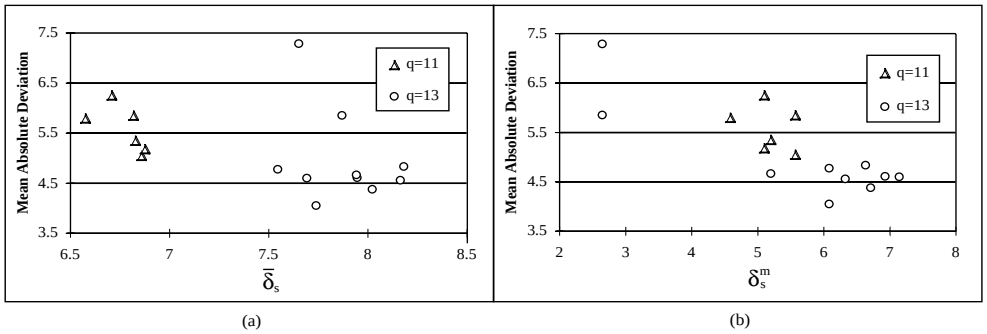


FIG. 5.6. *The nine-dimensional* SDP *data. In plot* (a)*, the average minimum-length slope,* $\overline{\delta}_s$*, is plotted versus mean absolute deviation. There is a negative trend when* $q = 11$*, but none visible when* $q = 13$*. In plot* (b)*, the smallest minimum-length slope,* δ_s^m*, appears to be negatively correlated with mean absolute deviation. In particular the two worst* δ_s^m *values for* $q = 13$ *are distinguished from the others. Missing from both plots is the* $q = 11$ OA *discretization, which had a mean absolute deviation of* 102.07 *with the worst* $\overline{\delta}_s$ *and* δ_s^m *values of* 6.566 *and* 4.0*, respectively.*

discretizations achieved the worst accuracy in Figure 5.6(b), reinforcing the notion of choosing OA discretizations with high δ_s^m values.

6. Conclusions. An efficient OA discretization enabled the development of the first truly high-dimensional continuous-state SDP solution method. Discretization can also be employed to generate scenarios for event spaces, and application of a discretization approach was discussed for SP in the context of a two-stage stochastic linear program. Two new space-filling measures were developed for the OAs used by Chen, Ruppert, and Shoemaker (1999) to discretize a continuous SDP state space. Comparisons were made against "minimax" measures (distance from any nondiscretization point to the nearest discretization point) and "maximin" measures (distance from any discretization point to its nearest neighbor in the discretization). Although only OAs of strength three were tested in this paper, the new "slicing" measures (distance between points in $\{n - d + 1\}$-dimensional slices) were derived for general strength d. With strength three OA discretizations, strong correlations were identified between one of the slicing measures and the minimax measures, and between the other slicing measure and the maximin measures. In addition, it was significantly faster to compute the slicing measures than the minimax measures, and it was somewhat faster than computing the maximin measures. Finally, in an empirical study of the last period for three inventory-forecasting SDP problems, it was found that the strength three OA discretizations deemed "good" by the slicing and minimax measures produced more accurate MARS approximations of the future value function than those deemed "poor." The maximin measures were generally inferior to the others but could become more effective in higher dimensional problems. Although only problems with up to nine dimensions were studied, the graphical justification discussed in section 4.2 holds in higher dimensions, implying similar expectations with larger problems. The results of this study provide motivation to employ space-filling measures for OAs, since not all OA discretizations are ideally spaced. One important issue to be considered in future work is to extend these measures to situations in which a nonuniform distribution (e.g., normal distribution) of points is desired.

Appendix A. Projective plane. The primary distinction between projective geometry and Euclidean geometry is that parallel lines are defined to meet at a single

"point at infinity" in projective geometry. Since lines in Euclidean geometry may be partitioned into equivalence classes of parallel lines, every distinct class of parallel lines is associated with a distinct point at infinity. In the projective plane, we complete the geometry by defining all points at infinity to lie on a single line at infinity. These extra points are called *ideal* points and the extra line is called the *ideal* line.

In the Euclidean plane, points are represented by pairs (x, y), and lines are represented by equations of the form

$$Ax + By + C = 0. \tag{A.1}$$

Two parallel lines have equations that may written as

$$\begin{aligned} Ax + By + C &= 0, \\ Ax + By + C' &= 0. \end{aligned} \tag{A.2}$$

In the projective plane, these two lines meet at an ideal point, so the coordinates of this ideal point must "solve" the equations of these lines. Instead of trying to solve the equations in (A.2), we introduce a third variable, z, and rewrite (A.1) as

$$Ax + By + Cz = 0.$$

When z is zero, the equations for two parallel lines become equivalent, thus the triple $(x, y, 0)$, where x and y satisfy

$$Ax + By = 0, \qquad \frac{-A}{B} = m, \tag{A.3}$$

represents the ideal point for lines with slope m. For fixed m, there are several eligible triples $(x, y, 0)$ that satisfy (A.3), and they are all representations of the same ideal point. When z is nonzero, then the triple (x, y, z) represents a finite point which has the unique representation in Euclidean coordinates (u, v), where $u = x/z$ and $v = y/z$. As with ideal points, there are several eligible triples (x, y, z) that can represent the finite point with Euclidean coordinates (u, v).

Placing z as the last coordinate is arbitrary, and we will now define the first coordinate of the triple (x_1, x_2, x_3) to be zero for ideal points and nonzero otherwise. The *duality* of the projective plane permits representation of a line by the coordinates $[u_1, u_2, u_3]$, where not all u_i are zero, corresponding to the line with equation

$$u_1x_1 + u_2x_2 + u_3x_3 = 0. \tag{A.4}$$

Note that this equation is symmetric in the u's and x's and is satisfied by the point (x_1, x_2, x_3) and the line $[u_1, u_2, u_3]$ if and only if the point lies on the line or the line passes through the point (equivalent incidence statements). Thus, given $[u_1, u_2, u_3]$, (A.4) is the *equation of a point*.

A finite projective plane has points whose coordinates are elements of a Galois field. For p, a positive prime number, and k, a positive integer, let $q = p^k$. The finite projective plane based on the Galois field $\mathrm{GF}(p^k)$ is denoted by $\mathrm{PG}(2, s)$. The projective plane $\mathrm{PG}(2, s)$ consists of $q^2 + q + 1$ points (x_1, x_2, x_3) and $q^2 + q + 1$ lines $[u_1, u_2, u_3]$, where all coordinates are elements of $\mathrm{GF}(p^k)$. Since each finite line additionally has an ideal point lying on it, there are $q + 1$ points on every line. Since the ideal line passes through every ideal point, there are also $q+1$ lines through every point. Thus the duality principle is upheld. This is dual for "all points lying on a

line." All the lines in the pencil can be represented as a weighted sum of two distinct lines in that pencil.

Appendix B. Coefficients of a point conic. Given two pencils through two distinct points, P_1 and P_2, one can find the points of a point conic with respect to a certain projectivity. Given any two distinct lines $\boldsymbol{u}_1$ and $\boldsymbol{v}_1$, from the pencil through P_1, then all the lines in the pencil through P_1 can be expressed in the form $c_1\boldsymbol{u}_1+d_1\boldsymbol{v}_1$, where c_1 and d_1 are elements of GF(p^k). Similarly, select two distinct lines $\boldsymbol{u}_2$ and $\boldsymbol{v}_2$ from the pencil through P_2. A point conic is constructed by projecting the lines in the pencil through P_1 into the lines in the pencil through P_2. The equation

$$k\begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{pmatrix} d_1 \\ d_2 \end{pmatrix} \tag{B.1}$$

determines a projectivity from the pencil through P_1 onto the pencil through P_2 in which the line $c_1\boldsymbol{u}_1 + d_1\boldsymbol{v}_1$ is transformed into the line $c_2\boldsymbol{u}_2 + d_2\boldsymbol{v}_2$ for $kc_1 = c_2$ and $kd_1 = d_2$. Given the projectivity in (B.1), the ratios c_1/d_1 and c_2/d_2 must be equal.

A point in the conic, denoted by a column vector as $\boldsymbol{x} = (x_1, x_2, x_3)^T$, occurs at the intersection of corresponding members of the pencils; thus $\boldsymbol{x}$ must lie on both $c_1\boldsymbol{u}_1 + d_1\boldsymbol{v}_1$ and $c_2\boldsymbol{u}_2 + d_2\boldsymbol{v}_2$, i.e.,

$$\begin{aligned} (c_1\boldsymbol{u}_1 + d_1\boldsymbol{v}_1)^T\boldsymbol{x} &= 0 \text{ and} \\ (c_2\boldsymbol{u}_2 + d_2\boldsymbol{v}_2)^T\boldsymbol{x} &= 0. \end{aligned} \tag{B.2}$$

Denote the lines $\boldsymbol{u}_1$, $\boldsymbol{v}_1$, $\boldsymbol{u}_2$, and $\boldsymbol{v}_2$ by column vectors. For example, let $\boldsymbol{u}_1 = [u_{11}, u_{12}, u_{13}]^T$. Then from (B.2) we can express the ratios as

$$\frac{c_1}{d_1} = -\frac{v_{11}x_1 + v_{12}x_2 + v_{13}x_3}{u_{11}x_1 + u_{12}x_2 + u_{13}x_3}, \tag{B.3}$$

$$\frac{c_2}{d_2} = -\frac{v_{21}x_1 + v_{22}x_2 + v_{23}x_3}{u_{21}x_1 + u_{22}x_2 + u_{23}x_3}. \tag{B.4}$$

Setting the right-hand sides of (B.3) and (B.4) equal, we can rewrite this relationship as a homogeneous second-degree equation for the point conic

$$\sum_{i=1}^{3}\sum_{j=1}^{3} b_{ij}x_ix_j = 0, \qquad \text{where } b_{ij} = u_{1i}v_{2j} - u_{2j}v_{1i}.$$

Letting $a_{ij} = a_{ji} = (b_{ij} + b_{ji})/2$, we find the *symmetric* homogeneous second-degree equation for the point conic $\sum_{i=1}^{3}\sum_{j=1}^{3} a_{ij}x_ix_j = \boldsymbol{x}^TA\boldsymbol{x} = 0$. The coefficients a_{ij} are those calculated in (4.1).

Appendix C. Specific OA discretizations. The algorithm in section 4.1 was used to generate the six OA($q^3, q+1, q, 3$) discretizations discussed in section 5.2. The OA($5^3, 6, 5, 3$) discretization denoted by OA$(5)_1$ was created by choosing

$$\begin{aligned} P_1 &= (1,0,0), \quad \boldsymbol{u}_1 = [0,1,1], \quad \boldsymbol{v}_1 = [0,2,1], \\ P_2 &= (0,1,0), \quad \boldsymbol{u}_2 = [1,0,1], \quad \boldsymbol{v}_2 = [1,0,2], \end{aligned}$$

which then generated

$$A = \begin{bmatrix} 0 & 4 & 0 \\ 4 & 0 & 0 \\ 0 & 0 & 2 \end{bmatrix} \text{ and } C^T = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 4 & 4 \\ 0 & 0 & 1 & 4 & 2 & 3 \end{bmatrix}.$$

The $\mathrm{OA}(5^3, 6, 5, 3)$ discretization denoted by $\mathrm{OA}(5)_2$ was created by choosing

$$\begin{aligned} P_1 &= (1,2,3), \quad \boldsymbol{u}_1 = [1,0,3], \quad \boldsymbol{v}_1 = [1,2,0], \\ P_2 &= (1,2,4), \quad \boldsymbol{u}_2 = [0,1,2], \quad \boldsymbol{v}_2 = [1,3,2], \end{aligned}$$

which then generated

$$A = \begin{bmatrix} 2 & 2 & 3 \\ 2 & 1 & 0 \\ 3 & 0 & 2 \end{bmatrix} \text{ and } C^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 2 & 4 & 4 \\ 1 & 1 & 3 & 4 & 3 & 4 \end{bmatrix}.$$

The $\mathrm{OA}(7^3, 8, 7, 3)$ discretization denoted by $\mathrm{OA}(7)_1$ was created by choosing

$$\begin{aligned} P_1 &= (1,0,0), \quad \boldsymbol{u}_1 = [0,1,1], \quad \boldsymbol{v}_1 = [0,1,3], \\ P_2 &= (1,3,2), \quad \boldsymbol{u}_2 = [1,0,3], \quad \boldsymbol{v}_2 = [1,2,0], \end{aligned}$$

which then generated

$$A = \begin{bmatrix} 0 & 0 & 5 \\ 0 & 4 & 6 \\ 5 & 6 & 3 \end{bmatrix} \text{ and } C^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 3 & 3 & 5 & 5 \\ 0 & 6 & 4 & 5 & 2 & 6 & 2 & 5 \end{bmatrix}.$$

The $\mathrm{OA}(7^3, 8, 7, 3)$ discretization denoted by $\mathrm{OA}(7)_2$ was created by choosing

$$\begin{aligned} P_1 &= (1,0,2), \quad \boldsymbol{u}_1 = [1,1,3], \quad \boldsymbol{v}_1 = [1,3,3], \\ P_2 &= (1,5,3), \quad \boldsymbol{u}_2 = [1,0,2], \quad \boldsymbol{v}_2 = [1,2,1], \end{aligned}$$

which then generated

$$A = \begin{bmatrix} 0 & 0 & 6 \\ 0 & 4 & 1 \\ 6 & 1 & 1 \end{bmatrix} \text{ and } C^T = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 4 & 4 & 5 & 6 \\ 1 & 4 & 0 & 2 & 3 & 5 & 3 & 2 \end{bmatrix}.$$

The $\mathrm{OA}(11^3, 12, 11, 3)$ discretization denoted by $\mathrm{OA}(11)_1$ as created by choosing

$$\begin{aligned} P_1 &= (1,0,0), \quad \boldsymbol{u}_1 = [0,1,1], \quad \boldsymbol{v}_1 = [0,1,3], \\ P_2 &= (1,3,2), \quad \boldsymbol{u}_2 = [1,0,5], \quad \boldsymbol{v}_2 = [1,5,3], \end{aligned}$$

which then generated

$$A = \begin{bmatrix} 0 & 0 & 9 \\ 0 & 10 & 3 \\ 9 & 3 & 9 \end{bmatrix} \text{ and } C^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 3 & 3 & 5 & 5 & 8 & 8 & 9 & 9 & 10 & 10 \\ 0 & 9 & 2 & 5 & 5 & 8 & 1 & 10 & 1 & 2 & 8 & 9 \end{bmatrix}.$$

The $\mathrm{OA}(13^3, 14, 13, 3)$ discretization denoted by $\mathrm{OA}(13)_1$ was created by choosing

$$\begin{aligned} P_1 &= (1,0,0), \quad \boldsymbol{u}_1 = [0,1,1], \quad \boldsymbol{v}_1 = [0,1,3], \\ P_2 &= (1,0,2), \quad \boldsymbol{u}_2 = [1,0,6], \quad \boldsymbol{v}_2 = [1,11,6], \end{aligned}$$

which then generated

$$A = \begin{bmatrix} 0 & 0 & 6 \\ 0 & 6 & 2 \\ 6 & 2 & 7 \end{bmatrix} \text{ and } C^T = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 2 & 2 & 3 & 5 & 5 & 10 & 10 & 12 \\ 7 & 11 & 0 & 2 & 3 & 4 & 5 & 7 & 2 & 4 & 10 & 3 & 10 & 5 \end{bmatrix}.$$

Acknowledgment. The author thanks Alan King for his invaluable comments, which greatly improved the presentation of this material.

REFERENCES

L. M. Batten (1986), *Combinatorics of Finite Geometries*, Cambridge University Press, New York.
R. E. Bellman (1957), *Dynamic Programming*, Princeton University Press, Princeton, NJ.
J. R. Birge and F. Louveaux (1977), *Introduction to Stochastic Programming*, Springer-Verlag, New York.
R. C. Bose and K. A. Bush (1952), *Orthogonal arrays of strength two and three*, Annals of Mathematical Statistics, 23, pp. 508–524.
KABush (1952), *Orthogonal arrays of index unity*, Annals of Mathematical Statistics, 23, pp. 426–434.
H. Caffey, L. Z. Liao, and C. A. Shoemaker (1993), *Parallel processing of large scale discrete-time unconstrained differential dynamic programming*, Parallel Comput., 19, pp. 1003–1018.
V. C. P. Chen, D. Ruppert, and C. A. Shoemaker (1999), *Applying experimental design and regression splines to high-dimensional continuous-state stochastic dynamic programming*, Oper. Res., 47, pp. 38–53.
V. C. P. Chen (1999), *Application of MARS and orthogonal arrays to inventory forecasting stochastic dynamic programs*, Comput. Statist. Data Anal., 30, pp. 317–341.
G.B.Dantzig (1955), *Linear programming under uncertainty*, Management Science, 1, pp. 197–206.
J. H. Friedman (1991), *Multivariate adaptive regression splines (with discussion)*, Ann. Statist., 19, pp. 1–141.
M. E. Johnson, L. M. Moore, and D. Ylvisaker (1990), *Minimax and maximin distance designs*, J. Statist. Plann. Inference, 26, pp. 131–148.
J. R. Kalagnanam and U. M. Diwekar (1997), *An efficient sampling technique for off-line quality control*, Technometrics, 39, pp. 308–319.
A. King (1988), *Stochastic Programming Problems: Examples from the Literature*, in Numerical Techniques for Stochastic Optimization, Y. Ermoliev and R. Wets, eds., Springer-Verlag, Berlin, pp. 543–567.
M. D. McKay, W. J. Conover, and R. J. Bechman (1979), *A comparison of three methods for selecting values of input variables in the analysis of output from a computer code*, Technometrics, 21, pp. 239–245.
D. C. Montgomery (1997), *Design and Analysis of Experiments*, Wiley, New York.
G. L. Nemhauser (1966), *Introduction to Dynamic Programming*, Wiley, New York.
A. B. Owen (1992), *Orthogonal arrays for computer experiments, integration, and visualization*, Statist. Sinica, 2, pp. 439–452.
A. B. Owen (1995), *Randomly permuted (t, m, s)-nets and (t, s)-sequences*, in Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing, H. Niederreiter and P. J.-S. Shiue, eds., Lecture Notes in Statist. 106, Springer-Verlag, New York, pp. 299–315.
K. D. Palmer (1998), *Data Collection Plans and Meta Models for Chemical Process Flowsheet Simulators*, Ph.D. thesis, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA.
K. D. Palmer and K. L. Tsui (2001), *A minimum bias Latin hypercube design*, Institute of Industrial Engineers Transactions, 33, pp. 793–808.
M. L. Puterman (1994), *Markov Decision Processes*, Wiley, New York.
J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn (1989), *Design and analysis of computer experiments*, Statist. Sci., 4, pp. 409–435.
B. Tang (1993), *Orthogonal array-based Latin hypercubes*, J. Amer. Statist. Assoc., 88, pp. 1392–1397.
D. J. White (1988), *Further real applications of Markov decision processes*, Interfaces, 18, pp. 55–61.

SIAM J. OPTIM.
Vol. 12, No. 2, pp. 345–375

TIGHTER LINEAR AND SEMIDEFINITE RELAXATIONS FOR MAX-CUT BASED ON THE LOVÁSZ–SCHRIJVER LIFT-AND-PROJECT PROCEDURE*

MONIQUE LAURENT†

Abstract. We study how the lift-and-project method introduced by Lovász and Schrijver [*SIAM J. Optim.,* 1 (1991), pp. 166–190] applies to the cut polytope. We show that the cut polytope of a graph can be found in k iterations if there exist k edges whose contraction produces a graph with no K_5-minor. Therefore, for a graph G with $n \geq 4$ nodes with stability number $\alpha(G)$, $n-4$ iterations suffice instead of the m (number of edges) iterations required in general and, under some assumption, $n-\alpha(G)-3$ iterations suffice. The exact number of needed iterations is determined for small $n \leq 7$ by a detailed analysis of the new relaxations. If positive semidefiniteness is added to the construction, then one finds in one iteration a relaxation of the cut polytope which is tighter than its basic semidefinite relaxation and than another one introduced recently by Anjos and Wolkowicz [*Discrete Appl. Math.,* to appear]. We also show how the Lovász–Schrijver relaxations for the stable set polytope of G can be strengthened using the corresponding relaxations for the cut polytope of the graph G^{∇} obtained from G by adding a node adjacent to all nodes of G.

Key words. linear relaxation, semidefinite relaxation, lift-and-project, cut polytope, stable set polytope

AMS subject classifications. 05C50, 15A57, 52B12, 90C22, 90C27

PII. S1052623400379371

1. Introduction.

Lovász and Schrijver [22] have introduced a method for constructing a higher dimensional convex set whose projection $N(K)$ approximates the convex hull P of the 0–1 valued points in a polytope K defined by a given system of linear inequalities. If the linear system is in d variables, the convex set consists of symmetric matrices of order $d+1$ satisfying certain linear conditions. A fundamental property of the projection $N(K)$ is that one can optimize over it in polynomial time and thus find an approximate solution to the original problem in polynomial time. Moreover, after d iterations of the operator N, one finds the polytope P. Lovász and Schrijver [22] also introduce some strengthenings of the basic construction; in particular, adding positive semidefinite constraints leads to the operator N_+, and adding stronger linear conditions in the definition of the higher dimensional set of matrices leads to the operators N' and N'_+. They study in detail how the method applies to the stable set polytope. Starting with $K = \mathrm{FRAC}(G)$ (the fractional stable set polytope defined by nonnegativity and the edge constraints), they show that in one iteration of the N operator one obtains all odd hole inequalities (and no more), while in one iteration of the N_+ operator one obtains many inequalities including odd wheel, clique, and odd antihole inequalities and orthogonality constraints; therefore, the relaxation $N_+(\mathrm{FRAC}(G))$ is tighter than the basic semidefinite relaxation of the stable set polytope by the theta body $\mathrm{TH}(G)$. In particular, this method permits one to solve the maximum stable set problem in a t-perfect graph or in a perfect graph in polynomial time. They also show that the stable set polytope of G is found after at most $n-\alpha(G)-1$ iterations of the N operator (resp., $\alpha(G)$ iterations of the N_+ operator) applied to $\mathrm{FRAC}(G)$, if G has at least one edge.

*Received by the editors October 12, 2000; accepted for publication (in revised form) May 18, 2001; published electronically November 13, 2001.
http://www.siam.org/journals/siopt/12-2/37937.html
†CWI, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands (monique@cwi.nl).

On the other hand, there exist "easy" polytopes P (meaning that their linear description is known and one can optimize over them in polynomial time) for which the number of iterations of the N or N_+ operators needed in order to find P grows linearly with the dimension of P. For example, Stephen and Tunçel [29] showed that n iterations are needed for finding the matching polytope of K_{2n+1} (starting with the polytope defined by nonnegativity and the degree constraints) using the N_+ operator. Recently, Cook and Dash [8] and Goemans and Tunçel [12] constructed examples where positive semidefiniteness does not help; namely, the same number d of iterations is needed for finding some d-dimensional polytope P using the N or the N_+ operator. This is the case, for instance, for the polytope $P := \{x \in \mathbf{R}^d \mid \sum_{i=1}^d x_i \geq 1\}$ if we start from its relaxation $K := \{x \in \mathbf{R}^d \mid \sum_{i=1}^d x_i \geq \frac{1}{2}\}$.

In this paper we study how the method applies to the cut polytope when starting with its linear relaxation by the metric polytope $\mathrm{MET}(G)$ (to be defined later). When using the operator N_+, one obtains in one iteration a semidefinite relaxation of the cut polytope which is tighter than its basic semidefinite relaxation and also tighter than a refinement of the basic relaxation introduced recently by Anjos and Wolkowicz [2]. One can, in fact, refine the relaxation $N(\mathrm{MET}(G))$ by first applying the N operator to the metric polytope of the complete graph and then projecting on the edge set of the graph; the relaxation denoted as $N(G)$ obtained in this way satisfies $\mathrm{CUT}(G) \subseteq N(G) \subseteq N(\mathrm{MET}(G))$. We consider in this paper both constructions $N(G)$ and $N(\mathrm{MET}(G))$, also for the stronger operators N_+, N', N'_+ and their iterates.

We show that $\mathrm{CUT}(G) = N^k(\mathrm{MET}(G))$ if there exist k edges in G whose contraction produces a graph with no K_5-minor. In particular, the cut polytope of a graph on n nodes can be found after $n-4$ (resp., $n-5$) iterations of the N (resp., N') operator if $n \geq 4$ (resp., $n \geq 6$) (while the cut polytope has dimension m, the number of edges of the graph). Moreover, if G has stability number $\alpha(G)$, then $\mathrm{CUT}(G) = N^k(G)$, where $k := \max(0, n - \alpha(G) - 3)$; equality $\mathrm{CUT}(G) = N^k(\mathrm{MET}(G))$ holds if there exists a maximum stable set in G whose complement induces a graph with at most three connected components. The upper bound $n - \alpha(G) - 3$ is similar to the upper bound in [22] for the stable set polytope. It is well known that the stable set polytope $\mathrm{STAB}(G)$ can be realized as a face of the cut polytope $\mathrm{CUT}(G^\nabla)$, where G^∇ is obtained by adding a new node to G adjacent to all nodes of G; moreover, an analogous relation exists between their basic linear and positive semidefinite relaxations. We study how this fact extends to their relaxations obtained via the Lovász–Schrijver procedure. Namely, we show that $N^k(\mathrm{MET}(G^\nabla))$ (resp., $\nu^k(\mathrm{MET}(G^\nabla))$) yields a relaxation of $\mathrm{STAB}(G)$ which is tighter than $N^{k+1}(\mathrm{FRAC}(G))$ (resp., $\nu^k(\mathrm{FRAC}(G))$ for $\nu = N_+, N', N'_+$).

Although the inclusion $N_+(\mathrm{MET}(G)) \subseteq N(\mathrm{MET}(G))$ is strict for certain graphs (e.g., for any complete graph on $n \geq 6$ nodes), we do not know of an example of a graph G for which the number of iterations needed for finding $\mathrm{CUT}(G)$ is smaller when using the operator N_+ than when using the operator N. This contrasts with the case of the stable set polytope where, for instance, $\mathrm{STAB}(K_n)$ is found in one iteration of the N_+ operator applied to $\mathrm{FRAC}(G)$, while $n-2$ iterations of the N operator are needed.

The paper is organized as follows. Section 2 gives a general description of the Lovász–Schrijver (LS) procedure, and section 3 contains a presentation of the various relaxations of the cut polytope considered in the paper. In section 4, we study the index of a graph (the smallest number of iterations of the LS procedure needed for finding its cut polytope); upper bounds are proved in sections 4.1 and 4.3, the behavior

of the index under taking graph minors and clique sums is investigated in section 4.4, and a number of needed technical tools are provided in section 4.2. We study in section 5 the validity of hypermetric inequalities for the new relaxations, which enables us to determine the exact value of the index of a graph on $n \leq 7$ nodes; some technical proofs are delayed until section 7. Finally, in section 6 we study the links between the LS relaxations for the cut polytope and the original LS relaxations for the stable set polytope.

2. The LS Procedure. Let $F \subseteq \{\pm 1\}^d$, let $P := \text{conv}(F)$ be the integral polytope whose linear description one wishes to find, and let

$$K = \{x \in \mathbf{R}^d \mid Ax \geq b\}$$

be a linear relaxation of P such that $K \subseteq [-1,1]^d$ and $K \cap \{\pm 1\}^d = F$ (K is a *linear programming formulation* for P).

Starting from K, the LS method constructs a hierarchy of linear relaxations for P which in d steps finds the exact description of P. The basic idea is as follows. If we multiply an inequality $a^T x \geq \beta$, valid for F, by $1 \pm x_i \geq 0$, we obtain two nonlinear inequalities which remain valid for F. Applying this to all the inequalities from the system $Ax \geq b$, substituting x_i^2 by 1, and linearizing $x_i x_j$ by a new variable y_{ij} for $i \neq j$, we obtain a polyhedron in the $\binom{d+1}{2}$-space whose projection $N(K)$ on the original d-space contains P and is contained in K. The method was described in [22] in terms of 0–1 variables, but for our application to the max-cut problem it is more convenient to work with ± 1 variables, which is why we present it here in this setting.

It is useful to reformulate the construction in matrix terms. First we introduce some notation. As it is often more convenient to work with homogeneous systems of inequalities, i.e., with cones rather than polytopes, one embeds the d-space into $\mathbf{R}^{d+1}$ as the hyperplane: $x_0 = 1$. For a polytope P in $\mathbf{R}^d$, $\tilde{P} := \{\lambda(1,x) \mid x \in P, \lambda \geq 0\}$ denotes the cone in $\mathbf{R}^{d+1}$ obtained by homogenization of P; thus $P = \{x \in \mathbf{R}^d \mid (1,x) \in \tilde{P}\}$. Given a cone K, its *dual cone* K^* is defined as

$$K^* = \{y \mid y^T x \geq 0 \ \text{ for all } x \in K\}.$$

Consider the cube $Q := [-1,1]^d$ and its homogenization $\tilde{Q} = \{(x_0, x) \in \mathbf{R}^{d+1} \mid -x_0 \leq x_i \leq x_0 \ \text{ for all } i = 1, \ldots, d\}$. Thus the dual cone of $\tilde{Q}$ is generated by the $2d$ vectors $e_0 \pm e_i$ ($i = 1, \ldots, d$), where $e_0, e_1, \ldots, e_d$ denote the standard unit vectors in $\mathbf{R}^{d+1}$.

Given two polytopes $K_1 \subseteq K_2 \subseteq Q$, let $M(K_1, K_2)$ denote the set of symmetric matrices $Y = (y_{ij})_{i,j=0}^d$ satisfying the conditions

$$y_{i,i} = y_{0,0} \quad \text{for } i = 1, \ldots, d, \tag{2.1}$$

$$Y \tilde{K}_2^* \subseteq \tilde{K}_1, \tag{2.2}$$

and set

$$N(K_1, K_2) := \{x \in \mathbf{R}^d \mid (1,x) = Y e_0 \ \text{ for some } Y \in M(K_1, K_2)\}.$$

One can easily verify that

$$K_1 \cap \{\pm 1\}^d \subseteq N(K_1, K_1) \subseteq N(K_1, K_2) \subseteq N(K_1, Q) \subseteq K_1.$$

Therefore, the choice $(K_1, K_2) = (K, K)$ provides the best relaxation $N(K, K)$ for P. However, it is also interesting to consider the choice $(K_1, K_2) = (K, Q)$, giving the weaker relaxation $N(K, Q)$, as it behaves better algorithmically. Indeed, as observed in [22], if one can solve in polynomial time the (weak) separation problem over K, then the same holds for $M(K, Q)$ and thus also for its projection $N(K, Q)$; this property holds for $N(K, K)$ under the more restrictive assumption that an explicit linear description whose size is polynomial is known for K (details will be given later in this section).

One can obtain tighter relaxations for P by iterating the constructions $N(K, Q)$ and $N(K, K)$. One can iterate the construction $N(K, Q)$ by the sequence $N(K, Q)$, $N(N(K, Q), Q)$, etc. A first way in which the construction $N(K, K)$ can be iterated is by considering the sequence $N(K, K)$, $N(N(K, K), N(K, K))$, etc. A major drawback is then that, even if K is given by an explicit linear system of polynomial length, it is not clear whether this holds for the next iterate $N(K, K)$. A more tractable way is to consider the sequence $N(K, K)$, $N(N(K, K), K)$, etc. For simplicity in the notation, for a polytope $H \subseteq K \subseteq Q$ set

$$M(H) := M(H, Q),\ M'(H) := M(H, K),\ N(H) := N(H, Q),\ N'(H) := N(H, K).$$

The sequences $K, N(K, Q), N(N(K, Q), Q), \ldots$ and $K, N(K, K), N(N(K, K), K), \ldots$ can then be defined iteratively by

$$N^0(K) = (N')^0(K) := K, \qquad N^k(K) := N(N^{k-1}(K), Q),$$
$$(N')^k(K) := N((N')^{k-1}(K), K)$$

for $k \geq 1$. Thus $x \in \nu^k(K)$ if and only if $(1, x) = Ye_0$ for some $Y \in \mu(\nu^{k-1}(K))$, where $\mu = M$ (resp., M') if $\nu = N$ (resp., N').

One can reinforce the operators N and N' by adding positive semidefiniteness constraints. For a polytope $H \subseteq Q$, define $M_+(H)$ (resp., $M'_+(H)$) as the set of *positive semidefinite* matrices $Y \in M(H)$ (resp., $Y \in M'(H)$); the projections $N_+(H)$ and $N'_+(H)$ and their iterates are then defined in the obvious way. The following hierarchy holds:

$$P \subseteq N'_+(K) \subseteq N'(K) \subseteq N(K) \subseteq K, \quad P \subseteq N'_+(K) \subseteq N_+(K) \subseteq N(K) \subseteq K. \tag{2.3}$$

For membership in $M(K)$, condition (2.2) can be rewritten as

$$Y(e_0 \pm e_i) \in \tilde{K} \quad \text{for } i = 1, \ldots, d. \tag{2.4}$$

As $Ye_0 = \frac{1}{2}(Y(e_0 + e_i) + Y(e_0 - e_i))$, we deduce that

$$N(K) \subseteq \operatorname{conv}(K \cap \{x \mid x_i = \pm 1\}) \ \text{ for any } i = 1, \ldots, d. \tag{2.5}$$

Using this fact and induction, one can prove that after d iterations of the operator N, one finds the polytope P.

THEOREM 2.1 (see [22]). $N^d(K) = P$.

Obviously, the same holds for the operators N_+, N', or N'_+, but the corresponding sequences of relaxations may converge faster to P.

2.1. Comparison with other lift-and-project methods. Other lift-and-project methods have been proposed in the literature, in particular by Balas, Ceria, and Cornuéjols [3], by Sherali and Adams [28], and, recently, by Lasserre [16, 17].

Each of these methods produces a hierarchy of linear or semidefinite (in the case of Lasserre) relaxations: $P \subseteq K^d \subseteq \cdots \subseteq K^1 \subseteq K$ such that $P = K^d$. For $k \geq 1$, the kth iterate $S_k(K)$ in the Sherali–Adams hierarchy is obtained by multiplying the system $Ax \geq b$ by each of the products $\prod_{i\in I}(1+x_i)\prod_{j\in J}(1-x_j)$ for $I, J \subseteq [1,d]$ disjoint with $|I \cup J| = k$ and then replacing each square x_i^2 by 1, linearizing each product $\prod_{i\in I} x_i$, and projecting back on $\mathbf{R}^d$; hence, the first step is identical to the first step of the LS method, i.e., $S_1(K) = N(K)$. It is shown in [22] that $S_t(K) \subseteq N^k(K)$ (see [18] for a simple proof).

The first relaxation $P_i(K)$ in the Balas–Ceria–Cornuéjols hierarchy is obtained by multiplying $Ax \geq b$ by $1 \pm x_i$ for some given $i \in [1,d]$ (and then linearizing and projecting back on $\mathbf{R}^d$); the next relaxations are defined iteratively by $P_{i_1\dots i_k}(K) := P_{i_k}(P_{i_1\dots i_{k-1}}(K))$. It is shown in [3] that $P_{i_1\dots i_k}(K) = \operatorname{conv}(K \cap \{x \mid x_{i_1}, \dots, x_{i_k} = \pm 1\})$. Setting

$$N_0(K) := \bigcap_{i=1}^{d} P_i(K) = \bigcap_{i=1}^{d} \operatorname{conv}(K \cap \{x \mid x_i = \pm 1\}), \tag{2.6}$$

we deduce from (2.5) that

$$N(K) \subseteq N_0(K), \tag{2.7}$$

and thus $N^k(K) \subseteq N_0^k(K) = \bigcap_{i_1\dots i_k} P_{i_1\dots i_k}(K)$ for $k \geq 1$. In fact, $N_0(K)$ can be seen as the "noncommutative" analogue of $N(K)$, as $N_0(K) = \{x \in \mathbf{R}^d \mid (1,x) = Ye_0 \text{ for some } Y \in M_0(K)\}$, where $M_0(K)$ is the set of matrices (not necessarily symmetric) satisfying (2.1) and (2.4).

Using facts about moment sequences and representations of positive polynomials as sums of squares, Lasserre [16, 17] introduces a new hierarchy of semidefinite relaxations $Q_k(K)$ of P. It is shown in [18] that this new hierarchy refines the LS hierarchy; that is, $Q_k(K) \subseteq N_+^k(K)$, and its relation to the Sherali–Adams hierarchy is explained.

2.2. Algorithmic aspects. Given a convex body $B \subseteq \mathbf{R}^d$, the *separation problem* for B is the problem of deciding whether a given point $y \in \mathbf{R}^d$ belongs to B and, if not, of finding a hyperplane separating y from B; the *weak separation problem* is the analogous problem where one allows for numerical errors. An important application of the ellipsoid method is that if one can solve in polynomial time the weak separation problem for B, then one can optimize any linear objective function over B in polynomial time (with an arbitrary precision), and vice versa. (One should assume some technical information over B, like the knowledge of a ball contained in B and of a ball containing B.) See [14] for details.

An important property of the LS construction is that if one can solve in polynomial time the weak separation problem for K, then the same holds for $M(K)$ and $M_+(K)$, and thus for their projections $N(K)$ and $N_+(K)$. Therefore, for any fixed k, one can optimize in polynomial time a linear objective function over the relaxations $N^k(K)$ and $N_+^k(K)$; the same holds for the relaxations $S_k(K)$ and $P_{i_1\dots i_k}(K)$ of Sherali–Adams and of Balas–Ceria–Cornuéjols. For the operators N' and N'_+ and for the Lasserre hierarchy, an analogous result holds under the more restrictive assumption that an explicit linear description is known for K whose size is part of the input data.

2.3. Identifying valid inequalities for $N(K)$ and $N_+(K)$. We mention two results from [22] permitting us to construct inequalities valid for $N(K)$ and $N_+(K)$; the first one follows directly from (2.5) and we prove the second one for completeness.

LEMMA 2.2. *Suppose that, for some $i = 1, \dots, d$, the inequality $a^T x \geq \beta$ is valid for $K \cap \{x \mid x_i = \pm 1\}$. Then the inequality $a^T x \geq \beta$ is valid for $P_i(K)$ and thus for $N_0(K)$ and $N(K)$.*

LEMMA 2.3. *Suppose that $a_i \geq 0$ for $i = 1, \dots, d$ and $\beta \leq 0$. If the inequality $a^T x \geq \beta$ is valid for $K \cap \{x \mid x_i = -1\}$ for every i for which $a_i > 0$, then the inequality $a^T x \geq \beta$ is valid for $N_+(K)$.*

Proof. Set $b := (-\beta, a) \in \mathbf{R}^{d+1}$; thus $b \geq 0$. Let $Y \in M_+(K)$. We show that $b^T Y e_0 \geq 0$. By the assumption, we know that $b^T Y(e_0 - e_i) \geq 0$ if $a_i > 0$. Multiplying both sides of the inequality by a_i and summing over $i = 1, \dots, d$ yields

$$\left(\sum_{i=1}^d a_i\right) b^T Y e_0 \geq b^T Y \left(\sum_{i=1}^d a_i e_i\right) = b^T Y(b + \beta e_0),$$

and thus $(\sum_i a_i - \beta) b^T Y e_0 \geq b^T Y b$. The result now follows since $b^T Y b \geq 0$ (as Y is positive semidefinite) and $\sum_i a_i - \beta > 0$ (else, there is nothing to prove). □

2.4. Comparing $N_+(K)$ with the basic semidefinite relaxation in the equality case. The relaxation $N_+(K)$ is often stronger than some basic semidefinite relaxation one can think of for the problem at hand; this is the case for the stable set problem and for max-cut (see later) and, as we see now, when K is defined by an equality system. Suppose that $K = \{x \in \mathbf{R}^d \mid Ax = b\}$. The set $\hat{K}$ consisting of the vectors $x \in \mathbf{R}^d$ for which there exists a positive semidefinite matrix $Y = \left(\begin{smallmatrix} 1 & x^T \\ x & X \end{smallmatrix}\right)$, satisfying $X_{ii} = 1$ $(i = 1, \dots, d)$ and $\mathrm{Tr}(A^T A X) = b^T b$, is a natural semidefinite relaxation for P which is contained in K. (This relaxation can be obtained by taking the dual of the Lagrange dual of the formulation: $Ax = b$, $x_i^2 = 1$ $(i = 1, \dots, d)$, and $(Ax - b)^T (Ax - b) = 0$; cf. [24], [21]).

PROPOSITION 2.4. $N_+(K) \subseteq \hat{K}$.

Proof. Let $x \in N_+(K)$ and $Y \in M_+(K)$ such that $(1, x) = Y e_0$. Then $Y(e_0 \pm e_i) \in \widetilde{K}$, which means that $Ax = b$ and $AXe_i = bx_i$ $(i = 1, \dots, d)$ (setting $X := (Y_{i,j})_{i,j=1}^d$). Since $b = Ax = \sum_{i=1}^d (Ae_i) x_i$, then $\mathrm{Tr}(A^T A X) - b^T b = \sum_{i=1}^d (Ae_i)^T A X e_i - \sum_{i=1}^d (Ae_i)^T b x_i = \sum_{i=1}^d (Ae_i)^T (AXe_i - bx_i) = 0$, implying $x \in \hat{K}$. □

3. The cut polytope and some relaxations.

3.1. The cut polytope and the metric polytope. Given an integer $n \geq 3$, set $V_n := \{1, \dots, n\}$, $E_n := \{ij \mid 1 \leq i < j \in V_n\}$, and $d_n := |E_n| = \binom{n}{2}$. Let $\mathcal{S}_n$ denote the set of $n \times n$ symmetric matrices. For $X \in \mathcal{S}_n$, $X \succeq 0$ means that X is positive semidefinite (abbreviated as *sdp*). Set

$$\mathcal{S}_n^1 := \{X \in \mathcal{S}_n \mid x_{ii} = 1 \ \text{ for all } i \in V_n\}, \qquad \mathcal{E}_n := \{X \in \mathcal{S}_n^1 \mid X \succeq 0\}.$$

Given a vector $x \in \mathbf{R}^{E_n}$, let $\mathrm{smat}(x)$ denote the matrix $X \in \mathcal{S}_n^1$ whose off-diagonal entries are given by x; conversely, given a symmetric matrix $X = (x_{ij})_{i,j=1}^n$, $\mathrm{svec}(X) := (x_{ij})_{1 \leq i < j \leq n}$ denotes the vector consisting of the upper triangular entries of X. Hence, smat and svec are inverse bijections between the sets $\mathbf{R}^{E_n}$ and $\mathcal{S}_n^1$.

Given $x \in \{\pm 1\}^n$, xx^T is called a *cut matrix* and $\mathrm{svec}(xx^T) \in \mathbf{R}^{E_n}$ is the associated *cut vector* of the complete graph $K_n = (V_n, E_n)$. Thus, $\mathrm{svec}(xx^T)$ is the (± 1)-incidence vector of the *cut* $\delta(S) := \{ij \in E_n \mid |S \cap \{i, j\}| = 1\}$, where $S := \{i \mid x_i = 1\}$.

Let $G = (V_n, E)$ be a graph where $E \subseteq E_n$. The cut polytope $\mathrm{CUT}(K_n)$ of the complete graph K_n is defined as the convex hull of the cut vectors $\mathrm{svec}(xx^T)$ for $x \in \{\pm 1\}^n$, and the cut polytope $\mathrm{CUT}(G)$ of G is then defined as the projection of

$\mathrm{CUT}(K_n)$ on the subspace $\mathbf{R}^E$ indexed by the edge set of G. As linear programming formulation for $\mathrm{CUT}(G)$ we consider the *metric polytope* $\mathrm{MET}(G)$ defined by the conditions $x \in [-1,1]^E$ and the *circuit inequalities*:

$$\sum_{ij\in D} x_{ij} - \sum_{ij\in C\setminus D} x_{ij} \geq 2 - |C| \tag{3.1}$$

for all circuits C of G and all subsets $D \subseteq C$ with $|D|$ odd. It is known that $\mathrm{CUT}(G) = \mathrm{MET}(G)$ if and only if G has no K_5-minor [7]. In the linear description of $\mathrm{MET}(G)$, it suffices to consider the circuit inequalities for *chordless* circuits [7]. Therefore, $\mathrm{MET}(K_n)$ is defined by the $4\binom{n}{3}$ *triangle inequalities*:

$$x_{ij} + x_{ik} + x_{jk} \geq -1, \qquad x_{ij} - x_{ik} - x_{jk} \geq -1 \tag{3.2}$$

for all distinct $i,j,k \in V_n$. The polytope $\mathrm{MET}(G)$ coincides with the projection of $\mathrm{MET}(K_n)$ on the subspace $\mathbf{R}^E$ [6]; therefore, one can optimize a linear objective function over $\mathrm{MET}(G)$ in polynomial time and thus solve the separation problem for $\mathrm{MET}(G)$ in polynomial time. For a direct proof of the latter fact, see [7].

3.2. Semidefinite relaxations. We present here a number of semidefinite relaxations for the cut polytope.

The basic sdp relaxation As every cut matrix xx^T ($x \in \{\pm 1\}^n$) belongs to $\mathcal{E}_n$, we have

$$\mathrm{smat}(\mathrm{CUT}(K_n)) \subseteq \mathcal{E}_n.$$

The set $\mathcal{E}_n$ is the basic semidefinite relaxation of the cut polytope underlying the approximative algorithm for max-cut of Goemans and Williamson [13].

The Anjos–Wolkowicz sdp relaxation. In what follows, matrices in $\mathcal{S}_{d_n+1}$ or $\mathcal{E}_{d_n+1}$ are assumed to be indexed by the set $E_n \cup \{0\}$, and e_0, e_{ij} ($ij \in E_n$) denote the standard unit vectors in $\mathbf{R}^{d_n+1}$. For $x \in \{\pm 1\}^n$, let $y := (1, \mathrm{svec}(xx^T))$ be the associated cut vector in $\widetilde{\mathrm{CUT}}(K_n)$ and set $Y := yy^T$. Then $\mathrm{svec}(xx^T) = (Y_{0,ij})_{ij\in E_n}$. Moreover, Y belongs to $\mathcal{E}_{d_n+1}$ and satisfies the equations

$$Y_{ik,jk} = Y_{0,ij} \quad \text{for all distinct } i,j,k \in V_n, \tag{3.3}$$

$$Y_{ij,hk} = Y_{ih,jk} = Y_{ik,jh} \quad \text{for all distinct } i,j,h,k \in V_n. \tag{3.4}$$

Anjos and Wolkowicz [2] used condition (3.3) for defining the following sets $\mathcal{F}_n$ and F_n:

$$\mathcal{F}_n := \{Y \in \mathcal{E}_{d_n+1} \mid Y \text{ satisfies (3.3)}\}, \qquad F_n := \{(Y_{0,ij})_{ij\in E_n} \mid Y \in \mathcal{F}_n\}.$$

The set $\mathcal{F}_n$ is obviously contained in the set $\mathcal{G}_n$ of matrices $Y \in \mathcal{E}_{d_n+1}$ satisfying

$$Y_{0,ij} = \frac{1}{n-2} \sum_{k\in V_n,\ k\neq i,j} Y_{ik,jk} \ (ij \in E_n);$$

the relaxation $\mathcal{G}_n$ is introduced in [2] as bidual (dual of the Lagrange dual) of some formulation of max-cut.

PROPOSITION 3.1 (see [2]). $\mathrm{CUT}(K_n) \subseteq F_n \subseteq \mathrm{MET}(K_n) \cap \mathrm{svec}(\mathcal{E}_n)$.

Proof. The inclusion $\mathrm{CUT}(K_n) \subseteq F_n$ has already been observed above. The inclusion $F_n \subseteq \mathrm{svec}(\mathcal{E}_n) \cap \mathrm{MET}(K_n)$ can be verified as follows. For $Y \in \mathcal{F}_n$, set $y := (Y_{0,ij})_{ij \in E_n}$ and $X := \mathrm{smat}(y)$. By the relation (3.3), the matrix X coincides with the principal submatrix of Y with row and column indices in the set $\{0, 12, \ldots, 1n\}$. Therefore $X \in \mathcal{E}_n$, and thus $y \in \mathrm{svec}(\mathcal{E}_n)$. In order to show the triangle inequality $y_{12} + y_{13} + y_{23} \geq -1$, consider the principal submatrix Z of Y indexed by the set $\{0, 12, 13, 23\}$ and let σ denote the sum of the entries of Z. As $Z \succeq 0$, we have $\sigma \geq 0$, which implies that $y_{12} + y_{13} + y_{23} \geq -1$. The other triangle inequalities follow by the same argument after suitably flipping signs in Z. □

For $n \leq 4$, equality $\mathrm{MET}(K_n) = \mathrm{CUT}(K_n)$ holds. It is shown in [2] that both inclusions in Proposition 3.1 are strict for $n \geq 5$; for instance, the minimum of the linear objective function $\sum_{ij \in E_5} x_{ij}$ over $\mathrm{CUT}(K_5)$ is -2, while its minimum over F_5 is -2.5.

New sdp relaxations based on the LS procedure. If we apply the LS construction to the cut polytope $\mathrm{CUT}(G)$ starting with its linear relaxation by the metric polytope $\mathrm{MET}(G)$, we obtain the relaxations $N(\mathrm{MET}(G))$, $N_+(\mathrm{MET}(G))$, $N'(\mathrm{MET}(G))$, and $N'_+(\mathrm{MET}(G))$ satisfying the hierarchy (2.3).

As $G = (V_n, E)$ is a subgraph of the complete graph $K_n = (V_n, E_n)$, we have that $\mathrm{CUT}(G) = \pi_E(\mathrm{CUT}(K_n))$ and $\mathrm{MET}(G) = \pi_E(\mathrm{MET}(K_n))$, where $\pi_E : \mathbf{R}^{E_n} \longrightarrow \mathbf{R}^E$ denotes the projection onto the subspace indexed by the edge set of G. Let ν stand for one of the operators N, N_+, N', or N'_+ and let μ denote the corresponding operator M, M_+, M', M'_+ (i.e., $\mu = M$ if $\nu = N$, etc.). Taking projections at both sides of the inclusion $\mathrm{CUT}(K_n) \subseteq \nu(\mathrm{MET}(K_n))$, we obtain

$$\mathrm{CUT}(G) \subseteq \pi_E(\nu(\mathrm{MET}(K_n))).$$

LEMMA 3.2. $\pi_E(\nu(\mathrm{MET}(K_n))) \subseteq \nu(\mathrm{MET}(G))$.

Proof. Let $y \in \pi_E(\nu(\mathrm{MET}(K_n)))$. Then $(1, y) = \pi_E(Ye_0)$, where $Y \in \mu(\mathrm{MET}(K_n))$. Let X denote the principal submatrix of Y indexed by the set $\{0\} \cup E$. Then $X \in \mu(\mathrm{MET}(G))$. (This follows from the fact that each column of X is the projection on $\mathbf{R}^{\{0\} \cup E}$ of the corresponding column of Y and $\mathrm{MET}(G) = \pi_E(\mathrm{MET}(K_n))$.) Therefore $y = ((Xe_0)_f)_{f \in E}$ belongs to $\nu(\mathrm{MET}(G))$. □

Equality holds obviously in the inclusion of Lemma 3.2 when $G = K_n$. We do not know whether equality holds in general, i.e., whether the two operators ν and π_E commute. Note that not every matrix $Y \in M(\mathrm{MET}(G))$ can be extended to a matrix of $M(\mathrm{MET}(K_n))$; for example, the matrix

$$Y := \begin{array}{c} \\ 0 \\ 12 \\ 23 \\ 34 \\ 14 \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 12 & 23 & 34 & 14 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

belongs to $M(\mathrm{MET}(G))$, where G is the circuit $(1, 2, 3, 4)$, but Y cannot be extended to a matrix of $M(\mathrm{MET}(K_4))$ (because $Y_{12,23} \neq Y_{14,34}$; cf. Proposition 3.4(i) below). For simplicity in the notation, we set

$$\nu(G) := \pi_E(\nu(\mathrm{MET}(G))).$$

Iterates are defined in the obvious manner: $\nu^k(G) := \pi_E(\nu^k(\mathrm{MET}(K_n)))$. The inclusion from Lemma 3.2 will be extended to higher iterates in Corollary 4.13.

It seems preferable to work with the relaxation $\nu(G)$ rather than $\nu(\mathrm{MET}(G))$, as it provides a better relaxation for $\mathrm{CUT}(G)$. Moreover, one can optimize a linear objective function over $\nu(G)$ in polynomial time for any graph and $\nu = N, \ldots, N'_+$. In contrast, this is true for $\nu(\mathrm{MET}(G))$ for any graph G if $\nu = N, N_+$ and, if $\nu = N', N'_+$, for any graph G for which the list of circuit inequalities (3.1) (for chordless circuits) has a polynomial length (thus, for instance, if G is a complete graph or more generally a chordal graph). One more attractive feature of the relaxation $\nu(G)$ is that the class of graphs G for which $\mathrm{CUT}(G) = \nu(G)$ is well behaved; e.g., it is closed under taking deletion minors while it is not clear whether this property holds for the relaxation $\nu(\mathrm{MET}(G))$ (cf. section 4.4). On the other hand, it will be convenient to work with the relaxation $\nu(\mathrm{MET}(G))$ in order to establish results about valid inequalities (cf. section 4.2).

Permutation and switching. Every permutation σ acts in a natural way on an $n \times n$ symmetric matrix X and on a vector $x \in \mathbf{R}^{E_n}$, producing the vector $x^\sigma := (x_{\sigma(i)\sigma(j)})_{ij \in E_n}$. As σ induces a permutation of E_n, it also acts on a matrix $Y \in \mathcal{S}_{d_n+1}$, producing the matrix $Y^\sigma \in \mathcal{S}_{d_n+1}$ defined by

$$Y^\sigma_{0,ij} := Y_{0,\sigma(i)\sigma(j)}, \quad Y^\sigma_{ij,rs} := Y_{\sigma(i)\sigma(j),\sigma(r)\sigma(s)} \quad \text{for } ij, rs \in E_n. \tag{3.5}$$

Permutation preserves the cut polytope of the complete graph K_n and all its relaxations considered in the paper.

Given a subset $S \subseteq V_n$ and $X \in \mathcal{S}_n$, let X^S denote the matrix obtained from X by changing the signs of its rows and columns indexed by S; in other words, one switches the signs of the entries of X indexed by edges in the cut $\delta(S)$. Switching extends naturally to matrices $Y \in \mathcal{S}_{d_n+1}$ and produces $Y^{\delta(S)}$ obtained from Y by changing signs of its rows and columns indexed by the set $\delta(S)$. Switching also applies to vectors $x \in \mathbf{R}^E$ ($E \subseteq E_n$): simply change the signs of the entries of x indexed by the set $\delta(S) \cap E$.

Clearly, $X^S \in \mathrm{smat}(\mathrm{CUT}(K_n))$ (resp., $X^S \in \mathcal{E}_n$) if and only if $X \in \mathrm{smat}(\mathrm{CUT}(K_n))$ (resp., $X \in \mathcal{E}_n$). For $X, Y \in \mathcal{S}_n$, one has $\langle X, Y\rangle = \langle X^S, Y^S\rangle$. (Here $\langle X, Y\rangle = \sum_{i,j=1}^n x_{ij}y_{ij}$ denotes the usual inner product in $\mathcal{S}_n$.) Therefore, if an inequality $\langle A, X\rangle \geq \beta$ is valid for $\mathrm{smat}(\mathrm{CUT}(K_n))$, its switching $\langle A^S, X\rangle \geq \beta$ remains valid for $\mathrm{smat}(\mathrm{CUT}(K_n))$. Note that the classes of triangle inequalities and of circuit inequalities are closed under switching. Switching preserves all the relaxations of the cut polytope considered in the paper.

3.3. Basic properties of the new relaxations. The following is an easy but important property of the metric polytope that will be repeatedly used in this paper.

PROPOSITION 3.3. *If $y \in \mathrm{MET}(G)$ satisfies $y_{uv} = \epsilon$ for some edge $uv \in E$ and $\epsilon \in \{\pm 1\}$, then*

$$y_{ui} = \epsilon y_{vi} \quad \text{for every node } i \text{ adjacent to both } u \text{ and } v. \tag{3.6}$$

Proof. Apply the triangle inequalities (3.2) to the triple uvi. □

As a first application, we find that (3.3) and (3.4) are valid for $M(\mathrm{MET}(G))$ and $M'(\mathrm{MET}(G))$, respectively.

PROPOSITION 3.4.

(i) *If $Y \in M(\mathrm{MET}(G))$, then $Y_{ik,jk} = Y_{0,ij}$ for all distinct pairwise adjacent $i, j, k \in V_n$.*

(ii) *If $Y \in M'(\mathrm{MET}(G))$, then $Y_{ij,hk} = Y_{ih,jk} = Y_{ik,jh}$ for all distinct pairwise adjacent $i, j, h, k \in V_n$.*

Proof. (i) Let $1,2,3$ be pairwise adjacent nodes and $Y \in M(\mathrm{MET}(G))$. By assumption, the vector $y := Y(e_0 - e_{12})$ belongs to $\widetilde{\mathrm{MET}(G)}$. As $y_0 = -y_{12}$, we have from (3.6) that $y_{13} = -y_{23}$, which implies

$$Y_{0,13} + Y_{0,23} = Y_{12,13} + Y_{12,23}.$$

Similarly, using the fact that $Y(e_0 - e_{13}), Y(e_0 - e_{23}) \in \widetilde{\mathrm{MET}(G)}$, we obtain

$$Y_{0,12} + Y_{0,23} = Y_{13,12} + Y_{13,23} \quad \text{and} \quad Y_{0,12} + Y_{0,13} = Y_{23,12} + Y_{23,13}.$$

From this it follows that $Y_{0,12} = Y_{23,13}$, which shows (i).

(ii) Let $1,2,3,4$ be pairwise adjacent nodes in G and $Y \in M'(\mathrm{MET}(G))$. By assumption, the vector $y := Y(e_0 + e_{12} + e_{13} + e_{23})$ belongs to $\widetilde{\mathrm{MET}(G)}$ and thus satisfies the triangle inequalities $-y_{12} + y_{14} - y_{24} \geq -y_0$ and $-y_{12} - y_{14} + y_{24} \geq -y_0$. Using the above result (i), we find that $y_{12} = y_0$. Now (3.6) implies that $y_{14} = y_{24}$, which, using (i) again, yields $Y_{14,23} = Y_{13,24}$. □

COROLLARY 3.5. $N_+(K_n) \subseteq F_n$.

We will see later that $N_+(K_5) = \mathrm{CUT}(K_5)$; therefore, the inclusion $N_+(K_n) \subseteq F_n$ is strict for $n \geq 5$.

4. The index of a graph. The *N-index* $\eta_N(G)$ of a graph G is defined as the smallest integer k for which $\mathrm{CUT}(G) = N^k(\mathrm{MET}(G))$, and its *projected N-index* $\eta_N^\pi(G)$ is the smallest k for which $\mathrm{CUT}(G) = N^k(G)$; the indexes η_ν and η_ν^π are defined analogously with respect to the other operators $\nu = N_+$, N', or N'_+. Obviously, $\eta_\nu^\pi(G) \leq \eta_\nu(G)$. By Theorem 2.1, the N-index of G is bounded by the number of edges of G; in section 4.1, we show some sharper upper bounds which, in fact, remain valid for the N_0-index since they are obtained using Lemma 2.2. In particular, we show that $\eta_N(G) \leq n - 4$ for a graph G on $n \geq 4$ nodes, and in section 4.3 we prove the upper bound $n - 5$ for the N'-index of a graph on $n \geq 6$ nodes. In section 4.4, we study how the index of a graph behaves with respect to the graph operations of taking minors and clique sums. Section 4.2 contains some technical results needed for establishing the upper bounds on the N'-index and for proving the minor monotonicity of the index of a graph.

4.1. Upper bounds for the N-index of a graph. Let $G = (V_n, E)$ be a graph. We show here a linear upper bound in $O(n)$ for the N-index of G (in place of the bound $|E|$). The basic idea is to use Lemma 2.2 and to reformulate the validity of an inequality $a^T x \geq \beta$ for $\mathrm{MET}(G) \cap \{x \mid x_{uv} = \epsilon\}$ in terms of the validity of a transformed inequality for $\mathrm{MET}(G/uv)$, the metric polytope of the contracted graph G/uv.

We need some definitions. For $u \in V_n$, $N_G(u)$ denotes the set of nodes adjacent to u in G. Given an edge $uv \in E$, let $H := G/uv$ denote the graph obtained from G by contracting uv; its node set is $V_n \setminus \{u,v\} \cup \{w\}$, where w is the new node created by contraction of edge uv, and we denote by F its edge set (multiple edges are erased). Clearly F is in bijection with the subset $\hat{F} := \{\hat{f} \mid f \in F\}$ of E where, for $f \in F$,

$$\begin{aligned} &\hat{f} := f \text{ if } w \notin f, \qquad \hat{f} := ui \text{ if } f = wi \text{ with } i \in N_G(u), \\ &\hat{f} := vi \text{ if } f = wi \text{ with } i \in N_G(v) \setminus N_G(u). \end{aligned} \tag{4.1}$$

Given $y \in \mathbf{R}^E$ satisfying $y_{uv} = \epsilon \in \{\pm 1\}$ and (3.6), its *ϵ-restriction* $y^{F,\epsilon} \in \mathbf{R}^F$ is defined by

$$y_f^{F,\epsilon} := y_{\hat{f}} \text{ for all } f \in F \text{ except } y_{wi}^{F,\epsilon} := \epsilon y_{vi} \text{ for } i \in N_G(v) \setminus N_G(u). \tag{4.2}$$

Conversely, relation (4.2) permits us to define for any vector $x \in \mathbf{R}^F$ its *ϵ-extension* $y \in \mathbf{R}^E$ in such a way that $y_{uv} = \epsilon$ and $y^{F,\epsilon} = x$. Note that for $\epsilon = -1$, $y^{F,-1}$ coincides with the 1-restriction of the vector y' obtained from y by switching the signs of its entries indexed by edges in the cut $\delta(v)$. Our objective is to show that membership of y in some iterate $\nu^k(\mathrm{MET}(G))$ is equivalent to membership of its ϵ-restriction in the corresponding iterate $\nu^k(\mathrm{MET}(G/uv))$ of the contracted graph (ν being any of the operators $N, \ldots, N'_+$). We treat here the case $k = 0$, and the general case will be treated in the next subsection. It will be convenient to use the following correspondence between the circuits of G and those of $H = G/uv$:

(4.3) To any circuit C of H there corresponds a circuit C' of G, where $C' := \hat{C} \cup \{uv\}$ if $w \in C$ and its neighbors a, b on C satisfy $a \in N_G(u)$, $b \in N_G(v) \setminus N_G(u)$, and $C' := \hat{C}$ otherwise

(setting $\hat{C} := \{\hat{f} \mid f \in C\}$, where $\hat{f}$ is defined by (4.1)).

LEMMA 4.1. *Let $x \in \mathbf{R}^F$ and let $y \in \mathbf{R}^E$ be its ϵ-extension, where $\epsilon = \pm 1$. Then*

(i) $x \in \mathrm{MET}(G/uv) \iff y \in \mathrm{MET}(G)$,

(ii) $x \in \mathrm{CUT}(G/uv) \iff y \in \mathrm{CUT}(G)$.

Proof. (i) We let $\epsilon = 1$, as the case $\epsilon = -1$ can be derived from it by applying switching. Obviously, $y \in [-1,1]^E$ if and only if $x \in [-1,1]^F$. Suppose first that $y \in \mathrm{MET}(G)$; we show that $x \in \mathrm{MET}(H)$. For this let C be a circuit in H and let $D \subseteq C$ be a subset of odd cardinality; we show that $x(D) - x(C \setminus D) \geq 2 - |C|$. Let $\hat{D} := \{\hat{f} \mid f \in D\}$ and let C' be the circuit in G derived from C as indicated in (4.3). Then, $x(D) - x(C \setminus D) = y(\hat{D}) - y(\hat{C} \setminus \hat{D})$. If $C' = \hat{C}$, then $y(\hat{D}) - y(\hat{C} \setminus \hat{D}) \geq 2 - |C'| = 2 - |C|$; if $C' = \hat{C} \cup \{uv\}$, then $y(\hat{D}) - y(\hat{C} \setminus \hat{D}) \geq 2 - |C'| + y_{uv} = 2 - |C|$, using the assumption $y_{uv} = 1$. We omit the proof for the reverse implication which is similar. Assertion (ii) follows from the fact that the extension/restriction operation maps the cut vectors of H to cut vectors of G. ∎

Given $a \in \mathbf{R}^E$ and $\epsilon = \pm 1$, let $a_\epsilon \in \mathbf{R}^F$ be defined by

(4.4)
$$(a_\epsilon)_{wi} := a_{ui} \text{ for } i \in N_G(u) \setminus N_G(v), \quad (a_\epsilon)_{wi} := \epsilon a_{vi} \text{ for } i \in N_G(v) \setminus N_G(u),$$
$$(a_\epsilon)_{wi} := a_{ui} + \epsilon a_{vi} \text{ for } i \in N_G(u) \cap N_G(v), \quad (a_\epsilon)_{ij} := a_{ij} \text{ for } ij \in E, i, j \neq u, v.$$

It follows from these definitions that

(4.5) $$a^T y = a_\epsilon^T x + \epsilon a_{uv} \quad \text{for } x \in \mathbf{R}^F \text{ and its } \epsilon\text{-extension } y \in \mathbf{R}^E.$$

LEMMA 4.2. *Let $a \in \mathbf{R}^E$, $\epsilon \in \{\pm 1\}$, $a_\epsilon \in \mathbf{R}^F$ as in (4.4), and $\beta \in \mathbf{R}$ be given. Then*

$$a^T y \geq \beta \text{ is valid for } \mathrm{MET}(G) \cap \{y \mid y_{uv} = \epsilon\}$$
$$\iff a_\epsilon^T x \geq \beta - \epsilon\, a_{uv} \text{ is valid for } \mathrm{MET}(G/uv),$$
$$a^T y \geq \beta \text{ is valid for } \mathrm{CUT}(G) \cap \{y \mid y_{uv} = \epsilon\}$$
$$\iff a_\epsilon^T x \geq \beta - \epsilon\, a_{uv} \text{ is valid for } \mathrm{CUT}(G/uv).$$

Proof. Apply Lemma 4.1 and (4.5). ∎

THEOREM 4.3. *Let G be a graph and $e_1, \ldots, e_k$ be distinct edges in G. Then*

$$\mathrm{CUT}(G) = \mathrm{conv}(\mathrm{MET}(G) \cap \{x \mid x_{e_1}, \ldots, x_{e_k} = \pm 1\})$$

if and only if the graph $G/\{e_1,\dots,e_k\}$ has no K_5-minor.

Proof. The proof is by induction on $k \geq 0$. The result holds for $k = 0$ since it is shown in [7] that $\mathrm{CUT}(G) = \mathrm{MET}(G)$ if and only if G has no K_5-minor. Let $k \geq 1$ and suppose that the result from Theorem 4.3 holds for $k-1$; we show that it also holds for k. Applying the induction assumption to the graph G/e_k, we obtain that $\mathrm{CUT}(G/e_k) = \mathrm{conv}(\mathrm{MET}(G/e_k) \cap \{x \mid x_{e_1},\dots,x_{e_{k-1}} = \pm 1\})$ if and only if $G/\{e_1,\dots,e_k\}$ has no K_5-minor. Therefore, it remains to show that the two statements

$$\mathrm{CUT}(G/e_k) = \mathrm{conv}(\mathrm{MET}(G/e_k) \cap \{x \mid x_{e_1},\dots,x_{e_{k-1}} = \pm 1\}),$$
$$\mathrm{CUT}(G) = \mathrm{conv}(\mathrm{MET}(G) \cap \{x \mid x_{e_1},\dots,x_{e_k} = \pm 1\})$$

are equivalent, which is a simple verification using Lemma 4.1. □

COROLLARY 4.4. *If a graph G has a set of k edges whose contraction produces a graph with no K_5-minor, then $\mathrm{CUT}(G) = N_0^k(G) = N^k(G)$. In particular, $\mathrm{CUT}(G) = N^{n-4}(\mathrm{MET}(G))$ if G has $n \geq 4$ nodes.*

Proof. The first statement is a direct application of Theorem 4.3 and (2.6), (2.7). We now show that in a graph G on n nodes there exist at most $n-4$ edges whose contraction produces a graph with no K_5-minor. If G is connected, let T be a spanning tree in G and let $u, v, w \in V_n$ for which $T' := T \setminus \{u, v, w\}$ is still a tree. (Such nodes can be easily found if T is a path, and otherwise choose three leaves of T.) Then the graph obtained from G by contracting the $n-4$ edges of T' has no K_5-minor. If G is not connected, apply the same reasoning to each connected component of G. □

Given an integer $r \geq 1$, let $\alpha_r(G)$ denote the maximum cardinality of a subset $S \subseteq V_n$ for which the induced subgraph $G[S]$ has no K_{r+1} minor; thus $\alpha_1(G)$ is the stability number $\alpha(G)$ of G, and $\alpha_{r+1}(G) \geq \alpha_r(G) + 1$ if $\alpha_r(G) \leq n-1$. As a consequence of Corollary 4.4, we can show the following.

COROLLARY 4.5. *Let $r \in \{1, 2, 3\}$ and $G = (V_n, E)$ be a graph on n nodes. Then,*

$$\eta_N^\pi(G) \leq \max(0, n - \alpha_r(G) + r - 4). \tag{4.6}$$

If there exists a subset $S \subseteq V_n$ for which $G[S]$ has no K_{r+1} minor, $G[V_n \setminus S]$ has at most $4-r$ connected components, and $|S| = \alpha_r(G)$, then

$$\eta_N(G) \leq \max(0, n - \alpha_r(G) + r - 4). \tag{4.7}$$

Proof. We use the following observation: The graph G^*, obtained from $G[S]$ by adding to it $4-r$ pairwise adjacent nodes that are adjacent to all nodes of S, has no K_5-minor, and thus the same holds for any subgraph of G^*. We first verify that (4.7) holds. For this, suppose that $S \subseteq V_n$ with $|S| = \alpha_r(G)$, $G[S]$ has no K_{r+1} minor, and $G[V_n \setminus S]$ has at most $4-r$ connected components; we show that a graph with no K_5-minor can be obtained from G by contracting at most $k_r := \max(0, n - \alpha_r(G) + r - 4)$ edges. Indeed, using the assumption that $G[V_n \setminus S]$ has at most $4-r$ components, one can find at most k_r edges in $G[V_n \setminus S]$ whose contraction transforms $G[V_n \setminus S]$ into a graph on at most $4-r$ nodes. We now verify (4.6). If $G[V_n \setminus S]$ has t components, let G' be the graph obtained from G by adding $t-1$ edges between the components of $G[V_n \setminus S]$ so as to make $G'[V_n \setminus S]$ connected. We just saw that $\eta_N(G') \leq k_r$ and thus $\mathrm{CUT}(G') = N^{k_r}(G')$. By projecting out the added edges, we obtain that $\mathrm{CUT}(G) = N^{k_r}(G)$, that is, $\eta_N^\pi(G) \leq k_r$. □

In particular, the N-index of the graph G^∇, obtained from G by adding a new node adjacent to all nodes of G, is at most $n - \alpha(G) - 2$. Some rationale for the similarity between this upper bound and the known upper bound $n - \alpha(G) - 1$ for the N-index of the stable set polytope of G will be given in section 6.

Consider, for example, the complete bipartite graph $K_{4,5}$: then $\eta_N^\pi(K_{4,5}) = 1$ (by (4.6)) but the upper bound from (4.7) does not apply (since the complement of a maximum stable set induces a graph with four connected components). It would be interesting to determine whether $\eta_N(K_{4,5}) = 1$. If not, then $K_{4,5}$ would be an example of a graph for which the inclusion $N(G) \subseteq N(\mathrm{MET}(G))$ is strict; moreover, this would show that the N-index is not monotone with respect to deletion of edges, since the N-index of the graph obtained from $K_{4,5}$ by adding one edge is equal to 1.

As another consequence of Corollary 4.4, we have found a compact representation for the cut polytope of a graph having k edges whose contraction produces a graph with no K_5-minor. Therefore, the max-cut problem can be solved in polynomial time for such graphs (for fixed k). This result can, however, be checked directly using a branching strategy. For instance, if G/uv has no K_5-minor and one wishes to find the maximum weight W of a cut in G with respect to some weight function a, then $W = \max(W_1, W_{-1} + a(\delta_G(v)))$, where, for $\epsilon = \pm 1$, W_ϵ is the maximum weight of a cut in G/uv with respect to the weight function a_ϵ (defined as in (4.4)). (This idea is also present, e.g., in [23].)

4.2. Validity for the new relaxations via contraction. We saw in Lemma 4.2 that the validity of an inequality $a^T x \geq \beta$ for $\mathrm{MET}(G) \cap \{x \mid x_{uv} = \epsilon\}$ can be reformulated in terms of the validity of the transformed inequality $a_\epsilon^T x \geq \beta - \epsilon\, a_{uv}$ for $\mathrm{MET}(G/uv)$. We here extend this result for any iterate $\nu^k(\mathrm{MET}(G))$, where $\nu = N, \ldots, N'_+$ and $k \geq 1$. For this we need to extend the notions of ϵ-extension and restriction to matrices. We begin with an application of (3.6) to matrices in $M(\mathrm{MET}(G))$.

PROPOSITION 4.6. *Let $Y \in M(\mathrm{MET}(G))$ and assume that $Y_{0,uv} = \epsilon Y_{0,0}$ for some edge $uv \in E$ and $\epsilon = \pm 1$. Then Y satisfies*

$$Ye_0 = \epsilon Y e_{uv},\ Ye_{ui} = \epsilon Y e_{vi} \quad \text{for every node } i \in N_G(u) \cap N_G(v); \tag{4.8}$$

that is, Y has the following block decomposition:

$$Y = \begin{array}{c} \\ I \\ K \\ J \end{array} \begin{array}{c} \begin{array}{ccc} I & K & J \end{array} \\ \begin{pmatrix} A & B^T & \epsilon A \\ B & C & \epsilon B \\ \epsilon A & \epsilon B^T & A \end{pmatrix} \end{array}, \tag{4.9}$$

setting $I := \{0\} \cup \{ui \mid i \in N_G(u) \cap N_G(v)\}$, $J := \{uv\} \cup \{vi \mid i \in N_G(u) \cap N_G(v)\}$, and $K := E \setminus (I \cup J)$.

Proof. As $y := Y(e_0 - \epsilon e_{uv}) \in \widetilde{\mathrm{MET}(G)}$ with $y_0 = 0$, we have that $-y_0 \leq y_f \leq y_0$, which yields $y_f = 0$ for all $f \in E$, and thus $Ye_0 = \epsilon Y e_{uv}$. Let i be a node adjacent to both u and v. As $x := Ye_0 \in \widetilde{\mathrm{MET}(G)}$ with $x_0 = \epsilon x_{uv}$, we have from (3.6) that $x_{ui} = \epsilon x_{vi}$, i.e., $Y_{0,ui} = \epsilon Y_{0,vi}$. Given $f \in E$, set $z := Y(e_0 - e_f)$; then $z \in \widetilde{\mathrm{MET}(G)}$ and $z_0 = \epsilon z_{uv}$ by the above. By (3.6) this implies that $z_{ui} = \epsilon z_{vi}$ and thus $Y_{ui,f} = \epsilon Y_{vi,f}$. This shows that $Ye_{ui} = \epsilon Y e_{vi}$. ∎

Let Y be a symmetric matrix indexed by $\{0\} \cup E$ and satisfying (4.8) for some $\epsilon = \pm 1$ and $uv \in E$; then, Y has the form (4.9). We define its *ϵ-restriction* $Y^{F,\epsilon}$ in the following manner: If $\epsilon = 1$, then $Y^{F,1}$ is the principal submatrix $\left(\begin{smallmatrix} A & B^T \\ B & C \end{smallmatrix}\right)$ of Y indexed by the subset $\{0\} \cup \hat{F} = I \cup K$. If $\epsilon = -1$, let Y' be the matrix obtained from Y by switching the signs of its rows/columns indexed by edges in the cut $\delta(v)$; then $Y^{F,-1}$ is the principal submatrix of Y' indexed by $I \cup K$. As F is in bijection

with $\hat{F}$ we can view $Y^{F,\epsilon}$ as being indexed by $\{0\} \cup F$. Conversely, one can define the *ϵ-extension* Y of a matrix X indexed by $\{0\} \cup F$ in such a way that $Y_{0,0} = \epsilon Y_{0,uv}$ and $Y^{F,\epsilon} = X$. Clearly,

$$Y \succeq 0 \Longleftrightarrow Y^{F,\epsilon} \succeq 0. \tag{4.10}$$

Recall that the dual cone of the cone $\widetilde{\mathrm{MET}(G)}$ is spanned by the vectors $e_0 \pm e_f$ ($f \in E$) and

$$\xi^{C,D} := (|C| - 2)e_0 + \sum_{f \in D} e_f - \sum_{f \in C \setminus D} e_f$$

for all chordless circuits C of G and all odd subsets $D \subseteq C$.

PROPOSITION 4.7. *Let $k \geq 0$ be an integer, let Y be a symmetric matrix indexed by $\{0\} \cup E$ satisfying* (4.8) *for some $\epsilon = \pm 1$ and $uv \in E$, and let $Y^{F,\epsilon}$ be its ϵ-restriction. Let ν be one of the operators $N, \ldots, N'_+$ and μ the corresponding operator from $M, \ldots, M'_+$. Then*

$$Y \in \mu(\nu^k(\mathrm{MET}(G))) \Longleftrightarrow Y^{F,\epsilon} \in \mu(\nu^k(\mathrm{MET}(G/uv))).$$

Proof. Let $\epsilon = 1$, as the case $\epsilon = -1$ can be derived from it by applying switching. In view of relation (4.10) it suffices to show the result for the operators $\nu = N$, N'. The proof is by induction on $k \geq 0$ and uses Lemma 4.1 together with the following observation: For $f \in F$, the $\hat{f}$th column of Y is the 1-extension of the corresponding fth column of $Y^{F,1}$, while the remaining columns of Y are duplicates of some of those. We first consider the case $k = 0$. The statement for the case $\nu = N$ follows as a direct application of the above observation. Suppose now that $Y \in M'(\mathrm{MET}(G))$; we show that $Y^{F,1} \in M'(\mathrm{MET}(H))$. For this let C be a circuit in H and let $D \subseteq C$ with an odd cardinality; we show that $x := Y^{F,1}\xi^{C,D} \in \widetilde{\mathrm{MET}(H)}$. Set $\hat{D} := \{\hat{f} \mid f \in D\}$ and let C' be the circuit in G obtained from C as indicated in (4.3). By assumption, $y := Y\xi^{C',\hat{D}} \in \widetilde{\mathrm{MET}(G)}$ and $y_0 = y_{uv}$. Thus, by Lemma 4.1, its 1-restriction $y^{F,1}$ belongs to $\widetilde{\mathrm{MET}(H)}$. It suffices now to observe that $Y^{F,1}\xi^{C,D}$ coincides with $y^{F,1}$ (using the fact that $Ye_0 = Ye_{uv}$ in the case in which $C' = \hat{C} \cup \{uv\}$). The proof for the implication $Y^{F,1} \in M(\mathrm{MET}(H)) \Longrightarrow Y \in M(\mathrm{MET}(G))$ is analogous and thus omitted.

Let $k \geq 1$ and suppose that the result from Proposition 4.7 holds for $k - 1$; we show that it holds for k. We treat only the case when $\nu = N$, as the proof is analogous for N'. Suppose first that $Y \in M(N^k(\mathrm{MET}(G)))$; we show that $Y^{F,1} \in M(N^k(\mathrm{MET}(H)))$. For this, let $f \in F$, $\epsilon' = \pm 1$, and $x := Y^{F,1}(e_0 + \epsilon' e_f)$; we show that $x \in N^k(\widetilde{\mathrm{MET}(H)})$. By assumption, the vector $y := Y(e_0 + \epsilon' e_{\hat{f}})$ belongs to $N^k(\widetilde{\mathrm{MET}(G)})$ and satisfies $y_0 = y_{uv}$. Hence there exists a matrix $A \in M(N^{k-1}(\mathrm{MET}(G)))$ such that $y = Ae_0$. As $A_{0,0} = A_{0,uv}$, A satisfies (4.8) by Proposition 4.6, and we deduce from the induction assumption that $A^{F,1} \in M(N^{k-1}(\mathrm{MET}(H)))$. Thus $y^{F,1} = A^{F,1}e_0$ belongs to $N^k(\widetilde{\mathrm{MET}(H)})$. The result now follows since $x = y^{F,1}$. We omit the details of the proof for the converse implication: $Y^{F,1} \in M(N^k(\mathrm{MET}(H))) \Longrightarrow Y \in M(N^k(\mathrm{MET}(G)))$. □

COROLLARY 4.8. *Let $k \geq 0$ be an integer, let $y \in \mathbf{R}^{\{0\} \cup E}$ satisfying $y_{uv} = \epsilon y_0$ and* (3.6) *for some $\epsilon = \pm 1$ and $uv \in E$, and let $y^{F,\epsilon} \in \mathbf{R}^{\{0\} \cup F}$ be its ϵ-restriction*

defined by (4.2). *Let* ν *be one of the operators* $N, \dots, N'_+$. *Then*

$$y \in \nu^k(\widetilde{\mathrm{MET}(G)}) \Longleftrightarrow y^{F,\epsilon} \in \nu^k(\widetilde{\mathrm{MET}(G/uv)}).$$

Proof. For $k = 0$ the result holds by Lemma 4.1, and for $k \geq 1$ it follows from Proposition 4.7. ∎

Relation (4.5) together with Corollary 4.8 imply the following.

PROPOSITION 4.9. *Let* $k \geq 0$ *be an integer,* $\epsilon = \pm 1$, $uv \in E$, $a \in \mathbf{R}^E$, $\beta \in \mathbf{R}$, *and* ν *one of* $N, \dots, N'_+$. *The inequality* $a^T x \geq \beta$ *is valid for* $\nu^k(\mathrm{MET}(G)) \cap \{x \mid x_{uv} = \epsilon\}$ *if and only if the inequality* $a^T_\epsilon x \geq \beta - \epsilon a_{uv}$ *is valid for* $\nu^k(\mathrm{MET}(G/uv))$.

Let us say that the inequality $a^T_\epsilon x \geq \beta - \epsilon a_{uv}$ is obtained from the inequality $a^T x \geq \beta$ by *collapsing* ($\epsilon = 1$) or *anticollapsing* ($\epsilon = -1$) nodes u and v. Recall that anticollapsing amounts to first switching the signs of entries of a indexed by the cut $\delta(v)$ and then collapsing u and v. The following reformulations of Lemmas 2.2 and 2.3 will be used later in the paper.

PROPOSITION 4.10. *Let* $\nu = N, \dots, N'_+$. *The inequality* $a^T x \geq \beta$ *is valid for* $\nu^{k+1}(\mathrm{MET}(G))$ *if there is an edge* $uv \in E$ *for which both inequalities obtained from it by collapsing and anticollapsing nodes* u *and* v *are valid for* $\nu^k(\mathrm{MET}(G/uv))$.

PROPOSITION 4.11. *Suppose that* $a_f \geq 0$ *for all* $f \in E$ *and* $\beta \leq 0$. *The inequality* $a^T x \geq \beta$ *is valid for* $N^{k+1}_+(\mathrm{MET}(G))$ *if, for every edge* $uv \in E$ *for which* $a_{uv} > 0$, *the inequality obtained from* $a^T x \geq \beta$ *by anticollapsing nodes* u *and* v *is valid for* $N^k_+(\mathrm{MET}(G/uv))$.

It is obvious that $\mathrm{CUT}(K_n)$ is equal to the projection of $\mathrm{CUT}(K_{n+1})$ on the subspace $\mathbf{R}^{E_n}$ indexed by the edge set of K_n; similarly for $\mathrm{MET}(K_n)$. The same can be verified for F_n and for any iterate $\nu^k(\mathrm{MET}(K_n))$. (In the latter case, use Corollaries 4.8 and 4.13.)

PROPOSITION 4.12. *Let* $G = (V_n, E)$ *be a graph,* $F \subseteq E$, *and* $H := (V_n, F)$ *the corresponding subgraph of* G. *Let* ν *be one of the operators* $N, \dots, N'_+$, μ *the associated operator from* $M, \dots, M'_+$, *and let* $k \geq 0$ *be an integer. If* $Y \in \mu(\nu^k(\mathrm{MET}(G)))$, *then its principal submatrix* X *indexed by the set* $\{0\} \cup F$ *belongs to* $\mu(\nu^k(\mathrm{MET}(H)))$.

Proof. It suffices to consider the case when $\nu = N, N'$ as $Y \succeq 0$ implies $X \succeq 0$. We use the following facts in the proof: $\mathrm{MET}(H)$ is the projection on $\mathbf{R}^F$ of $\mathrm{MET}(G)$; if ξ belongs to the dual cone of $\mathrm{MET}(H)$, then its extension $\xi' := (\xi, 0, \dots, 0) \in \mathbf{R}^E$ belongs to the dual of $\mathrm{MET}(G)$; and $X\xi$ is the projection on $\mathbf{R}^{\{0\}\cup F}$ of $Y\xi'$.

The proof is by induction for $k \geq 0$. The case $k = 0$ is obvious in view of the above observations. Let $k \geq 1$ and suppose that the result holds for $k - 1$. Assume that $Y \in \mu(\nu^k(\mathrm{MET}(G)))$; we show that $X \in \mu(\nu^k(\mathrm{MET}(H)))$. For this, consider $\xi \in \mathrm{MET}(H)^*$ and its extension $\xi' \in \mathrm{MET}(G)^*$. We show that $x := X\xi \in \nu^k(\widetilde{\mathrm{MET}(H)})$. By assumption, $y := Y\xi' \in \nu^k(\widetilde{\mathrm{MET}(G)})$. Therefore, $y = Ae_0$ for some $A \in \mu(\nu^{k-1}(\mathrm{MET}(G)))$. Using the induction assumption, the principal submatrix B of A indexed by $\{0\}\cup F$ belongs to $\mu(\nu^{k-1}(\mathrm{MET}(H)))$, and thus $Be_0 \in \nu^k(\widetilde{\mathrm{MET}(H)})$. Note now that x, being the projection on $\mathbf{R}^{\{0\}\cup F}$ of y, is equal to Be_0. This shows the result; indeed, for $\nu = N$, restrict the above argument to ξ of the form $e_0 \pm e_f$ ($f \in F$). ∎

COROLLARY 4.13. *Let* $G = (V_n, E)$ *be a graph,* $H = (V_n, F)$ *a subgraph of* G, π_F *the projection from* $\mathbf{R}^E$ *onto* $\mathbf{R}^F$, $k \geq 0$ *an integer, and* $\nu = N, \dots, N'_+$. *Then* $\pi_F(\nu^k(\mathrm{MET}(G))) \subseteq \nu^k(\mathrm{MET}(H))$. *In particular,* $\mathrm{CUT}(G) \subseteq \nu^k(G) \subseteq \nu^k(\mathrm{MET}(G))$.

4.3. Upper bound for the N'-index of a graph. We showed in section 4.1 the upper bound $n - 4$ for the N-index of a graph on $n \geq 4$ nodes. We will see in

section 5 that

$$\eta_N(K_6) = \eta_{N_+}(K_6) = 2, \quad \eta_{N'}(K_6) = 1, \quad \eta_{N'_+}(K_n) \geq 2 \quad \text{for } n \geq 7.$$

Thus $\eta_{N'}(G) \leq 1$ for a graph on $n \leq 6$ nodes. Based on this fact, one can show the slightly better upper bound $n-5$ for the N'-index of a graph on $n \geq 6$ nodes.

THEOREM 4.14. *Let ν be one of $N, \ldots, N'_+$ and let $h, k \geq 0$ be integers. If there exist k edges $e_1, \ldots, e_k$ in G for which* $\mathrm{CUT}(G/\{e_1, \ldots, e_k\}) = \nu^h(\mathrm{MET}(G/\{e_1, \ldots, e_k\}))$, *then* $\mathrm{CUT}(G) = \nu^{h+k}(\mathrm{MET}(G))$.

Proof. The proof is by induction for $k \geq 0$. The result holds trivially for $k = 0$. Let $k \geq 1$ and suppose that the result holds for $k-1$. Let $a^T x \geq \beta$ be an inequality valid for $\mathrm{CUT}(G)$. By Lemma 4.2, the inequalities obtained from it by collapsing and anticollapsing the end nodes of e_k are valid for $\mathrm{CUT}(G/e_k)$, which is equal to $\nu^{h+k-1}(\mathrm{MET}(G/e_k))$ by the induction assumption. By Proposition 4.10, this implies that $a^T x \geq \beta$ is valid for $\nu^{h+k}(\mathrm{MET}(G))$. □

COROLLARY 4.15. *The N'-index of a graph on $n \geq 6$ nodes is at most $n-5$.*

Proof. If G is connected, one can find a set F of $n-6$ edges whose contraction produces a graph on six nodes; as $\mathrm{CUT}(G/F) = N'(\mathrm{MET}(G/F))$, we deduce from Theorem 4.14 that $\mathrm{CUT}(G) = (N')^{|F|+1}(\mathrm{MET}(G)) = (N')^{n-5}(\mathrm{MET}(G))$. If G is not connected, then, by the above, $\mathrm{CUT}(G_i) = (N')^{n-5}(\mathrm{MET}(G_i))$ for each connected component G_i of G; using Proposition 4.17, this implies that $\mathrm{CUT}(G) = (N')^{n-5}(\mathrm{MET}(G))$. □

4.4. Behavior of the index under taking graph minors and clique sums. An important motivation for the study of the LS relaxations is that one can solve the max-cut problem in polynomial time over the class of graphs having bounded ν-index ($\nu = N, N_+$) or bounded projected ν-index ($\nu = N, \ldots, N'_+$). It is therefore of great interest to understand which graphs have small index, e.g., ≤ 1. This is, however, a difficult question. As a first step, we study here whether these graph classes are closed under taking minors and clique sums.

Let $G = (V_n, E)$ be a graph with edge set $E \subseteq E_n$. Given an edge $e = uv \in E$, recall that $G \backslash e$ is the graph obtained from G by *deleting* edge e, and G/e is the graph obtained from G by contracting e; a *minor* of G is then a graph obtained from G by a sequence of deletions and/or contractions. Let $G_i(V_i, E_i)$ $(i = 1, 2)$ be two graphs such that the set $V_1 \cap V_2$ induces a clique in both G_1 and G_2. Then the graph $G := (V_1 \cup V_2, E_1 \cup E_2)$ is called the *clique t-sum* of G_1 and G_2, where $t := |V_1 \cap V_2|$.

PROPOSITION 4.16. *For $\nu = N, \ldots, N'_+$, $\eta^\pi_\nu(H) \leq \eta^\pi_\nu(G)$ if H is a minor of G, and $\eta_\nu(H) \leq \eta_\nu(G)$ if H is a contraction minor of G.*

Proof. Monotonicity of the projected index under taking deletion minors follows directly from the definitions. Suppose now that H is a contraction minor of G; say, $G = (V_n, E)$, $e := uv \in E$, and $H = G/uv = (V_n \setminus \{u, v\} \cup \{w\}, F)$. We show that $\eta_\nu(H) \leq \eta_\nu(G)$. For this, suppose that $\widetilde{\mathrm{CUT}}(G) = \nu^k(\mathrm{MET}(G))$; we show that $\mathrm{CUT}(H) = \nu^k(\mathrm{MET}(H))$. Let $x \in \nu^k(\widetilde{\mathrm{MET}}(H))$; then $x = Xe_0$ for some $X \in \mu(\nu^{k-1}(\mathrm{MET}(H)))$. By Proposition 4.7, the 1-extension Y of X belongs to $\mu(\nu^{k-1}(\mathrm{MET}(G)))$, and $Y_{0,0} = Y_{0,uv}$. Thus $y := Ye_0 \in \nu^k(\widetilde{\mathrm{MET}}(G)) = \widetilde{\mathrm{CUT}}(G)$. By Lemma 4.1(ii), this implies that $x = y^{F,1} \in \widetilde{\mathrm{CUT}}(H)$.

We now show that $\eta^\pi_\nu(H) \leq \eta^\pi_N(G)$. Suppose that $\mathrm{CUT}(G) = \nu^k(G)$; we show that $\mathrm{CUT}(H) = \nu^k(H)$. For this, let $x \in \nu^k(H)$. Thus $x = \pi_F(Xe_0)$ for some $X \in \mu(\nu^{k-1}(\mathrm{MET}(K_{n-1})))$ with $X_{0,0} = 1$. Viewing K_{n-1} as K_n/uv, we have from Proposition 4.7 that the 1-extension Y of X belongs to $\mu(\nu^{k-1}(\mathrm{MET}(K_n)))$, and

$Y_{0,0} = Y_{0,uv} = 1$. Thus $y := \pi_E(Ye_0) \in \nu^k(G) = \mathrm{CUT}(G)$, implying $x = y^{F,1} \in \mathrm{CUT}(H)$. □

PROPOSITION 4.17. *Let G be the clique t-sum of two graphs G_1 and G_2, where $t = 0, 1, 2, 3$. Then $\eta_\nu^\pi(G) \le \max(\eta_\nu^\pi(G_1), \eta_\nu^\pi(G_2))$ and $\eta_\nu(G) \le \max(\eta_\nu(G_1), \eta_\nu(G_2))$.*

Proof. Let $G = (V_n, E)$ be the clique t-sum of two graphs $G_i = (V_i, E_i)$ for $i = 1, 2$ with $t \le 3$; thus $V_n = V_1 \cup V_2$ and $E = E_1 \cup E_2$. We use the following fact shown in [4]: Given $y \in \mathbf{R}^{E_1 \cup E_2}$ and its projections $y_i := (y(e))_{e \in E_i}$ for $i = 1, 2$, we then have $y \in \mathrm{CUT}(G) \iff y_i \in \mathrm{CUT}(G_i)$ for $i = 1, 2$. Let $k \ge 0$ be an integer. Suppose first that $\mathrm{CUT}(G_i) = \nu^k(G_i)$ for $i = 1, 2$ and let $y \in \nu^k(G)$; we show that $y \in \mathrm{CUT}(G)$. For this it suffices to show that $y_i \in \nu^k(G_i)$ for $i = 1, 2$. There exists $Y \in \mu(\nu^{k-1}(\mathrm{MET}(K_n)))$ such that $y = \pi_E(Ye_0)$. By Proposition 4.12, the principal submatrix Y_i of Y indexed by $\{0\} \cup F_i$, where F_i is the edge set of the complete graph on V_i, belongs to $\mu(\nu^{k-1}(\mathrm{MET}(K_{V_i})))$. Thus $y_i = \pi_{E_i}(Y_i e_0) \in \nu^k(G_i)$ for $i = 1, 2$.

Suppose now that $\widetilde{\mathrm{CUT}(G_i)} = \nu^k(\mathrm{MET}(G_i))$ for $i = 1, 2$ and let $y \in \nu^k(\widetilde{\mathrm{MET}(G)})$; we show that $y_i \in \nu^k(\widetilde{\mathrm{MET}(G_i)})$. There exists $Y \in \mu(\nu^{k-1}(\mathrm{MET}(G)))$ such that $y = Ye_0$. By Proposition 4.12, the principal submatrix Y_i of Y indexed by $\{0\} \cup E_i$ belongs to $\mu(\nu^{k-1}(\mathrm{MET}(G_i)))$, and thus $y_i = Y_i e_0 \in \nu^k(\widetilde{\mathrm{MET}(G_i)})$. □

As the class of graphs G with $\eta_\nu^\pi(G) \le 1$ is closed under taking minors, we know from the theory of Robertson and Seymour [26] that there exists a finite list of *minimal forbidden minors* characterizing membership in that class; that is, $\eta_\nu^\pi(G) \le 1$ if and only if G does not contain any member of the list as a minor. For $\nu = N, N_+$, $\eta_\nu^\pi(K_6 \backslash e) = 1$ while $\eta_\nu^\pi(K_6) = 2$; hence the graph K_6 is a minimal forbidden minor for both properties $\eta_N^\pi(G) \le 1$ and $\eta_{N_+}^\pi(G) \le 1$. There are necessarily other minimal forbidden minors. Indeed, the max-cut problem is known to be NP-hard for the class of graphs having no K_6-minor (in fact, also for the class of apex graphs; that is, the graphs having a node whose deletion results in a planar graph) (cf. [5]).

Let G_0 denote the graph obtained from K_7 by removing a matching of size 3. We have verified that, for a graph G on 7 nodes distinct from G_0, $\eta_N^\pi(G) \le 1$ if and only if it does not contain K_6 as a minor. It would be interesting to compute $\eta_N^\pi(G_0)$; if its value is ≥ 2, then G_0 is another minimal forbidden minor.

In view of Propositions 4.16 and 4.17, the property $\nu_\nu^\pi(G) \le 1$ is preserved under the ΔY operation (which consists of replacing a triangle by a claw $K_{1,3}$). However, it is not preserved under the converse $Y\Delta$ operation. Indeed, if G is the graph obtained from K_6 by applying one ΔY transformation, then $\eta_N(G) = \eta_N^\pi(G) = 1$ (by (4.7)) while $\eta_N(K_6) = 2$. We have verified that all the graphs in the Petersen family (consisting of the graphs that can be obtained from K_6 by $Y\Delta$ and ΔY transformations) except K_6 have projected N-index equal to 1.

5. Valid inequalities for the new relaxations. We saw above that the N-index of K_n is at most $n - 4$, with equality for $n = 4, 5$. We conjecture that equality holds for any n. In order to show this conjecture, one has to find an inequality valid for $\mathrm{CUT}(K_n)$ which is not valid for $N^{n-5}(K_n)$. A possible candidate is the inequality

$$\sum_{1 \le i < j \le n} x_{ij} \ge -\left\lfloor \frac{n}{2} \right\rfloor. \tag{5.1}$$

Note that (5.1) is not valid for $N^{n-5}(K_n)$ if and only if there exists $a < -\frac{1}{n}$ (n odd) or $a < -\frac{1}{n-1}$ (n even) for which $(a, \ldots, a) \in N^{n-5}(K_n)$. We will show in Proposition 5.3 that inequality (5.1) is not valid for $N^{n-5}(K_n)$ if $n = 7$; we conjecture that

this remains true for any odd n. However, for n even, inequality (5.1) *is* valid for $N^{n-5}(K_n)$. (Indeed, for n even, inequality (5.1) follows by summation from the inequalities (5.1) for $n-1$; as the latter inequalities are valid for $N^{n-5}(K_{n-1})$, we deduce that (5.1) too is valid for $N^{n-5}(K_n)$.) Therefore, for n even, one should use some more complicated inequality. We will show in Proposition 5.2 that the inequality

$$(n-4)\sum_{i=2}^{n} x_{1i} + \sum_{2\le i<j\le n} x_{ij} \ge -\frac{1}{2}(n^2-7n+14) \tag{5.2}$$

is not valid for $N^{n-5}(K_n)$ if $n=6$, and we conjecture that this holds for any even $n \ge 6$. The inequalities (5.1) and (5.2) are special instances of gap inequalities that we now introduce.

5.1. Gap inequalities. Given an integer vector $b=(b_1,\dots,b_n)\in\mathbf{Z}^n$, its *gap* $\gamma(b)$ is defined as

$$\gamma(b) := \min_{S\subseteq V_n} \left| \sum_{i\in S} b_i - \sum_{i\in V_n\setminus S} b_i \right|,$$

and the inequality

$$\sum_{1\le i<j\le n} b_i b_j x_{ij} \ge \frac{1}{2}\left(\gamma(b)^2 - \sum_{i=1}^{n} b_i^2\right) \tag{5.3}$$

in the variable $x\in\mathbf{R}^{E_n}$ is called the *gap inequality* associated with b. The analogue of (5.3) in the matrix variable $X\in\mathcal{S}_n^1$ takes the simpler form

$$b^T X b \ge \gamma(b)^2. \tag{5.4}$$

Inequality (5.4) is obviously valid for any cut matrix xx^T ($x\in\{\pm1\}^n$); that is, inequality (5.3) is valid for the cut polytope $\mathrm{CUT}(K_n)$. The gap inequalities are introduced in [19] as a generalization of negative-type inequalities (case $\gamma(b)=0$, [27]) and hypermetric inequalities (case $\gamma(b)=1$, [9]); see [10] for a detailed survey.

The class of gap inequalities is closed under switching; indeed, switching the gap inequality for $b\in\mathbf{Z}^n$ along the cut $\delta(S)$ amounts to flipping the signs of the components of b on S. (Anti)collapsing specializes to gap inequalities in the following manner. Given $b=(b_1,b_2,\dots,b_n)\in\mathbf{Z}^n$, set $b':=(b_1+b_2,b_3,\dots,b_n)\in\mathbf{Z}^{n-1}$ and $b'':=(b_1-b_2,b_3,\dots,b_n)\in\mathbf{Z}^{n-1}$. As $\gamma(b'),\gamma(b'')\ge\gamma(b)$, we have that $\frac{1}{2}(\gamma(b')^2-\sum_{i=2}^n b_i'^2)\ge \frac{1}{2}(\gamma(b)^2-\sum_{i=1}^n b_i^2)-b_1b_2$ and $\frac{1}{2}(\gamma(b'')^2-\sum_{i=2}^n b_i''^2)\ge \frac{1}{2}(\gamma(b)^2-\sum_{i=1}^n b_i^2)+b_1b_2$. Therefore, if the gap inequality for b' (resp., b'') is valid for $\nu^k(K_{n-1})$, then the inequality obtained from the gap inequality for b by collapsing (resp., anticollapsing) nodes 1 and 2 is valid for $\nu^k(K_{n-1})$. This fact will be useful when applying Propositions 4.10 and 4.11 to gap inequalities.

The negative-type inequalities do not induce facets of $\mathrm{CUT}(K_n)$ (since they are implied by the hypermetric inequalities); moreover, they are implied by the condition $X\succeq 0$. In fact, no gap inequality for $b\in\mathbf{Z}^n$ with gap $\gamma(b)\ge 2$ and inducing a facet of the cut polytope is known (cf. [19]). On the other hand, hypermetric inequalities include large classes of facets for the cut polytope. This is the case, for instance, for the following vectors b:

$$b=(1,\dots,1)\in\mathbf{Z}^n \text{ for } n \text{ odd}, \qquad b=(n-4,1,\dots,1)\in\mathbf{Z}^n \text{ for } n\ge 4.$$

The hypermetric inequality for $b = (1,1,1)$ is a triangle inequality (occurring in case (5.1) for $n = 3$, and in case (5.2) for $n = 4$); the hypermetric inequality for $b = (1,1,1,1,1)$ is called the *pentagonal inequality* (occurring in cases (5.1) and (5.2) for $n = 5$). Moreover, for $n \leq 6$, all facets of $\mathrm{CUT}(K_n)$ are induced by hypermetric inequalities. More precisely, $\mathrm{CUT}(K_n) = \mathrm{MET}(K_n)$ for $n \leq 4$; up to switching, all facets of $\mathrm{CUT}(K_5)$ arise from the triangle inequality and the pentagonal inequality; up to switching and permutation, all facets of $\mathrm{CUT}(K_6)$ arise from the triangle inequality, the pentagonal inequality, and the hypermetric inequality for $b = (2,1,1,1,1,1)$ (case $n = 6$ of (5.2)).

5.2. Valid hypermetric inequalities for the new relaxations. By construction, the triangle inequalities are valid for $N(K_n)$. As $\mathrm{CUT}(K_5) = N(K_5)$ (by Corollary 4.4), the pentagonal inequality (that is, the gap inequality for $b = (1,1,1,1,1,0,\ldots,0)$) is also valid for $N(K_n)$. We now examine the validity of the gap inequalities for $(1,\ldots,1) \in \mathbf{Z}^n$ ($n \geq 7$, odd) and $(n-4,1,\ldots,1)$ ($n \geq 6$).

PROPOSITION 5.1. *Let $k \geq 1$ be an integer and $n := 2k+3$. The gap inequality for $(1,\ldots,1) \in \mathbf{Z}^n$ is valid for $N_+^k(K_n)$.*

Proof. We proceed by induction for $k \geq 1$. The result holds for $k = 1$. Let $k \geq 2$ and assume that the result holds for $k-1$. By the induction assumption, the gap inequality for $b'' := (0,1,\ldots,1) \in \mathbf{Z}^{n-1}$ is valid for $N_+^{k-1}(K_{n-1})$. Therefore, using Proposition 4.11, we deduce that the gap inequality for b is valid for $N_+^k(K_n)$. ∎

One cannot hope to improve the above result and show validity for $N^k(K_n)$ with the help of Proposition 4.10; indeed, collapsing of the gap inequality for $(1,1,1,1,1,1,1) \in \mathbf{Z}^7$ gives the gap inequality for $(2,1,1,1,1,1) \in \mathbf{Z}^6$ which, as we see below, is *not* valid for $N(K_6)$. In fact, the gap inequality for $(1,1,1,1,1,1,1)$ is *not* valid for $N^2(K_7)$ (cf. Proposition 5.3). The proofs of Propositions 5.2–5.4 below, being quite technical, are delayed until section 7.

PROPOSITION 5.2. *The gap inequality for $(n-4,1,\ldots,1) \in \mathbf{Z}^n$ is valid for $N'(K_n)$ if $n = 6,7$, it is not valid for $N'(K_n)$ if $n \geq 8$, it is not valid for $N_+(K_n)$ if $n \geq 6$, and it is valid for $N^{n-5}(K_n)$ for $n \geq 7$.*

PROPOSITION 5.3. *The hypermetric inequality for $(1,\ldots,1) \in \mathbf{Z}^n$ ($n \geq 7$, odd) is not valid for $N'_+(K_n)$ nor for $N^2(K_n)$.*

PROPOSITION 5.4. *$\mathrm{CUT}(K_n) = N(K_n)$ if $n \leq 5$, $\mathrm{CUT}(K_6) = N'(K_6) \subset N_+(K_6)$, $N'_+(K_n) \subset N_+(K_n) \subset N(K_n)$ for $n \geq 6$, and $\mathrm{CUT}(K_n) \subset N'_+(K_n)$ for $n \geq 7$.*

Let $a^T x \geq \beta$ be an inequality valid for $\mathrm{CUT}(K_n)$ and let G denote its *support graph*, whose edges are the pairs ij for which $a_{ij} \neq 0$. Obviously, the inequality $a^T x \geq \beta$ is valid for $N(\mathrm{MET}(G))$ if $\eta_N(G) \leq 1$. This is the case, for instance, for parachute inequalities (cf. section 30.4 in [10]) and for bicycle odd wheel inequalities, that is, the inequalities

$$x_{uv} + \sum_{ij \in E(C)} x_{ij} + \sum_{i \in V(C)} (x_{ui} + x_{vi}) \geq 1 - |C|,$$

where C is an odd circuit and u, v two adjacent nodes that are adjacent to all nodes of C.

6. Application to the stable set polytope. We explain here how the LS relaxations $\nu(\mathrm{MET}(G))$ for the cut polytope permit us to tighten the corresponding LS relaxations for the stable set polytope. Given a graph $G = (V_n, E)$, its *fractional*

stable set polytope is

$$\mathrm{FRAC}(G) := \{d \in \mathbf{R}^n \mid d \geq 0,\ d_i + d_j \leq 1 \text{ for all } ij \in E\},$$

and its *stable set polytope* is

$$\mathrm{STAB}(G) := \mathrm{conv}(x \in \{0,1\}^n \mid x \in \mathrm{FRAC}(G)).$$

Lovász and Schrijver [22] studied the relaxations $N(\mathrm{FRAC}(G))$ and $N_+(\mathrm{FRAC}(G))$ in detail. (As $\mathrm{FRAC}(G)$ lives in the unit cube $Q = [0,1]^d$, the operators N, N_+ are now defined in the context of $0,1$ variables, which means that condition (2.1) is replaced by $y_{i,i} = y_{0,i}$ for $i = 1, \ldots, d$, while condition (2.4) is replaced by $Y(e_i), Y(e_0 - e_i) \in \tilde{K}$ $(i = 1, \ldots, d)$.) In particular, they have shown the following results. The relaxation $N(\mathrm{FRAC}(G))$ is equal to the polytope $\mathrm{ODD}(G)$ defined by nonnegativity, the edge inequalities $d_i + d_j \leq 1$ $(ij \in E)$, and the odd hole inequalities $\sum_{i \in V(C)} d_i \leq \frac{|C|-1}{2}$ (C being an odd circuit in G). Any clique inequality $\sum_{i \in V(K)} d_i \leq 1$ (K a clique in G) is valid for $N_+(\mathrm{FRAC}(G))$ and $N^{|K|-2}(\mathrm{FRAC}(G))$ but not for $N^{|K|-3}(\mathrm{FRAC}(G))$; odd wheel inequalities, odd antihole inequalities, orthogonality constraints are valid for $N_+(\mathrm{FRAC}(G))$.

Let G^∇ denote the graph obtained from G by adding a new node a (the *apex* node) adjacent to all nodes of G and set

$$\mathcal{L}_G := \{x \in \mathbf{R}^{E(G^\nabla)} \mid x_{ij} - x_{ai} - x_{aj} = -1 \text{ for all } ij \in E\}.$$

For $d \in \mathbf{R}^{V_n}$ define $x := \varphi(d) \in \mathbf{R}^{E(G^\nabla)}$ by

$$x_{ai} := 1 - 2d_i \ (i \in V_n), \quad x_{ij} := 1 - 2d_i - 2d_j \ (ij \in E). \tag{6.1}$$

Then φ is a bijection between $\mathbf{R}^{V_n}$ and $\mathbf{R}^{E(G^\nabla)}$. For $S \subseteq V_n$, the (± 1)-incidence vector of the cut $\delta(S)$ (in G^∇) lies in $\mathcal{L}_G$ if and only if S is a stable set in G. This shows the following well-known fact (cf., e.g., [25]):

$$\varphi\,(\mathrm{STAB}(G)) = \mathrm{CUT}(G^\nabla) \cap \mathcal{L}_G. \tag{6.2}$$

As $\varphi(\mathrm{STAB}(G))$ is a face of $\mathrm{CUT}(G^\nabla)$, every valid inequality for $\mathrm{CUT}(G^\nabla)$ gives rise to a valid inequality for $\mathrm{STAB}(G)$. For instance, if C is an odd circuit in G, the circuit inequality $\sum_{ij \in E(C)} x_{ij} \geq 2 - |C|$ for $\mathrm{CUT}(G^\nabla)$ gives rise to the odd hole inequality $\sum_{i \in V(C)} d_i \leq \frac{|C|-1}{2}$ for $\mathrm{STAB}(G)$ (as $\sum_{ij \in E(C)} x_{ij} = |C| - 4\sum_{i \in V(C)} d_i$); one can verify that the (switching of the) bicycle odd wheel inequality

$$-x_{au} + \sum_{i \in V(C)} (-x_{ai} + x_{ui}) + \sum_{ij \in E(C)} x_{ij} \geq 1 - |C|$$

for $\mathrm{CUT}(G^\nabla)$ gives rise to the odd wheel inequality $\sum_{i \in V(C)} d_i + \frac{|C|-1}{2} d_u \leq \frac{|C|-1}{2}$ for $\mathrm{STAB}(G)$, and that the gap inequality for $(b_a, b_1, \ldots, b_n) = (-(n-3), 1, \ldots, 1) \in \mathbf{Z}^{n+1}$ for $\mathrm{CUT}(G^\nabla)$ gives rise to the clique inequality $\sum_{i=1}^n d_i \leq 1$. It is shown in [20] that the correspondence (6.2) extends at the level of the basic linear and semidefinite relaxations; namely,

$$\varphi\,(\mathrm{ODD}(G)) = \mathrm{MET}(G^\nabla) \cap \mathcal{L}_G \quad \text{and} \quad \varphi\,(\mathrm{TH}(G)) = \mathcal{E}(G^\nabla) \cap \mathcal{L}_G, \tag{6.3}$$

where $\mathcal{E}(G^\nabla)$ is the projection of $\mathcal{E}_{n+1}$ on $\mathbf{R}^{E(G^\nabla)}$ and $\mathrm{TH}(G)$ is the *theta body* defined as the set of vectors $x \in \mathbf{R}^{V_n}$ for which $(1,x) = Xe_0$ for some positive semidefinite matrix $X = (x_{ij})_{i,j=0}^n$ satisfying $x_{0i} = x_{ii}$ $(i = 1,\dots,n)$ and $x_{ij} = 0$ $(ij \in E)$. It follows from the above that

$$\varphi(\mathrm{STAB}(G)) \subseteq \mathrm{MET}(G^\nabla) \cap \mathcal{L}_G = \varphi(N(\mathrm{FRAC}(G))).$$

We now examine how the correspondence between the relaxations $\nu(\mathrm{MET}(G^\nabla))$ and $\nu(\mathrm{FRAC}(G))$ carries out for $\nu = N, N_+, N', N'_+$ and their iterates.

PROPOSITION 6.1. *Let $k \geq 0$ be an integer. Then*

$$\varphi(\mathrm{STAB}(G)) \subseteq N^k(\mathrm{MET}(G^\nabla)) \cap \mathcal{L}_G \subseteq \varphi\left(N^{k+1}(\mathrm{FRAC}(G))\right),$$

and, for $\nu = N_+, N', N'_+$,

$$\varphi(\mathrm{STAB}(G)) \subseteq \nu^k(\mathrm{MET}(G^\nabla)) \cap \mathcal{L}_G \subseteq \varphi\left(\nu^k(\mathrm{FRAC}(G))\right).$$

Proof. The left inclusions follow from (6.2). We show that $N^k(\mathrm{MET}(G^\nabla)) \cap \mathcal{L}_G$ is contained in $\varphi(N^{k+1}(\mathrm{FRAC}(G)))$ by induction on $k \geq 0$. The inclusion holds for $k = 0$. Let $k \geq 1$ and suppose that the inclusion holds for $k-1$. Let $x \in N^k(\mathrm{MET}(G^\nabla)) \cap \mathcal{L}_G$; then $(1,x) = Ye_0$ for some $Y \in M(N^{k-1}(\mathrm{MET}(G^\nabla)))$. Let Z denote the matrix indexed by $\{0\} \cup V_n$ defined by

$$\begin{array}{l} Z_{0,0} := 1,\ Z_{0,i} = Z_{i,i} := \frac{1}{2}(1 - Y_{0,ai}) \quad (i \in V_n), \\ Z_{i,j} := \frac{1}{4}(1 + Y_{ai,aj} - Y_{0,ai} - Y_{0,aj}) \quad (i,j \in V_n). \end{array} \tag{6.4}$$

Then $\varphi^{-1}(x) = (Z_{0,i})_{i \in V_n}$. Therefore the result will follow if we can show that the matrix Z belongs to $M(N^k(\mathrm{FRAC}(G)))$, i.e., that $Z(e_k)$, $Z(e_0 - e_k)$ belong to $N^k(\widetilde{\mathrm{FRAC}(G)})$. By assumption, $Y(e_0 \pm e_f) \in N^{k-1}(\widetilde{\mathrm{MET}(G^\nabla)})$ for all $f \in E(G^\nabla)$. As $Ye_0 \in \widetilde{\mathcal{L}_G}$ and $Ye_0 = \frac{1}{2}(Y(e_0+e_f)+Y(e_0-e_f))$, we deduce that $Y(e_0 \pm e_f) \in \widetilde{\mathcal{L}_G}$, and thus $Ye_f \in \widetilde{\mathcal{L}_G}$ for all $f \in E(G^\nabla)$, which can be rewritten as

$$1 + Y_{0,ij} - Y_{0,ai} - Y_{0,aj} = 0,\ Y_{0,f} + Y_{ij,f} - Y_{ai,f} - Y_{aj,f} = 0 \ \text{ for } f \in E(G^\nabla). \tag{6.5}$$

Using the induction assumption, we obtain that $\varphi^{-1}(Y(e_0 \pm e_{ak}))$ $(k \in V_n)$ belongs to $N^k(\widetilde{\mathrm{FRAC}(G)})$. (We have extended the bijection φ as a bijection between the homogenized spaces $\mathbf{R}^{V_n \cup \{0\}}$ and $\mathbf{R}^{E(G^\nabla) \cup \{0\}}$ in the obvious way; namely, $(x_0, x) = \varphi(d_0, d)$ if $x_0 = d_0$, $x_{ai} = d_0 - 2d_i$, and $x_{ij} = d_0 - 2d_i - 2d_j$.) In order to conclude, it suffices now to observe that $Ze_k = \varphi^{-1}(\frac{1}{2}Y(e_0 - e_{ak}))$ and $Z(e_0 - e_k) = \varphi^{-1}(\frac{1}{2}Y(e_0 + e_{ak}))$ for $k \in V_n$; this is an easy verification using the relation (6.5).

We now show the result for the N' operator. In view of the above, it suffices to show the following result: If $Y \in M'((N')^{k-1}(\mathrm{MET}(G^\nabla)))$ satisfies $Ye_0 \in \widetilde{\mathcal{L}_G}$ and if Z is the associated matrix defined by (6.5), then $Z \in M'((N')^{k-1}(\mathrm{FRAC}(G)))$; that is, Ze_k, $Z(e_0 - e_h - e_k)$ belong to $(N')^{k-1}(\widetilde{\mathrm{FRAC}(G)})$ for all $k \in V_n$, all $hk \in E(G)$, respectively. By assumption, the vectors $Y(e_0 \pm e_f)$ $(f \in E(G^\nabla))$ and $Y(e_0 \pm e_{ai} \pm e_{aj} \pm e_{ij})$ (with an even number of minus signs) $(ij \in E(G))$ belong to $(N')^{k-1}(\mathrm{MET}(G^\nabla))$; as $Ye_0 \in \widetilde{\mathcal{L}_G}$, their images under φ^{-1} belong to $(N')^{k-1}(\mathrm{FRAC}(G))$ (by the induction assumption) and (6.5) holds. To conclude the proof it suffices to verify (using (6.5)) that $Z(e_0 - e_h - e_k) = \varphi^{-1}\left(\frac{1}{4}Y(e_0 + e_{ah} + e_{ak} + e_{hk})\right)$ for $hk \in E(G)$.

The result for the N_+ and N'_+ operators follows, using the fact that $Y \succeq 0 \Longrightarrow Z \succeq 0$, which holds because $b^T Z b = c^T Y c$, where $b \in \mathbf{R}^{n+1}$ and $c := (-(b_0 + \sum_{i=0}^n b_i), b_1, \ldots, b_n)$. □

It is shown in [22] that the smallest integer k for which $N^k(\mathrm{FRAC}(G)) = \mathrm{STAB}(G)$ is less than or equal to $n - \alpha(G) - 1$ if G has at least one edge. On the other hand, by (4.7), $\eta_N(G^\nabla) \leq n + 1 - \alpha(G^\nabla) - 3 = n - \alpha(G) - 2$ if $\alpha(G) \leq n - 2$. The similarity between the two bounds reflects the fact that $\mathrm{STAB}(G)$ arises as a face of $\mathrm{CUT}(G^\nabla)$. In fact the two upper bounds match, as the discrepancy of 1 can be explained by the fact that in the case of the cut polytope we start with a stronger relaxation than in the case of the stable set polytope; indeed, in view of (6.3), we "win" one iteration at the beginning step.

The inclusion $N^k(\mathrm{MET}(G^\nabla)) \cap \mathcal{L}_G \subseteq \varphi\left(N^{k+1}(\mathrm{FRAC}(G))\right)$ holds at equality for $k = 0$ for all graphs and is strict for $k \geq 1$ for certain graphs. Indeed, for $k \geq 1$,

$$\mathrm{STAB}(K_{k+4}) = \varphi^{-1}\left(N^k(\mathrm{MET}(K^\nabla_{k+4})) \cap \mathcal{L}_{K_{k+4}}\right) \subset N^{k+1}(\mathrm{FRAC}(K_{k+4})).$$

To see it, note that the clique inequality $\sum_{i=1}^{k+4} d_i \leq 1$ is not valid for $N^{k+1}(\mathrm{FRAC}(K_{k+4}))$, while it is valid for $\varphi^{-1}\left(N^k(\mathrm{MET}(K^\nabla_{k+4})) \cap \mathcal{L}_{K_{k+4}}\right)$. The latter holds because the clique inequality $\sum_{i=1}^{k+4} d_i \leq 1$ arises from the gap inequality for $(-(k+1), 1, \ldots, 1) \in \mathbf{Z}^{k+5}$ (assigning $-(k+1)$ to the apex node), which is valid for $N^k(\mathrm{MET}(K_{k+5}))$ when $k \geq 2$ by Proposition 5.2; in the case $k = 1$, while not valid for $N(\mathrm{MET}(K_6))$, the gap inequality for $(-2, 1, 1, 1, 1, 1)$ is valid for $N(\mathrm{MET}(K_5^\nabla)) \cap \mathcal{L}_{K_5}$ (cf. Lemma 7.5).

We know that clique and odd antihole inequalities are valid for $N_+(\mathrm{MET}(G^\nabla)) \cap \mathcal{L}_G$ (as they are valid for $N_+(\mathrm{FRAC}(G))$). It would be interesting to find for them some "parent" inequality for $\mathrm{CUT}(G^\nabla)$ which would be valid for $N_+(\mathrm{MET}(G^\nabla))$.

7. Proofs of Propositions 5.2–5.4. We study here in detail the validity of the gap inequalities for $c_n := (1, \ldots, 1) \in \mathbf{Z}^n$ ($n \geq 7$ odd) and for $b_n := (n-4, 1, \ldots, 1) \in \mathbf{Z}^n$ ($n \geq 6$) for some relaxations $\nu^k(K_n)$. Set

$$C_n := \min\left(\sum_{1 \leq i < j \leq n} x_{ij} \mid x \in \nu^k(K_n)\right), \tag{7.1}$$

$$B_n := \min\left((n-4)\sum_{i=2}^n x_{1i} + \sum_{2 \leq i < j \leq n} x_{ij} \mid x \in \nu^k(K_n)\right). \tag{7.2}$$

Given some scalars $a, c \in \mathbf{R}$, the vector $x(a,c) \in \mathbf{R}^{E_n}$ is defined by

$$x(a,c)_{1i} := a \quad \text{for } i = 2, \ldots, n, \qquad x(a,c)_{ij} := c \quad \text{for } 2 \leq i < j \leq n; \tag{7.3}$$

it is said to have *pattern* (a, c).

A first basic observation is that the minimum in the program (7.1) (resp., (7.2)) is attained at a point of $\nu^k(K_n)$ having some pattern (a, a) (resp., (a, c)). Indeed, let $x \in \nu^k(K_n)$ be an optimum solution to program (7.1) and set $x^* := \frac{1}{n!}\sum_\sigma x^\sigma$, where the sum is taken over all permutations σ of $[1, n]$; then $x^* \in \nu^k(K_n)$ is still optimum for (7.1) and has pattern (a, a) for some $a \in \mathbf{R}$. The reasoning is similar in the case of program (7.2), except $x^* := \frac{1}{(n-1)!}\sum_\sigma x^\sigma$, where the sum is now taken over all permutations of $[1, n]$ fixing 1; then x^* has pattern (a, c) for some $a, c \in \mathbf{R}$.

For the proofs of Propositions 5.2 and 5.3 we need to determine the conditions on a, c which will permit us to express membership of the vector $x(a, c)$ in $N(K_n)$ and $N'(K_n)$. The study of validity for $N^2(K_n)$ will involve checking the membership in $N(K_n)$ of a more complicated vector $x(a, b, c, d) := x$, defined as follows:

$$x_{12} := a,\ x_{1i} := b,\ x_{2i} := c \ \text{ for } i = 3, \dots, n, \quad x_{ij} := d \ \text{ for } 3 \le i < j \le n; \tag{7.4}$$

(a, b, c, d) is again called the *pattern* of the vector $x(a, b, c, d)$. Note that $x(a, b, c, d) = x(a, c)$ if $a = b$ and $c = d$.

The rest of this section is organized as follows. In section 7.1 we determine the conditions on a, b, c, d expressing membership in $N(K_n)$ for the vector $x(a, b, c, d)$ or membership in $N'(K_n)$ for the vector $x(a, c)$. These results are then applied in sections 7.2–7.3 to proving Propositions 5.3–5.4.

7.1. Vectors with pattern (a, b, c, d). We begin by determining the conditions on a, b, c, d expressing membership in $N(K_n)$ for a vector with pattern (a, b, c, d). By definition, $x := x(a, b, c, d) \in N(K_n)$ if and only if $(1, x) = Ye_0$ for some matrix $Y \in M(\mathrm{MET}(K_n))$. In fact, such a matrix Y can be assumed to satisfy certain symmetries. Indeed, set $Y^* := \frac{1}{(n-2)!} \sum_\sigma Y^\sigma$, where the sum is taken over all permutations σ of $[1, n]$, fixing 1 and 2 (recall the definition of Y^σ from (3.5)). Then $Y^* \in M(\mathrm{MET}(K_n))$ and $Y^* e_0 = (1, x)$. Moreover, the matrix Y^* has the property that the value of its (ij, hk)th entry depends only on whether the pairs ij and hk meet and whether they contain any of the points 1 and 2. Namely, if the pairs ij and hk meet, then the value of $Y_{ij,hk}$ is determined by relation (3.3) and is thus one of a, b, c, d; otherwise,

$$\begin{aligned}
&Y_{12,ij} = x \quad \text{for } 3 \le i < j \le n,\\
&Y_{1i,2j} = z \quad \text{for } 3 \le i \ne j \le n,\\
&Y_{1i,hk} = y,\ Y_{2i,hk} = u \quad \text{for } 3 \le i \le n,\ 3 \le i < j \le k,\ h, k \ne i,\\
&Y_{ij,hk} = v \quad \text{for } 3 \le i < j \le n,\ 3 \le h < k \le n,\ \{i, j\} \cap \{h, k\} = \emptyset
\end{aligned} \tag{7.5}$$

for some scalars x, y, z, u, v; $(a, b, c, d, x, y, z, u, v)$ is then called the *pattern* of Y.

Let $\mathcal{Y}_n$ denote the set of matrices $Y \in \mathcal{S}^1_{1+d_n}$ having some pattern $(a, b, c, d, x, y, z, u, v)$ as defined above. A matrix $Y \in \mathcal{Y}_6$ is shown in Figure 7.1. When $a = b$ and $c = d$ (i.e., when $x = x(a, c)$), the matrix Y can be assumed to satisfy the additional symmetry $x = y = z$ and $u = v$, and (a, c, x, u) is then called the *simplified pattern* of Y. (Such a matrix is pictured in Figure A.1.)

We first work out the conditions on $a, \dots, v$ for membership of $Y \in \mathcal{Y}_n$ in $M(\mathrm{MET}(K_n))$, and then deduce the conditions on a, b, c, d for membership of $x(a, b, c, d)$ in $N(K_n)$.

LEMMA 7.1. *Let $Y \in \mathcal{Y}_n$ with pattern $(a, b, c, d, x, y, z, u, v)$ and $n \ge 6$. Then Y*

$$
\begin{array}{c} \\ 0 \\ 12 \\ 13 \\ 14 \\ 15 \\ 16 \\ 23 \\ 24 \\ 25 \\ 26 \\ 34 \\ 35 \\ 36 \\ 45 \\ 46 \\ 56 \end{array}
\begin{array}{c}
\begin{array}{cccccccccccccccc} 0 & 12 & 13 & 14 & 15 & 16 & 23 & 24 & 25 & 26 & 34 & 35 & 36 & 45 & 46 & 56 \end{array} \\
\left(\begin{array}{cccccccccccccccc}
1 & a & b & b & b & b & c & c & c & c & d & d & d & d & d & d \\
a & 1 & c & c & c & c & b & b & b & b & x & x & x & x & x & x \\
b & c & 1 & d & d & d & a & z & z & z & b & b & b & y & y & y \\
b & c & d & 1 & d & d & z & a & z & z & b & y & y & b & b & y \\
b & c & d & d & 1 & d & z & z & a & z & y & b & y & b & y & b \\
b & c & d & d & d & 1 & z & z & z & a & y & y & b & y & b & b \\
c & b & a & z & z & z & 1 & d & d & d & c & c & c & u & u & u \\
c & b & z & a & z & z & d & 1 & d & d & c & u & u & c & c & u \\
c & b & z & z & a & z & d & d & 1 & d & u & c & u & c & u & c \\
c & b & z & z & z & a & d & d & d & 1 & u & u & c & u & c & c \\
d & x & b & b & y & y & c & c & u & u & 1 & d & d & d & d & v \\
d & x & b & b & b & y & c & u & c & u & d & 1 & d & d & v & d \\
d & x & b & y & y & b & c & u & u & c & d & d & 1 & v & d & d \\
d & x & y & b & b & y & u & c & c & u & d & d & v & 1 & d & d \\
d & x & y & b & y & b & u & c & u & c & d & v & d & d & 1 & d \\
d & x & y & y & b & b & u & u & c & c & v & d & d & d & d & 1
\end{array}\right)
\end{array}
$$

FIG. 7.1. *A matrix $Y \in \mathcal{Y}_6$ with pattern $(a, b, c, d, x, y, z, u, v)$.*

belongs to $M(\mathrm{MET}(K_n))$ if and only if $a, \ldots, v$ satisfy the linear inequalities

$$
\begin{array}{c}
a+2b+2c+d+x \geq -1, \quad a-2b-2c+d+x \geq -1, \\
-a+2b-2c+d-x \geq -1, \quad -a-2b+2c+d-x \geq -1, \\
a-d-x \geq -1, \ -a-d+x \geq -1, \quad a+3d+3x \geq -1, \quad -a+3d-3x \geq -1, \\
a+2b+2c+d+z \geq -1, \quad -a+2b-2c+d-z \geq -1, \quad a-d-z \geq -1, \\
-a-d+z \geq -1, \quad a-2b-2c+d+z \geq -1, \quad -a-2b+2c+d-z \geq -1, \\
3b+3d+y \geq -1, \quad -3b+3d-y \geq -1, \quad -b-d+y \geq -1, \quad b-d-y \geq -1, \\
b+2c+d+y+2z \geq -1, \quad b-2c+d+y-2z \geq -1, \quad b-d-y \geq -1, \\
-b+2c+d-y-2z \geq -1, \quad -b-2c+d-y+2z \geq -1, \quad -b-d+y \geq -1, \\
b+3d+3y \geq -1, \quad -b+3d-3y \geq -1, \quad 3c+3d+u \geq -1, \quad -3c+3d-u \geq -1, \\
2b+c+d+2z+u \geq -1, \quad -2b+c+d-2z+u \geq -1, \quad c-d-u \geq -1, \\
2b-c+d-2z-u \geq -1, \quad -2b-c+d+2z-u \geq -1, \quad -c-d+u \geq -1, \\
c+3d+3u \geq -1, \quad -c+3d-3u \geq -1, \quad 6d+v \geq -1, \quad -2d+v \geq -1, \quad v \leq 1.
\end{array} \tag{7.6}
$$

Proof. By definition, $Y \in M(\mathrm{MET}(K_n))$ if and only if, for all $ij \in E_6$, $y := Y(e_0 \pm e_{ij})$ satisfies all triangle inequalities. By symmetry, it suffices to consider the cases when $ij = 12$, 13, 23, or 34. Let $ij = 12$. Due to symmetry and to the fact that $y_{12} = \pm y_0$, it suffices to consider the triangle inequalities based on the triples 134 and 345. The triangle inequalities based on triple 134 can be reformulated as

$$
\begin{array}{c}
a+2b+2c+d+x \geq -1, \quad a-2b-2c+d+x \geq -1, \quad a-d-x \geq -1, \\
-a+2b-2c+d-x \geq -1, \quad -a-2b+2c+d-x \geq -1, \quad -a-d-x \geq -1,
\end{array}
$$

and those based on triple 345 give

$$
a+3d+3x \geq -1, \qquad -a+3d-3x \geq -1.
$$

Next let ij be one of $13, 23, 34$. Due to symmetry and to the fact that $y_{ij} = \pm y_0$, it suffices to consider the triangle inequalities based on the triples 124, 145, 245, and

456. When $ij = 13$, we find from (7.6) the relations $a + 2b + 2c + d + z \geq -1$ until $-b + 3d - 3y \geq -1$. When $ij = 23$, we find the relations $3c + 3d + u \geq -1$ until $-c + 3d - 3u \geq -1$. When $ij = 34$, we find the relations $6d + v \geq -1, -2d + v \geq -1,\ v \leq 1$. □

COROLLARY 7.2. *The vector $x(a, b, c, d)$ belongs to $N(K_n)$ $(n \geq 6)$ if and only if*

$$\begin{gathered} d \leq 1, \quad \pm 2b + d \geq -1, \quad \pm 2c + d \geq -1, \quad \pm 2b + 3d \geq -1, \quad \pm 2c + 3d \geq -1, \\ \pm a \pm b \pm c \geq -1, \quad \pm a \pm 3b \pm 3c + 3d \geq -2, \\ \pm 3a \pm 5b \pm 9c + 6d \geq -5, \quad \pm 3a \pm 9b \pm 5c + 6d \geq -5, \end{gathered} \tag{7.7}$$

where in lines 2 and 3 of the above system there is an even number of minus signs (e.g., $a + b + c \geq -1,\ -a - b + c \geq -1$, etc.). The vector $x(a, c)$ belongs to $N(K_n)$ $(n \geq 6)$ if and only if a, c satisfy

$$\pm 2a + c \geq -1, \quad \pm 2a + 3c \geq -1, \quad \pm 12a + 11c \geq -5, \quad -\tfrac{1}{5} \leq c \leq 1. \tag{7.8}$$

Proof. We saw above that $x = x(a, b, c, d) \in N(K_n)$ if and only if $(1, x) = Ye_0$ for some matrix $Y \in M(\mathrm{MET}(K_n))$ having pattern $(a, b, c, d, x, y, z, u, v)$ for some x, y, z, u, v. Using the computer code **cdd+** of Fukuda [11] for polyhedral computations, we have verified that the projection on the subspace indexed by the variables a, b, c, d of the polytope defined by linear system (7.6) is described by linear system (7.7). One can then verify that for $a = b$ and $c = d$, system (7.7) is equivalent to (7.8). □

We now characterize membership in $N'(K_n)$ for a vector with pattern (a, c).

LEMMA 7.3. *Let $Y \in \mathcal{Y}_n$ with pattern (a, c, x, u) and $n \geq 6$. Then $Y \in M'(\mathrm{MET}(K_n))$ if and only if a, c, x, u satisfy the linear inequalities*

$$\begin{gathered} -2c + u \geq -1, \quad 2c - 3u \geq -1, \quad 10c + 5u \geq -1, \quad 2a - 2x - u \geq -1, \\ -2a + 2x - u \geq -1, \quad 4a + 6c + 4x + u \geq -1, \quad -4a + 6c - 4x + u \geq -1, \\ 2a + 4c + 6x + 3u \geq -1, \quad -2a + 4c - 6x + 3u \geq -1, \end{gathered} \tag{7.9}$$

as well as $6c + 9u \geq -1$ when $n \geq 7$.

Proof. By definition, $Y \in M'(\mathrm{MET}(K_n))$ if and only if, for all $1 \leq i < j < k \leq n$, the vector $Y(e_0 \pm e_{ij} \pm e_{ik} \pm e_{jk})$ (with 0 or 2 minus signs) satisfies all triangle inequalities. By symmetry, it suffices to consider the two cases when $ijk = 123$ or 234. Consider first the case when $ijk = 123$. Due to symmetry, it suffices to consider the triangle inequalities for the vectors $x := Y(e_0 + e_{12} + e_{13} + e_{23})$, $y := Y(e_0 + e_{12} - e_{13} - e_{23})$, and $z := Y(e_0 - e_{12} - e_{13} + e_{23})$, based on the triples 145 and 456 (we also use the fact that $x_{12} = x_{13} = x_{23} = x_0$, $y_{13} = y_{23} = -y_{12} = -y_0$, and $z_{12} = z_{13} = -z_{23} = -z_0$). The triangle inequalities for x based on triple 145 are equivalent to

$$\text{(a)} \qquad 4a + 6c + 4x + u \geq -1, \quad 2a - 2x - u \geq -1, \quad -2c + u \geq -1,$$

and those based on triple 456 give the new relation

$$\text{(b)} \qquad 2a + 4c + 6x + 3u \geq -1.$$

The triangle inequalities for y based on triples 145 and 456 yield, respectively,

$$\text{(c)} \qquad -2a + 2x - u \geq -1, \qquad 2c - 3u \geq -1.$$

The triangle inequalities for z based on triples 145 and 456 give, respectively, the relations:

$$\text{(d)} \qquad -4a + 6c - 4x + u \geq -1, \qquad -2a + 4c - 6x + 3u \geq -1.$$

Consider now the case when $ijk = 234$. Due to symmetry, it suffices to look at the triangle inequalities for the vectors $x := Y(e_0 + e_{23} + e_{24} + e_{34})$ and $y := Y(e_0 - e_{23} - e_{24} + e_{34})$, based on the triples 125, 156, 256, and 567 (the last occurring only for $n \geq 7$). The triangle inequalities for x based on triples 125 and 156 give no new condition; those for triple 256 give the condition

$$\text{(e)} \qquad 10c + 5u \geq -1,$$

and, when $n \geq 7$, those for triple 567 yield

$$\text{(f)} \qquad 6c + 9u \geq -1.$$

No new condition is obtained when looking at the triangle inequalities for y. The inequalities from (a)–(f) are those from (7.9). □

COROLLARY 7.4. *For $n = 6$, $x(a, c) \in N'(K_n)$ if and only if*

$$\text{(7.10)} \qquad \pm 2a + c \geq -1, \quad \pm 5a + 5c \geq -2, \quad -\frac{1}{5} \leq c \leq 1,$$

and, for $n \geq 7$, $x(a, c) \in N'(K_n)$ if and only if a, c satisfy (7.10) *together with the inequalities* $\pm 18a + 15c \geq -7$.

Proof. We have verified (using the computer program **cdd+** [11]) that the projection on the subspace indexed by the variables a and c of the polytope defined by the linear system (7.9) (resp., (7.9) together with $6c + 5u \geq -1$) is described by the linear system (7.10) (resp., (7.10) together with $\pm 18a + 15c \geq -7$). □

We will also need to check whether a matrix $Y \in \mathcal{Y}_n$ is sdp. For concrete examples this can be checked using a computer. However, for a matrix Y with simplified pattern (a, c, x, u) one can explicitly describe the conditions on a, c, x, u ensuring $Y \succeq 0$. Indeed, the positive semidefiniteness of Y can be reformulated as the positive semidefiniteness of some smaller matrix Z whose eigenvalues can be computed because Z belongs to an association scheme. Details will be given in the appendix.

7.2. Proof of Proposition 5.2. We show here the (non)validity of the gap inequality for $b_n = (n - 4, 1, \ldots, 1) \in \mathbf{Z}^n$ for the relaxations $\nu(K_n)$ ($\nu = N, \ldots, N'_+$). Validity over $\nu(K_n)$ means that $B_n \geq \rho_n := -\frac{1}{2}(n^2 - 7n + 14)$, where B_n is defined in (7.2) (with $k = 1$); note that $\rho_6 = -4$, $\rho_7 = -7$, $\rho_8 = -11$. As the program (7.2) admits an optimum solution x having some pattern (a, c) we can, using the results from the preceding subsection, reformulate (7.2) as a program in the variables a and c. In particular, for $\nu = N'$ and $n = 6$, (7.2) can be reformulated as

$$\min(10a + 10c \mid \ a, c \text{ satisfy (7.10)}),$$

and, for $\nu = N'$ and $n = 7$, (7.2) is reformulated as

$$\min(18a + 15c \mid \ a, c \text{ satisfy (7.10) and } \pm 18a + 15c \geq -7).$$

Hence we deduce that the gap inequality for b_n is valid for $N'(K_n)$ when $n = 6, 7$.

We now show nonvalidity for $N'(K_n)$ ($n \geq 8$) and $N_+(K_n)$ ($n \geq 6$). We first observe that it suffices to consider the two bottom cases: $n = 8$ for N' and $n = 6$

for N_+. Indeed, the gap inequality for $b_n = (n-4, 1, \dots, 1) \in \mathbf{Z}^n$ coincides with the inequality obtained from the gap inequality for $b_{n+1} = (n-3, 1, \dots, 1) \in \mathbf{Z}^{n+1}$ by anticollapsing the nodes 1 and $n+1$. Therefore, if $x \in \nu(K_n)$ violates the gap inequality for b_n, then by taking successive (-1)-extensions of x we construct a point $y \in \nu(K_m)$ violating the gap inequality for b_m for any $m \geq n+1$.

Let $a := -\frac{2}{3}$ and $c := \frac{1}{3}$. Then $x(a,c) \in N'(K_n)$ for any $n \geq 7$, and $(n-4)(n-1)a + \binom{n-1}{2}c < \rho_n$ for any $n \geq 8$. This shows that the gap inequality for b_n is not valid for $N'(K_n)$ for $n \geq 8$.

Let $a := -\frac{5}{12}$, $c := \frac{1}{120}$, $x := \frac{11}{45}$, $u := -\frac{29}{90}$. Then $x(a,c) \in N_+(K_6)$. Indeed, the matrix $Y \in \mathcal{Y}_6$ with simplified pattern (a,c,x,u) belongs to $M_+(\mathrm{MET}(K_6))$; that is, a,c,x,u satisfy (7.6) and (A.2). (Note that $\lambda_0(X) = 0$ in (A.2).) As $10a + 10c = -\frac{49}{12} < -4$, $x(a,c)$ violates the gap inequality for b_6. We have found those values of a,c,x,u with the help of the software package SDPPACK [1]. Using SDPPACK, we have solved the semidefinite programming problem

$$\min(10a + 10c \mid Y \in M_+(\mathrm{MET}(K_6)) \text{ having some pattern } (a,c,x,u))$$

and found that the optimum is attained at the above values of a,c,x,u. (This is a problem in dimension $1 + \binom{6}{2} = 16$ with $\binom{16}{2} - 4 + 14 + 16 = 146$ linear (in)equalities; indeed, one can replace the $2\binom{6}{2} \times 4\binom{6}{3} = 2400$ triangle inequalities expressing $Y \in M(\mathrm{MET}(K_6))$ by the 14 linear inequalities from (7.6).)

Note that $\min(10a + 10c \mid x(a,c) \in N(K_6)) = -\frac{30}{7}$, attained at $a = -\frac{2}{7}$, $c = -\frac{1}{7}$. This again shows that the gap inequality for b_6 is not valid for $N(K_6)$ or, moreover, for the strict inclusion $N_+(K_6) \subset N(K_6)$. The following result has been referred to earlier in the paper.

LEMMA 7.5. *Although it is not valid for $N(\mathrm{MET}(K_6))$, the gap inequality for $(-2,1,1,1,1,1)$ is valid for $N(\mathrm{MET}(K_5^\nabla)) \cap \mathcal{L}_{K_5}$ (assigning -2 to the apex node).*

Proof. Indeed, $x(a,c)$ belongs to $\mathcal{L}_{K_5}$ if and only if $c = 2a - 1$. Then $x(a,c) \in N(\mathrm{MET}(K_6))$ implies that $-12a + 11c \geq -5$ and thus $10a \geq 6$; that is, $-10a + 10c \geq -4$. □

We now show that the gap inequality for b_n is valid for $N^{n-5}(K_n)$ for $n \geq 7$. Again it suffices to show the result for the bottom case $n = 7$, as the general result follows using induction. (Indeed, consider $b_{n+1} = (n-3, 1, \dots, 1) \in \mathbf{Z}^{n+1}$. Anticollapsing of nodes 1 and $n+1$ yields the gap inequality for b_n, which is valid for $N^{n-5}(K_n)$ by the induction assumption, while collapsing of these two nodes yields the gap inequality for $(n-2, 1, \dots, 1) \in \mathbf{Z}^n$, which is valid for $\mathrm{MET}(K_n)$ (as it is a sum of triangle inequalities). Therefore we deduce, using Proposition 4.10, that the gap inequality for b_{n+1} is valid for $N^{n-4}(K_{n+1})$.) Our task is now to show that

$$\min(18a + 15c \mid x(a,c) \in N^2(K_7)) \geq -7.$$

For this we need to characterize when $x(a,c) \in N^2(K_7)$. By definition, $x(a,c) \in N^2(K_7)$ if and only if $(1, x(a,c)) = Ye_0$ for some matrix $Y \in M(N(K_7))$ with simplified pattern (a,c,x,u) for some x,u. Due to symmetry, $Y \in M(N(K_7))$ if and only if $Y(e_0 \pm e_{12})$, $Y(e_0 \pm e_{23}) \in N(K_7)$. Note that the vector $Y(e_0 + e_{12})$ is the 1-extension of a vector in $\mathbf{R}^{E_6}$ with pattern $(\frac{a+c}{1+a}, \frac{c+x}{1+a})$; the vector $Y(e_0 - e_{12})$ is the (-1)-extension of $x(\frac{a-c}{1-a}, \frac{c-x}{1-a}) \in \mathbf{R}^{E_6}$; the vector $Y(e_0 + e_{23})$ is the 1-extension of $x(\frac{2a}{1+c}, \frac{a+x}{1+c}, \frac{2c}{1+c}, \frac{c+u}{1+c}) \in \mathbf{R}^{E_6}$; the vector $Y(e_0 - e_{23})$ is the (-1)-extension of $x(0, 0, \frac{a-x}{1-c}, \frac{c-u}{1-c}) \in \mathbf{R}^{E_6}$. Using Corollary 7.2, we find that $x(\frac{a+c}{1+a}, \frac{c+x}{1+a})$, $x(\frac{a-c}{1-a}, \frac{c-x}{1-a})$

belong to $N(K_6)$ if and only if

$$\begin{array}{c} a-c-x\geq -1,\quad -a-c+x\geq -1,\quad 3a+3c+x\geq -1,\quad -3a+3c-x\geq -1,\\ 17a+23c+11x\geq -5,\quad -17a+23c-11x\geq -5,\quad 7a-c-11x\geq -5,\\ -7a-c+11x\geq -5,\quad 3a+5c+3x\geq -1,\quad -3a+5c-3x\geq -1,\\ a+c-3x\geq -1,\quad -a+c+3x\geq -1,\quad a+5c+5x\geq -1,\quad -a+5c-5x\geq -1. \end{array} \tag{7.11}$$

Moreover, $x(\frac{2a}{1+c}, \frac{a+x}{1+c}, \frac{2c}{1+c}, \frac{c+u}{1+c}) \in N(K_6)$ if and only if

$$\begin{array}{c} -\frac{1}{3}\leq u\leq 1,\quad 2a+2c+2x+u\geq -1,\quad -2a+2c-2x+u\geq -1,\\ 2a+4c+2x+3u\geq -1,\quad -2a+4c-2x+3u\geq -1,\quad 6c+u\geq -1,\\ -2c+u\geq -1,\quad 8c+3u\geq -1,\quad 3a+3c+x\geq -1,\quad -3a+3c-x\geq -1,\\ a-c-x\geq -1,\quad -a-c+x\geq -1,\quad 5a+11c+3x+3u\geq -2,\\ -5a+11c-3x+3u\geq -2,\quad -a-c-3x+3u\geq -2,\quad a-c+3x+3u\geq -2,\\ 11a+29c+5x+6u\geq -5,\quad -11a+29c-5x+6u\geq -5,\quad a-7c-5x+6u\geq -5,\\ -a-7c+5x+6u\geq -5,\quad 15a+21c+9x+6u\geq -5,\quad -15a+21c-9x+6u\geq -5,\\ -3a+c-9x+6u\geq -1,\quad 3a+c+9x+6u\geq -5. \end{array} \tag{7.12}$$

Finally, after noting that $x(0,0,x,u) \in N(K_6)$ if and only if $-\frac{1}{3} \leq u \leq 1$, $-1 \leq x \leq 1$, $\pm 2x + u \geq -1$, $\pm 2x + 3u \geq -1$, we find that $x(0,0,\frac{a-x}{1-c}, \frac{c-u}{1-c}) \in N(K_6)$ if and only if

$$\begin{array}{c} -a-c+x\geq -1,\quad a-c-x\geq -1,\quad -2c+u\geq -1,\quad 2c-3u\geq -1,\\ 2a-2x-u\geq -1,\quad -2a+2x-u\geq -1,\quad 2a+2c-2x-3u\geq -1,\\ -2a+2c+2x-3u\geq -1. \end{array} \tag{7.13}$$

Using a computer, we verified that the minimum value of $18a+15c$ subject to a, c, x, u satisfying the linear system (7.11), (7.12), and (7.13) is equal to -7 (attained at $a = -\frac{1}{3}$, $c = -\frac{1}{15}$, $x = \frac{1}{5}$, $u = -\frac{1}{15}$). This shows that the gap inequality for b_7 is valid for $N^2(K_7)$.

7.3. Proof of Propositions 5.3 and 5.4. We begin by showing that the gap inequality for $c_n = (1, \ldots, 1) \in \mathbf{Z}^n$ is not valid for $N'_+(K_n)$ for $n \geq 7$ odd. First let $n = 7$ and set $a = c := -\frac{11}{70}$ and $x = u := \frac{4}{35}$. Then the matrix $Y \in \mathcal{Y}_7$ with pattern (a, c, x, u) belongs to $M'_+(K_7)$, because a, c, x, u satisfy (7.9) and (A.1) (for $n = 7$). Hence $x(a, a)$ belongs to $N'_+(K_7)$ and violates the gap inequality for c_7 as $21a < -3$. We extend the result for any odd $n \geq 7$ by induction. Suppose $x \in N'_+(K_n)$ violates the gap inequality for c_n for some odd $n \geq 7$. For $\epsilon = \pm 1$, the ϵ-extension x^ϵ of x belongs to $N'_+(K_{n+1})$, and thus $\hat{x} := \frac{1}{2}(x^1 + x^{-1}) \in N'_+(K_{n+1})$ with $\hat{x}_{i,n+1} = 0$ $(1 \leq i \leq n)$ and $\hat{x}_{ij} = x_{ij}$ $(ij \in E_n)$. Consider now the (-1)-extension y of $\hat{x}$ defined by $y_{n+1,n+2} = -1$. Then $y \in N'_+(K_{n+2})$ and violates the gap inequality for c_{n+2}. This proves the first part of Proposition 5.3 and the strict inclusion $\mathrm{CUT}(K_n) \subset N'_+(K_n)$ $(n \geq 7)$.

We now show that the gap inequality for c_n is not valid for $N^2(K_n)$ for odd $n \geq 7$. As observed above, it suffices to consider the case $n = 7$. We show that

$$\min(21a \mid x(a, a) \in N^2(K_7)) < -3.$$

Using the results from the preceding subsection, we find that $x(a, a) \in N^2(K_7)$ if and only if there exists $x \in \mathbf{R}$ satisfying $x(\frac{2a}{1+a}, \frac{a+x}{1+a})$, $x(0, \frac{a-x}{1-a}) \in N(K_6)$, which in turn is equivalent to the following linear system:

$$\begin{array}{c} -\frac{1}{3}\leq x\leq 1,\quad -2a+x\geq -1,\quad 6a+x\geq -1,\\ 40a+11x\geq -5,\quad -8a+11x\geq -5,\\ 8a+3x\geq -1,\quad 2a-3x\geq -1,\\ 6a+5x\geq -1,\quad 4a-5x\geq -1. \end{array} \tag{7.14}$$

One can verify that the minimum value of a for which (7.14) holds is $-\frac{9}{61}$ (attained at $a = -\frac{9}{61}$, $x = \frac{5}{61}$), and thus

$$x(a,a) \in N^2(K_7) \Longleftrightarrow -\frac{9}{61} \cdot 21 \leq a \leq 1.$$

As $-\frac{9}{61} \cdot 21 < -3$, we deduce that the gap inequality for c_7 is not valid for $N^2(K_7)$.

Finally we prove Proposition 5.4. The equality $\mathrm{CUT}(K_n) = N(K_n)$ $(n \leq 5)$ follows from Corollary 4.4, and $\mathrm{CUT}(K_6) = N'(K_6) \subset N_+(K_6)$ from Proposition 5.2. We now verify the strict inclusions $N'_+(K_n) \subset N_+(K_n) \subset N(K_n)$ for $n \geq 6$. It suffices to check them for $n = 6$; the first one follows from the above. For the second one note that $x(-\frac{2}{7}, -\frac{1}{7}) \in N(K_6) \setminus N_+(K_6)$. Indeed, if $x \in N_+(K_6)$, then there exist x, u for which the matrix Y with pattern $(-\frac{2}{7}, -\frac{1}{7}, x, u)$ belongs to $M_+(K_6)$. The inequalities $a + 2b + 2c + d + x \geq -1$ and $-a + 3d - 3x \geq -1$ from (7.6) imply that $x = \frac{2}{7}$, and the inequalities $3c + 3d + u \geq -1$ and $2b - c + d - 2z - u \geq -1$ imply that $u = -\frac{1}{7}$ (we have here $a = b, c = d, x = y = z, u = v$). However, the matrix Y is not sdp since the eigenvalue λ_0 (from (A.2)) is negative.

Appendix. Positive semidefinite matrices with a simplified pattern. We will use the following standard result about Schur complements (see, e.g., [15]).

LEMMA A.1. *Let* $X = \left(\begin{smallmatrix} A & B^T \\ B^T & C \end{smallmatrix}\right)$ *be a symmetric matrix. If* A *is nonsingular, then*

$$X \succeq 0 \Longleftrightarrow A \succeq 0 \quad \text{and} \quad C - B^T A^{-1} B \succeq 0.$$

The matrix $C - B^T A^{-1} B \succeq 0$ *is known as the* Schur complement *of* A *in* X.

	0	12	13	14	15	16	23	24	25	26	34	35	36	45	46	56
0	1	a	a	a	a	a	c	c	c	c	c	c	c	c	c	c
12	a	1	c	c	c	c	a	a	a	a	x	x	x	x	x	x
13	a	c	1	c	c	c	a	x	x	x	a	a	a	x	x	x
14	a	c	c	1	c	c	x	a	x	x	a	x	x	a	a	x
15	a	c	c	c	1	c	x	x	a	x	x	a	x	a	x	a
16	a	c	c	c	c	1	x	x	x	a	x	x	a	x	a	a
23	c	a	a	x	x	x	1	c	c	c	c	c	c	u	u	u
24	c	a	x	a	x	x	c	1	c	c	c	u	u	c	c	u
25	c	a	x	x	a	x	c	c	1	c	u	c	u	c	u	c
26	c	a	x	x	x	a	c	c	c	1	u	u	c	u	c	c
34	c	x	a	a	x	x	c	c	u	u	1	c	c	c	c	u
35	c	x	a	x	a	x	c	u	c	u	c	1	c	c	u	c
36	c	x	a	x	x	a	c	u	u	c	c	c	1	u	c	c
45	c	x	x	a	a	x	u	c	c	u	c	c	u	1	c	c
46	c	x	x	a	x	a	u	c	u	c	c	u	c	c	1	c
56	c	x	x	x	a	a	u	u	c	c	u	c	c	c	c	1

FIG. A.1. *A matrix* $Y \in \mathcal{Y}_6$ *with simplified pattern* (a, c, x, u).

Let $Y \in \mathcal{Y}_n$ with simplified pattern (a, c, x, u) (i.e., $a = b$, $c = d$, $x = y = z$, $u = v$) and let Z denote the Schur complement in Y of its $(0,0)$-entry. Suppose first that $a = c$ and $x = u$. Then Z has the property that the value of its (ij, hk)th entry

depends only on whether the pairs ij and hk meet. Let A_n (resp., B_n) denote the symmetric matrix indexed by E_n whose entries are all equal to 0, except entry (ij, hk) equal to 1 if $|\{i,j\} \cap \{h,k\}| = 1$ (resp., $= 0$). Then

$$Z = (1-a^2)I_{d_n} + (a-a^2)A_n + (x-a^2)B_n$$

(where I_{d_n} is the identity matrix of order d_n). The matrices A_n and B_n commute (they are the adjacency matrices of the Johnson scheme $J(n,2)$) and thus have a common basis of eigenvectors. From this it follows that a matrix $X = \alpha A_n + \beta B_n + \gamma I_{d_n}$ has three distinct eigenvalues

$$\lambda_0(X) = 2(n-2)\alpha + \tbinom{n-2}{2}\beta + \gamma, \quad \lambda_1(X) = -2\alpha + \beta + \gamma,$$
$$\lambda_3(X) = (n-4)\alpha - (n-3)\beta + \gamma.$$

Therefore we deduce that $Y \succeq 0$ if and only if

$$\begin{aligned} &\lambda_0(Z) = 2(n-2)(a-a^2) + \tbinom{n-2}{2}(x-a^2) + 1 - a^2 \geq 0, \\ &\lambda_1(Z) = -2a + x + 1 \geq 0, \quad \lambda_2(Z) = (n-4)a - (n-3)x + 1 \geq 0. \end{aligned} \tag{A.1}$$

In the general case, the matrix Z is not of the form $\alpha A_n + \beta B_n + \gamma_{d_n}$. Let Z_1 be its principal submatrix indexed by $\{12, \ldots, 1n\}$; its eigenvalues are $1-c$ and $1+(n-2)c-(n-1)a^2$. If $1-c \neq 0$ and $1+(n-2)c-(n-1)a^2 \neq 0$, we can define the Schur complement X of Z_1 in Z, which turns out to be of the form $\alpha A_{n-1} + \beta B_{n-1} + \gamma I_{d_{n-1}}$, and whose eigenvalues are therefore computable. We mention the result only in the case $n = 6$: Assuming that $c \neq 1,\ 1 + 4c - 5a^2 \neq 0$, $Y \succeq 0$ if and only if

$$\begin{aligned} &c \leq 1, \quad 1 + 4c - 5a^2 \geq 0, \\ &\lambda_0(X) = 1 + 6c - 10c^2 + 3u - 2\tfrac{(2a+3x-5ac)^2}{1+4c-5a^2} \geq 0, \\ &\lambda_1(X) = 1 - 2c + u \geq 0, \quad \lambda_2(X) = 1 + c - 2u - 3\tfrac{(a-x)^2}{1-c} \geq 0. \end{aligned} \tag{A.2}$$

REFERENCES

[1] F. Alizadeh, J.-P. Haeberly, M.V. Nayakkankuppam, M.L. Overton, and S. Shmieta, *SDP-pack User's Guide—Version* 0.9 *Beta,* Technical report TR1997-737, Courant Institute of Mathematical Sciences, New York University, New York, 1997.

[2] M.F. Anjos and H. Wolkowicz, *Strengthened semidefinite relaxations via a second lifting for the max-cut problem,* Discrete Appl. Math., to appear.

[3] E. Balas, S. Ceria, and G. Cornuéjols, *A lift-and-project cutting plane algorithm for mixed* 0-1 *programs,* Math. Programming, 58 (1993), pp. 295–324.

[4] F. Barahona, *The max-cut problem on graphs not contractible to* K_5*,* Oper. Res. Lett., 2 (1983), pp. 107–111.

[5] F. Barahona, *On the computational complexity of Ising spin glass models,* J. Phys. A, 15 (1982), pp. 3241–3253.

[6] F. Barahona, *On cuts and matchings in planar graphs,* Math. Programming, 60 (1993), /linebreak pp. 53–68.

[7] F. Barahona and A.R. Mahjoub, *On the cut polytope,* Math. Programming, 36 (1986), /linebreak pp. 157–173.

[8] W. Cook and S. Dash, *On the matrix-cut rank of polyhedram,* Math. Oper. Res., 26 (2001), pp. 19–30.

[9] M.E. Tylkin (= M. Deza), *On Hamming geometry of unitary cubes* (in Russian), Dokl. Akad. Nauk SSSR, 134 (1960), pp. 1037–1040. (English translation in Cybernetics and Control Theory, 134 (1961), pp. 940–943.)

[10] M. Deza and M. Laurent, *Geometry of Cuts and Metrics,* Algorithms Combin. 15, Springer-Verlag, Berlin, 1997.

[11] K. Fukuda, *cdd/cdd+ computer codes for polyhedral computations*, available online from http://www.ifor.math.ethz.ch/staff/fukuda/fukuda.html.

[12] M.X. Goemans and L. Tunçel, *When does the positive semidefiniteness constraint help in lifting procedures?,* Math. Oper. Res., to appear.
[13] M.X. Goemans and D.P. Williamson, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming,* J. ACM, 42 (1995), pp. 1115–1145.
[14] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Algorithms Combin. 2, Springer-Verlag, Berlin, 1988.
[15] R.A. Horn and C.R. Johnson, *Topics in Matrix Analysis,* Cambridge University Press, Cambridge, England, 1991.
[16] J.B. Lasserre, *Optimality Conditions and LMI Relaxations for* 0–1 *Programs,* Technical report 00099, 2000.
[17] J.B. Lasserre, *An explicit exact SDP relaxation for nonlinear* $0-1$ *programs,* in Lecture Notes in Comput. Sci. 2081, Springer, New York, 2001, pp. 293–303.
[18] M. Laurent, *A Comparison of the Sherali-Adams, Lovász-Schrijver and Lasserre Relaxations for* 0–1 *Programming,* Report PNA-R0108, CWI, Amsterdam, The Netherlands, 2001.
[19] M. Laurent and S. Poljak, *Gap inequalities for the cut polytope,* European J. Combin., 17 (1996), pp. 233–254.
[20] M. Laurent, S. Poljak, and F. Rendl, *Connections between semidefinite relaxations of the max-cut and stable set problems,* Math. Programming, 77 (1997), pp. 225–246.
[21] C. Lemaréchal and F. Oustry, *Semidefinite Relaxations and Lagrangian Duality with Application to Combinatorial Optimization,* Rapport de Recherche 3710, INRIA Rhône-Alpes, Montbonnot St. Martin, France, 1999.
[22] L. Lovász and A. Schrijver, *Cones of matrices and set-functions and* 0–1 *optimization,* SIAM J. Optim., 1 (1991), pp. 166–190.
[23] S. Poljak and F. Rendl, *Nonpolyhedral relaxation of graph-bisection problems,* SIAM J. Optim., 5 (1995), pp. 467–487.
[24] S. Poljak, F. Rendl, and H. Wolkowicz, *A recipe for semidefinite relaxation for* $(0,1)$*-quadratic programming,* J. Global Optim., 7 (1995), pp. 51–73.
[25] M. Padberg, *The Boolean quadric polytope: Some characteristics, facets and relatives,* Math. Programming, 45 (1989), pp. 139–172.
[26] N. Robertson and P.D. Seymour, *Graph Minors* XX. *Wagner's Conjecture,* manuscript, 1988.
[27] I.J. Schoenberg, *On certain metric spaces arising from Euclidean spaces by a change of metric and their imbedding in Hilbert spaces,* Ann. of Mathematics, 38 (1937), pp. 787–793.
[28] H.D. Sherali and W.P. Adams, *A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems,* SIAM J. Discrete Math., 3 (1990), pp. 411–430.
[29] T. Stephen and L. Tunçel, *On a representation of the matching polytope via semidefinite liftings,* Math. Oper. Res., 24 (1999), pp. 1–7.

SIAM J. OPTIM.
Vol. 12, No. 2, pp. 376–406

AN AUGMENTED LAGRANGIAN FUNCTION WITH IMPROVED EXACTNESS PROPERTIES*

GIANNI DI PILLO† AND STEFANO LUCIDI†

Abstract. In this paper we introduce a new exact augmented Lagrangian function for the solution of general nonlinear programming problems. For this Lagrangian function a complete equivalence between its unconstrained minimization on an open set and the solution of the original constrained problem can be established under mild assumptions and without requiring the boundedness of the feasible set of the constrained problem. Moreover we describe an unconstrained algorithmic model which is globally convergent toward KKT pairs of the original constrained problem. The algorithmic model can be endowed with a superlinear rate of convergence by a proper choice of the search direction in the unconstrained minimization, without requiring strict complementarity.

Key words. constrained optimization, nonlinear programming, nonlinear programming algorithms, merit functions, augmented Lagrangian functions

AMS subject classifications. 90C30, 65K05

PII. S1052623497321894

1. Introduction.

The problem considered here is the constrained nonlinear programming problem:

$$\text{(P)} \qquad \begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to (s.t.)} & g(x) \le 0, \end{array}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}^m$ are twice continuously differentiable functions. We denote by

$$\mathcal{F} = \{x \in \mathbb{R}^n : \ g(x) \le 0\}$$

the feasible set of problem (P) and by

$$I_0(x) = \{i : g_i(x) = 0\}$$

the index set of the active constraints at x. The *Lagrangian function* associated with problem (P) is the function $L : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ given by

$$L(x, \lambda) = f(x) + \lambda' g(x).$$

A *Karush–Kuhn–Tucker* (KKT) *pair* for problem (P) is a pair $(\bar{x}, \bar{\lambda}) \in \mathbb{R}^n \times \mathbb{R}^m$ such that

$$\nabla_x L(\bar{x}, \bar{\lambda}) = 0, \quad g(\bar{x})'\bar{\lambda} = 0, \quad \bar{\lambda} \ge 0, \quad g(\bar{x}) \le 0.$$

If the gradients $\nabla g_i(\bar{x})$, $i \in I_0(\bar{x})$, are linearly independent, the pair $(\bar{x}, \bar{\lambda})$ satisfies the KKT *first order necessary conditions* for $\bar{x}$ to be a local solution for problem (P).

*Received by the editors May 27, 1997; accepted for publication (in revised form) February 9, 2001; published electronically November 13, 2001. This work was supported by MURST, National Research Program *Algorithms for Complex Systems Optimization*.
http://www.siam.org/journals/siopt/12-2/32189.html
†Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza," via Buonarroti 12, 00185 Roma, Italy (dipillo@dis.uniroma1.it, lucidi@dis.uniroma1.it).

A KKT pair $(\bar{x}, \bar{\lambda})$ satisfies the *strict complementarity condition* if $\bar{\lambda}_i > 0$ for all $i \in I_0(\bar{x})$.

Given a KKT pair $(\bar{x}, \bar{\lambda})$, we denote by $I_+(\bar{x}, \bar{\lambda})$ the index set of the active constraints with positive multiplier, that is,

$$I_+(\bar{x}, \bar{\lambda}) = \{i \in I_0(\bar{x}) : \bar{\lambda}_i > 0\}.$$

A KKT pair $(\bar{x}, \bar{\lambda})$ satisfies the *strong second order sufficient condition* for $\bar{x}$ to be a strict local solution of problem (P) if

$$w' \nabla_x^2 L(\bar{x}, \bar{\lambda}) w > 0 \qquad \forall w \neq 0 : \nabla g_i(\bar{x})' w = 0, \quad i \in I_+(\bar{x}, \bar{\lambda}).$$

It is clear that the constrained problem (P) is determined by the interaction of two distinct subproblems: the feasibility subproblem and the subproblem of minimizing the objective function. In this paper we are concerned with the definition of a *merit function* able to properly balance the two distinct subproblems.

The definition of merit functions of this kind is of interest both from the theoretical and the computational point of view. In fact, problem (P) can be solved either by resorting to the unconstrained minimization of a suitable merit function or by employing a suitable merit function to enforce the global convergence of algorithms, based on the solution of subproblems of particular structure (such as SQP-type algorithms), that are locally convergent with superlinear convergence rate. We refer to [1, 2, 13, 8, 4, 14] for some basic references on merit functions for nonlinear programming problems.

We say that a merit function enjoys "exactness" properties if it is possible to establish some correspondence between its unconstrained minimizers and the solutions of problem (P).

The initial idea in defining merit functions was to add, to the original objective function, terms penalizing the violation of the constraints. This approach has led to the definition of merit functions which are exact but nondifferentiable or to merit functions which are continuously differentiable but not "exact" in the sense meant before.

The subsequent step has been the introduction of merit functions which characterize "better" the connections between the feasibility subproblem and the minimizing subproblem: namely, functions which consider not only the feasibility but also other characteristics of the constrained minimum points. A practicable choice has been to define merit functions which take the KKT conditions into account. Following this line, two classes of continuously differentiable merit functions have been proposed:

- merit functions which are defined on the same space of the variables of the original constrained problem,
- merit functions which are defined on the product space of the problem variables and of the KKT multipliers.

The functions in the first class penalize the violation of the KKT conditions by making use of multiplier functions, namely, of functions $\lambda(x)$ which yield estimates of the KKT multipliers as functions of the variable x. In general a multiplier function is quite expensive from the computational point of view when the number of constraints is large, which may limit somewhat the applicability of merit functions of this kind.

The functions belonging to the second class can be in turn divided into two subclasses:

- penalty functions (in the extended space) in which the terms that account for the KKT conditions are added to the objective function $f(x)$,

- *augmented Lagrangian functions* in which the terms that account for the KKT conditions are added to the Lagrangian function $L(x, \lambda)$.

An example of a penalty function (in the extended space) has been proposed in [27] in order to globalize a locally convergent Gauss–Newton-type algorithm. Under suitable assumptions, not requiring the strict complementarity, the resulting algorithm has been shown to be globally convergent towards a KKT pair, with a superlinear rate of convergence. However, even if a feasible starting point is known (and, hence, the feasibility subproblem is solved), it is not guaranteed that, at the attained KKT point, the objective function $f(x)$ is decreased.

In this paper we introduce a new augmented Lagrangian function $L_a(x, \lambda; \epsilon)$, where $\epsilon > 0$ is a penalty parameter. This function is continuously differentiable and has level sets that are compact for every value of the penalty parameter ϵ. These features are of basic importance for ensuring global convergence properties to the algorithms that employ it. From the exactness point of view, it is possible to prove that, for sufficiently small values of the penalty parameter ϵ but without requiring that ε goes to zero, every minimum point (KKT point) of the original problem corresponds to a minimum point (stationary point) of $L_a(x, \lambda; \epsilon)$ on $\mathcal{P} \times \mathbb{R}^m$ and conversely, where $\mathcal{P}$ is a given open set containing $\mathcal{F}$. These strong exactness results can be stated under assumptions weaker than all similar assumptions employed before. Under these assumptions, it is possible to propose an algorithmic model, based on the unconstrained minimization of the function L_a, which is globally convergent towards KKT pairs of problem (P). Moreover, if a feasible starting point is known, we can ensure that, at every produced KKT pair, the objective function $f(x)$ is decreased. In the case that it is not possible to guarantee that all the assumptions are satisfied, and even in the case that problem (P) is not feasible, it is possible to characterize some limit points produced by the algorithm with respect to problem (P).

In addition, if the problem functions are three times continuously differentiable, the algorithmic model can be endowed with a superlinear rate of convergence by a proper choice of the search direction in the minimization of L_a. In particular, among others, we discuss search directions that can be evaluated by solving a simple linear system and that provide superlinear convergence towards KKT pairs of problem (P) where the strong second order sufficient condition holds, without requiring that the strict complementarity condition also holds.

The paper is organized as follows. In section 2 we introduce the new augmented Lagrangian function $L_a(x, \lambda; \epsilon)$, and we list the assumptions employed to establish its exactness properties. In section 3 we point out some preliminary properties of L_a. In section 4 we study the correspondence between KKT pairs of problem (P) and stationary points of L_a. In section 5 we study the correspondence between local (global) solutions of problem (P) and local (global) minimum points of L_a. In section 6 we study the local behavior of the generalized Hessian of L_a in a neighborhood of a KKT pair of problem (P), and we establish some additional exactness results. In section 7 we describe an algorithmic model for the solution of problem (P) based on the unconstrained minimization of L_a, and we analyze its global convergence properties. Finally, in section 8 we discuss some search directions that can be employed in the minimization of L_a, in order to get a superlinear rate of convergence towards KKT pairs of problem (P). A numerical experiment to test the algorithmic model is out of the scope of this paper; it is currently being undertaken and will be a matter for future work.

We conclude this section by introducing some basic notation. For any vector v,

we denote by V the diagonal matrix $V = \mathrm{diag}(v_i)$; we denote by v_S the subvector with components v_i, $i \in S$, where S is a given index subset; we denote by $\|v\|$ the Euclidean norm of v. Given two vectors u, v of same dimension, we denote by $\max\{u, v\}$ the vector with components $\max\{u_i, v_i\}$. Given a set $\mathcal{S}$, we denote by $\mathring{\mathcal{S}}$ its interior, by $\bar{\mathcal{S}}$ its closure, and by $\partial\mathcal{S}$ its boundary.

2. The new augmented Lagrangian function. Let $\alpha, s \in \mathbb{R}$ be given scalars such that $\alpha > 0$ and $s \geq 2$. We can consider the open perturbation $\mathcal{P}$ of the feasible set $\mathcal{F}$, defined by

$$\mathcal{P} = \left\{ x \in \mathbb{R}^n : \sum_{i=1}^{m} \max\{g_i(x), 0\}^s < \alpha \right\};$$

it is clear that $\mathcal{F} \subset \mathcal{P}$. Moreover we can introduce the function

$$a(x) = \alpha - \sum_{i=1}^{m} \max\{g_i(x), 0\}^s, \tag{2.1}$$

which takes positive values on $\mathcal{P}$ and which is zero on the boundary $\partial\mathcal{P}$.

Then, let us consider the function

$$p(x, \lambda) = \frac{a(x)}{1 + \|\lambda\|^2}; \tag{2.2}$$

this function is characterized by the following properties:

$$p(x, \lambda) > 0 \quad \forall (x, \lambda) \in \mathcal{P} \times \mathbb{R}^m, \tag{2.3}$$

$$\lim_{x \to \partial\mathcal{P}} p(x, \lambda) = 0, \tag{2.4}$$

$$\lim_{\|\lambda\| \to \infty} p(x, \lambda) = 0. \tag{2.5}$$

Due to (2.3), (2.4), and (2.5), the term $1/p(x, \lambda)$ plays the role of a barrier term that penalizes both the fact that the variable x is too close to the boundary of $\mathcal{P}$ and the fact that the norm of the vector λ is too large.

Now we can define the following augmented Lagrangian function for problem (P):

$$\begin{aligned} L_a(x, \lambda; \epsilon) &= f(x) + \lambda' \max\{g(x), -\epsilon p(x, \lambda)\lambda\} \\ &\quad + \frac{1}{2\epsilon p(x, \lambda)} \| \max\{g(x), -\epsilon p(x, \lambda)\lambda\} \|^2 + \|\nabla g(x)' \nabla_x L(x, \lambda) + G(x)^2 \lambda\|^2, \end{aligned} \tag{2.6}$$

where $\epsilon > 0$ is a penalty parameter and $G = \mathrm{diag}(g_i)$.

We can recognize that $L_a(x, \lambda; \epsilon)$ is an augmented Lagrangian function by the fact that (2.6) can be rewritten in the form

$$\begin{aligned} L_a(x, \lambda; \epsilon) = L(x, \lambda) &+ \frac{1}{2\epsilon p(x, \lambda)} \left[\|g(x)\|^2 - \| \min\{0, g(x) + \epsilon p(x, \lambda)\lambda\} \|^2 \right] \\ &+ \|\nabla g(x)' \nabla_x L(x, \lambda) + G(x)^2 \lambda\|^2. \end{aligned} \tag{2.7}$$

By observing the expression (2.6), we note that $L_a(x, \lambda; \epsilon)$ is obtained by adding the terms $\frac{1}{2\epsilon p(x,\lambda)} \psi_1(x, \lambda; \epsilon)$ and $\psi_2(x, \lambda)$ to the function $f(x)$, where

$$\begin{aligned} \psi_1(x, \lambda; \epsilon) &= 2\epsilon p(x, \lambda) \lambda' \max\{g(x), -\epsilon p(x, \lambda)\lambda\} + \| \max\{g(x), -\epsilon p(x, \lambda)\lambda\} \|^2, \\ \psi_2(x, \lambda) &= \|\nabla g(x)' \nabla_x L(x, \lambda) + G(x)^2 \lambda\|^2. \end{aligned}$$

We refer to [14] for a detailed discussion about the reasons for the presence and the importance of the role of terms similar to ψ_1 and ψ_2 in an augmented Lagrangian function. Here we only point out that these terms guarantee a "smooth" penalization of the violation of the KKT conditions; in fact both are continuously differentiable and satisfy the following:

- $\lim_{\epsilon\to 0} \psi_1(x,\lambda;\epsilon) = \|\max\{g(x),0\}\|^2$;
- if $x \in \mathcal{F}$, then $\psi_1(x,\lambda;\epsilon) = 0$ if and only if $\lambda \geq 0$, $g(x)'\lambda = 0$;
- $\psi_2(x,\lambda)$ is a convex function w.r.t. λ;
- if $(\bar{x},\bar{\lambda})$ is a KKT pair and if the gradients of the active constraints are linearly independent at $\bar{x}$, then $\psi_2(\bar{x},\lambda) = 0$ if and only if $\lambda = \bar{\lambda}$.

Therefore, roughly speaking, the term ψ_1 forces the feasibility, the nonnegativity of λ, and the complementarity condition $g(x)'\lambda = 0$, while the term ψ_2 convexifies w.r.t. λ and penalizes the distance between the variable λ and a KKT multiplier $\bar{\lambda}$.

Looking at the expression of L_a given by (2.7), we can observe that the first two terms in the right-hand side (r.h.s.), with $p(x,\lambda) = 1$, correspond to the usual Hestenes–Powell–Rockafellar augmented Lagrangian function for problem (P); the addition of the third term was already proposed by Di Pillo and Grippo [9] and Lucidi [25]. Hence, the distinguishing element in the expression of the L_a is the presence of the term $p(x,\lambda)$.

This term is most effective in order to provide the function L_a with improved exactness properties under weaker assumptions. We can get an idea of the way this term acts by considering the following simple example problem with unbounded feasible set:

$$\text{(EP)} \qquad \begin{array}{ll} \text{minimize} & x^3 \\ \text{s.t.} & x \geq 0. \end{array}$$

Problem (EP) has the unique global solution $x^* = 0$, with associated multiplier $\lambda^* = 0$. For this problem, the Hestenes–Powell–Rockafellar augmented Lagrangian function L_a^{HPR} is given by

$$L_a^{HPR}(x,\lambda;\epsilon) = x^3 - \lambda x + \frac{1}{2\epsilon}\left[x^2 - \min\{0, -x + \epsilon\lambda\}^2\right],$$

the Di Pillo–Grippo–Lucidi augmented Lagrangian function L_a^{DGL} is given by

$$L_a^{DGL}(x,\lambda;\epsilon) = x^3 - \lambda x + \frac{1}{2\epsilon}\left[x^2 - \min\{0, -x + \epsilon\lambda\}^2\right] + (\lambda - 3x^2 + x^2\lambda)^2,$$

and the augmented Lagrangian function L_a of concern here is given by

$$L_a(x,\lambda;\epsilon) = x^3 - \lambda x + \frac{1}{2\epsilon p(x,\lambda)}\left[x^2 - \min\{0, -x + \epsilon p(x,\lambda)\lambda\}^2\right] + (\lambda - 3x^2 + x^2\lambda)^2,$$

with

$$p(x,\lambda) = \frac{\alpha - \min\{x,0\}^s}{1+\lambda^2}.$$

These functions are plotted in Figures 1, 2, and 3, respectively, for $\epsilon = 0.25$ and $\alpha = 1$, $s = 3$. It appears that the use of an unconstrained minimization algorithm for the solution of problem (EP) is much more reliable if applied to the function L_a. In fact, in this case the algorithm would search for the minimum point of a function with compact level sets, while if applied to L_a^{DGL} it could produce sequences, even

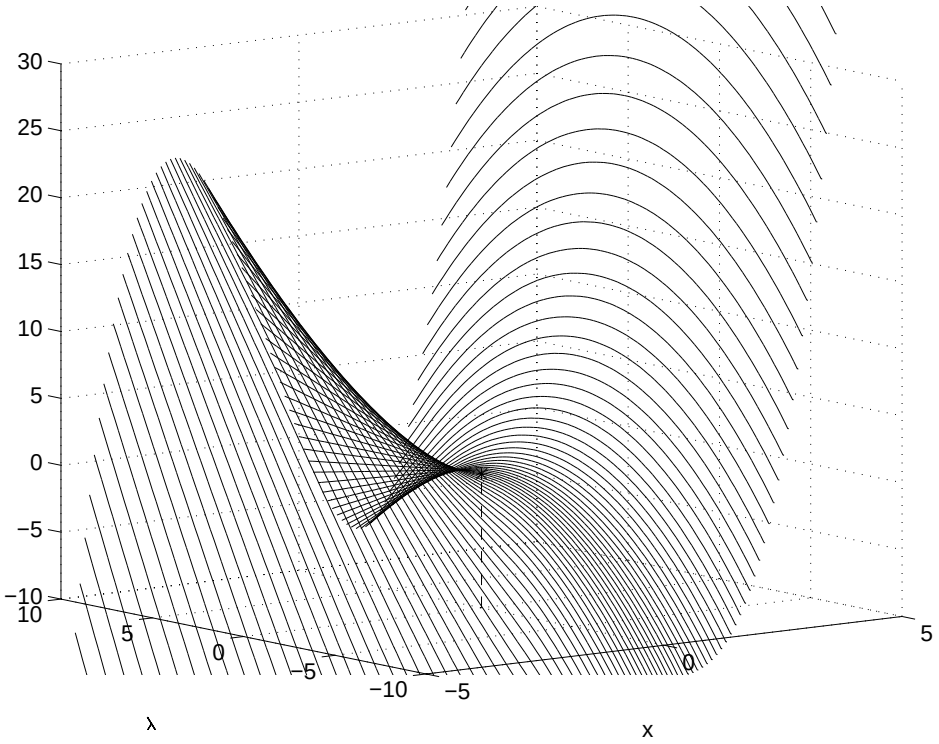


FIG. 1. *The augmented Lagrangian function* L_a^{HPR} *for problem* (EP).

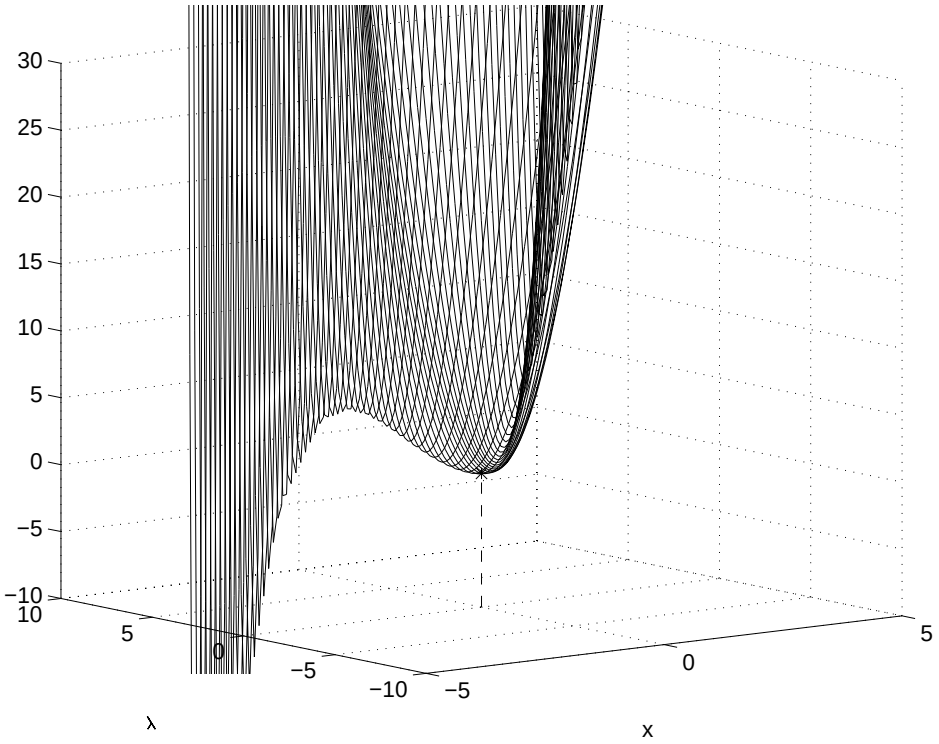


FIG. 2. *The augmented Lagrangian function* L_a^{DGL} *for problem* (EP).

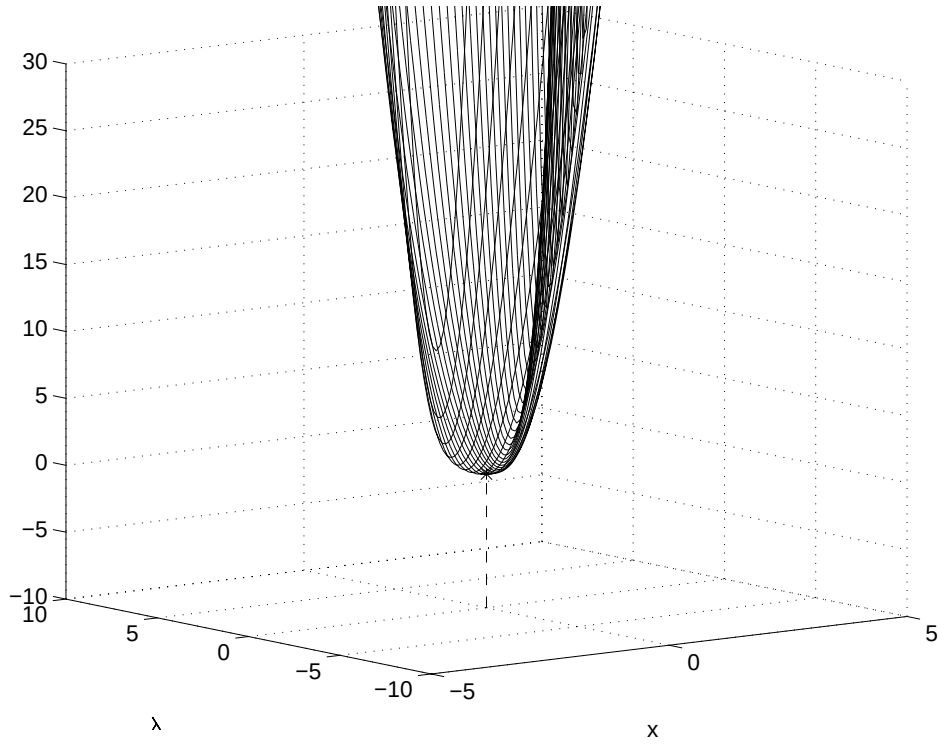


FIG. 3. *The augmented Lagrangian function L_a for problem* (EP).

unbounded, moving out of the region where exactness holds, and if applied to L_a^{HPR} it should search for a saddle point.

From the definition and the differentiability assumptions on f and g, it follows that the function $L_a(x,\lambda;\epsilon)$ is an SC^1 function for all $(x,\lambda) \in \mathcal{P} \times \mathbb{R}^m$, that is, a continuously differentiable function with a semismooth gradient (see [29]). The gradient of L_a is obtained from (2.6) as

$$\begin{aligned}\nabla_x L_a(x,\lambda;\epsilon) &= \nabla_x L(x,\lambda) + \frac{1}{\epsilon p(x,\lambda)} \nabla g(x) \max\{g(x), -\epsilon p(x,\lambda)\lambda\} \\ &\quad + \frac{s}{2\epsilon a(x) p(x,\lambda)} \|\max\{g(x), -\epsilon p(x,\lambda)\lambda\}\|^2 \sum_{i=1}^{m} \nabla g_i(x) \max\{g_i(x), 0\}^{s-1} \qquad (2.8) \\ &\quad + Q(x,\lambda)\left[\nabla g(x)' \nabla_x L(x,\lambda) + G(x)^2 \lambda\right], \\ \nabla_\lambda L_a(x,\lambda;\epsilon) &= \max\{g(x), -\epsilon p(x,\lambda)\lambda\} + \frac{1}{\epsilon a(x)} \|\max\{g(x), -\epsilon p(x,\lambda)\lambda\}\|^2 \lambda \\ &\quad + 2M(x)\left[\nabla g(x)' \nabla_x L(x,\lambda) + G(x)^2 \lambda\right], \qquad (2.9)\end{aligned}$$

where

$$Q(x,\lambda) = 2\left[\nabla_x^2 L(x,\lambda) \nabla g(x) + \sum_{i=1}^{m} \nabla_x^2 g_i(x) \nabla_x L(x,\lambda) e_i' + 2\nabla g(x) G(x) \Lambda\right], \qquad (2.10)$$

$$M(x) = \nabla g(x)' \nabla g(x) + G^2(x), \qquad (2.11)$$

e_i denotes the ith column of the $m \times m$ identity matrix, and $\Lambda = \text{diag}(\lambda_i)$.

Remark 2.1. It is a known result that if the gradients $\nabla g_i(x)$, $i \in I_0(x)$, are linearly independent, then the matrix $M(x)$ given by (2.11) is positive definite, and hence nonsingular (see, for instance, [19, 11]). □

Given a point $(x_0, \lambda_0) \in \mathcal{P} \times \mathbb{R}^m$, we can introduce the level set of L_a:

$$\Omega(x_0, \lambda_0; \epsilon) = \{(x, \lambda) \in \mathcal{P} \times \mathbb{R}^m : L_a(x, \lambda; \epsilon) \leq L_a(x_0, \lambda_0; \epsilon)\}.$$

In this regard, we point out that, given any point $(x_0, \lambda_0) \in \mathbb{R}^n \times \mathbb{R}^m$, it is easy to select values α and s in the definition of $\mathcal{P}$ and L_a such that $x_0 \in \mathcal{P}$.

As we said before, our aim is to solve problem (P) by an unconstrained minimization of L_a on $\mathcal{P} \times R^m$. Therefore we are interested in analyzing the correspondence between stationary points of L_a belonging to $\Omega(x_0, \lambda_0; \epsilon)$ and KKT pairs of problem (P), as well as the correspondence between local (global) minimizers of L_a belonging to $\Omega(x_0, \lambda_0; \epsilon)$ and local (global) solutions of problem (P).

In order to establish these correspondences we distinguish between two cases: the case that x_0 is a feasible point and the case that x_0 is not a feasible point. Indeed, in the first case, since the feasibility subproblem can be considered already solved, we deal with an easier task. This is reflected in the fact that it is possible to simplify the assumptions required in the study of the exactness properties of L_a.

More specifically, we make use of the following assumptions.

Assumption A1. One of the two following conditions is satisfied:

(a) $\bar{\mathcal{P}}$ is a bounded set;

(b) $x_0 \in \mathcal{F}$ and $f(x)$ is coercive on $\bar{\mathcal{P}}$ (that is, for any $\{x_k\} \subseteq \mathcal{P}$ such that $\|x_k\| \to \infty$, we have $f(x_k) \to \infty$).

Assumption A2. For every $x \in \mathcal{F}$ the gradients $\nabla g_i(x)$, $i \in I_0(x)$, are linearly independent.

Assumption A3. One of the two following conditions is satisfied:

(a) At every point $x \in \mathcal{P}/\mathcal{F}$,

$$\sum_{i:g_i(x)>0} c_i(x) \nabla g_i(x) \neq 0, \tag{2.12}$$

where

$$c_i(x) = \left[1 + \frac{s}{2} \frac{\| \max\{g(x), 0\}\|^2 g_i(x)^{(s-2)}}{a(x)}\right] g_i(x). \tag{2.13}$$

(b) $x_0 \in \mathcal{F}$.

In what follows, we assume that Assumption A1 holds everywhere. Assumptions A2 and A3 will be invoked when needed.

As already claimed, the assumptions employed to establish the exactness properties of L_a are weaker than the ones used by all Lagrangian functions defined before (see, e.g., [1, 9, 24]). More specifically, by Assumption A1(b), constrained optimization problems with unbounded feasible sets can be tackled, provided that a feasible point is known and that all the level sets of the objective function in the open set $\mathcal{P}$ are compact (this property is quite similar to the one usually used in the case of unconstrained optimization); by Assumption A2, the linear independence of the gradients of the active constraints is required only in the feasible set, a mild requirement which implies the existence and uniqueness of the KKT multipliers. Assumption A3(a) is a weakening of the Mangasarian–Fromovitz constraint qualification condition. In fact

the Mangasarian–Fromovitz constraint qualification condition (see, e.g, [26]) holds at x if $\sum_{i:g_i(x)\geq 0} c_i \nabla g_i(x) \neq 0$ for all $c_i \geq 0$. Of course, Assumption A3(a) is implied by the positive linear independence of the gradients of the violated constraints; moreover, it is also implied by the assumption that at every point $x \in \mathcal{P}/\mathcal{F}$ the set $\{z \in \mathbb{R}^n : \nabla g_i(x)'z + g_i(x) \leq 0,\ i : g_i(x) \geq 0\}$ is not empty (see [25]), an assumption widely used in the analysis of SQP algorithms. Assumption A3(a) involves the behavior of the constraint functions outside the feasible set and it is connected to the feasibility of the original problem. In fact it is a sufficient condition for the nonemptiness of the feasible set, and it is also necessary in the case of a bounded feasible set given by convex inequalities (see again [25]); therefore, at least for this class of constraints, the condition used is the weakest possible assumption which guarantees that the original constrained problem has a nonempty feasible set.

Finally we remark that Assumptions A1, A2, and A3 are all a priori assumptions on the problem, while, in the analysis of constrained optimization algorithms, a posteriori assumptions on the generated sequences are often required.

3. Preliminary properties. In this section we point out some preliminary properties of the function $L_a(x, \lambda; \epsilon)$. In particular, we establish results on the compactness of the level set $\Omega(x_0, \lambda_0; \epsilon)$.

PROPOSITION 3.1. *For every* $\epsilon > 0$,

(a) *for all KKT pairs* $(\bar{x}, \bar{\lambda})$ *of problem* (P) *it holds that*

$$L_a(\bar{x}, \bar{\lambda}; \epsilon) = f(\bar{x});$$

(b) *for all* $(x, \lambda) \in \mathcal{F} \times \mathbb{R}^m$ *it holds that*

$$L_a(x, \lambda; \epsilon) \leq f(x) + \|\nabla g(x)' \nabla_x L(x, \lambda) + G(x)^2 \lambda\|^2; \tag{3.1}$$

(c) *for all* $(x, \lambda) \in \mathcal{P} \times \mathbb{R}^m$ *it holds that*

$$\begin{aligned} L_a(x, \lambda; \epsilon) \geq f(x) - \frac{\epsilon\alpha}{2} + \frac{1}{2\epsilon a(x)} \|\max\{g(x), -\epsilon p(x, \lambda)\lambda\}\|^2 \\ + \|\nabla g(x)' \nabla_x L(x, \lambda) + G(x)^2 \lambda\|^2. \end{aligned} \tag{3.2}$$

Proof. (a) It can be easily verified that $\max\{g(\bar{x}), -\epsilon p(\bar{x}, \bar{\lambda})\bar{\lambda}\} = 0$ for all KKT pair $(\bar{x}, \bar{\lambda})$. Then (a) follows from (2.6).

(b) By (2.6) we have

$$\begin{aligned} &L_a(x, \lambda; \epsilon) - f(x) - \|\nabla g(x)' \nabla_x L(x, \lambda) + G(x)^2 \lambda\|^2 \\ &= \sum_{i=1}^{m} \left[\lambda_i \max\{g_i(x), -\epsilon p(x, \lambda)\lambda_i\} + \frac{1}{2\epsilon p(x, \lambda)} \max\{g_i(x), -\epsilon p(x, \lambda)\lambda_i\}^2 \right]. \end{aligned} \tag{3.3}$$

Consider the ith term of the summation in (3.3). If the index i is such that $\max\{g_i(x), -\epsilon p(x, \lambda)\lambda_i\} = g_i(x)$, taking into account that $g_i(x) \leq 0$ for all $x \in \mathcal{F}$, we have

$$0 \leq 2\,(g_i(x) + \epsilon p(x, \lambda)\lambda_i) \leq g_i(x) + 2\epsilon p(x, \lambda)\lambda_i,$$

so that we obtain

$$\begin{aligned} &\lambda_i \max\{g_i(x), -\epsilon p(x, \lambda)\lambda_i\} + \frac{1}{2\epsilon p(x, \lambda)} \max\{g_i(x), -\epsilon p(x, \lambda)\lambda_i\}^2 \\ &\qquad = \frac{1}{2\epsilon p(x, \lambda)} \left[g_i(x)^2 + 2\epsilon p(x, \lambda)\lambda_i g_i(x) \right] \leq 0. \end{aligned} \tag{3.4}$$

If the index i is such that $\max\{g_i(x), -\epsilon p(x,\lambda)\lambda_i\} = -\epsilon p(x,\lambda)\lambda_i$, we have

$$\begin{aligned}\lambda_i \max\{g_i(x), -\epsilon p(x,\lambda)\lambda_i\} &+ \frac{1}{2\epsilon p(x,\lambda)} \max\{g_i(x), -\epsilon p(x,\lambda)\lambda_i\}^2 \\ &= -\frac{\epsilon p(x,\lambda)}{2}\lambda_i^2 \le 0. \end{aligned} \tag{3.5}$$

By (3.4) and (3.5), all terms of the summation in (3.3) are not positive for any $x \in \mathcal{F}$ and $\lambda \in \mathbb{R}^m$, and this proves (b).

(c) Recalling (2.2), we can rewrite the function $L_a(x,\lambda;\epsilon)$ in the form

$$\begin{aligned}L_a(x,\lambda;\epsilon) = f(x) &+ \lambda' \max\{g(x), -\epsilon p(x,\lambda)\lambda\} + \frac{1}{2\epsilon a(x)}\|\lambda\|^2 \|\max\{g(x), -\epsilon p(x,\lambda)\lambda\}\|^2 \\ &+ \frac{1}{2\epsilon a(x)}\|\max\{g(x), -\epsilon p(x,\lambda)\lambda\}\|^2 + \|\nabla g(x)'\nabla_x L(x,\lambda) + G(x)^2\lambda\|^2,\end{aligned}$$

from which we have

$$\begin{aligned}L_a(x,\lambda;\epsilon) \ge f(x) &- \|\lambda\| \, \|\max\{g(x), -\epsilon p(x,\lambda)\lambda\}\| \\ &+ \frac{1}{2\epsilon a(x)}\left(1 + \|\lambda\|^2\right) \|\max\{g(x), -\epsilon p(x,\lambda)\lambda\}\|^2 \\ &+ \|\nabla g(x)'\nabla_x L(x,\lambda) + G(x)^2\lambda\|^2.\end{aligned}$$

Now, taking into account that the minimum of the quadratic form $-u + \frac{1}{2\epsilon a(x)}u^2$ is $-\frac{\epsilon a(x)}{2}$ and recalling that $a(x) < \alpha$ holds for $x \in \mathcal{P}$, we have

$$\begin{aligned}L_a(x,\lambda;\epsilon) \ge f(x) - \frac{\epsilon\alpha}{2} &+ \frac{1}{2\epsilon a(x)}\|\max\{g(x), -\epsilon p(x,\lambda)\lambda\}\|^2 \\ &+ \|\nabla g(x)'\nabla_x L(x,\lambda) + G(x)^2\lambda\|^2,\end{aligned} \tag{3.6}$$

which proves (c). □

From (a) and (b) of Proposition 3.1, a first interesting property of the level set Ω follows immediately; in fact, if a feasible point x_0 is known and if λ_0 is properly selected, the set $\Omega(x_0,\lambda_0;\epsilon)$ allows us to localize a subset of the KKT points of problem (P) that have an objective function value smaller than or equal to $f(x_0)$ (namely, the KKT points $(\bar{x},\bar{\lambda})$ such that $f(\bar{x}) \le P(x_0,\lambda_0;\epsilon) \le f(x_0)$).

PROPOSITION 3.2. *Let x_0 be a feasible point and λ_0 be a vector such that*

$$\nabla g(x_0)'\nabla f(x_0) + M(x_0)\lambda_0 = 0, \tag{3.7}$$

where $M(x_0)$ is given by (2.11); *then, for every $\epsilon > 0$, any KKT pair $(\bar{x},\bar{\lambda})$ of problem* (P) *contained in $\Omega(x_0,\lambda_0;\epsilon)$ is such that $f(\bar{x}) \le f(x_0)$.*

In the next two theorems we state some compactness properties of the level set Ω.

THEOREM 3.3. *For every $\epsilon_M > 0$, there exists a compact set $\mathcal{C} \subset \mathbb{R}^n$ such that $\Omega(x_0,\lambda_0;\epsilon) \subseteq \mathcal{C} \times \mathbb{R}^m$ for all $\epsilon \in (0,\epsilon_M]$.*

Proof. Recalling Assumption A1, if $\bar{\mathcal{P}}$ is compact, then $\mathcal{C} = \bar{\mathcal{P}}$. Otherwise we know that x_0 is feasible and that the function $f(x)$ is coercive on $\bar{\mathcal{P}}$. Now let $(x,\lambda) \in \Omega(x_0,\lambda_0;\epsilon)$; recalling (3.1) we can write

$$L_a(x,\lambda;\epsilon) \le L_a(x_0,\lambda_0;\epsilon) \le f(x_0) + \|\nabla g(x_0)'\nabla_x L(x_0,\lambda_0) + G(x_0)^2\lambda_0\|^2, \tag{3.8}$$

and from (3.2), (3.8) and the fact that $\epsilon \in (0, \epsilon_M]$ we have

$$f(x)-\frac{\alpha\epsilon_M}{2} \le L_a(x,\lambda;\epsilon) \le L_a(x_0,\lambda_0;\epsilon) \le f(x_0)+\|\nabla g(x_0)'\nabla_x L(x_0,\lambda_0)+G(x_0)^2\lambda_0\|^2.$$

Therefore we can take

$$\mathcal{C} = \left\{ x \in \bar{\mathcal{P}} : f(x) \le f(x_0) + \frac{\alpha\epsilon_M}{2} + \|\nabla g(x_0)'\nabla_x L(x_0,\lambda_0) + G(x_0)^2\lambda_0\|^2 \right\},$$

which is a compact set by Assumption A1(b). □

THEOREM 3.4. *Suppose that Assumption* A2 *holds; then for every $\epsilon > 0$ the level set $\Omega(x_0,\lambda_0;\epsilon)$ is compact.*

Proof. First we show that $\Omega(x_0,\lambda_0;\epsilon)$ is bounded. We prove this assertion by contradiction; therefore we assume that there exists a sequence $\{(x_k,\lambda_k)\}$ such that

$$L_a(x_k,\lambda_k;\epsilon) \le L_a(x_0,\lambda_0;\epsilon), \tag{3.9}$$

with $x_k \in \mathcal{C}$, where $\mathcal{C}$ is the compact set introduced in Theorem 3.3, and with $\|\lambda_k\| \to \infty$. Since $x_k \in \mathcal{C}$ there exists a subsequence, that we relabel $\{(x_k,\lambda_k)\}$, such that

$$x_k \to \tilde{x}, \qquad \frac{\lambda_k}{\|\lambda_k\|} \to \tilde{\lambda}. \tag{3.10}$$

We note that, by (2.2), we have

$$\lim_{k\to\infty} p(x_k,\lambda_k) = 0, \tag{3.11}$$

$$\lim_{k\to\infty} \|\lambda_k\|\, p(x_k,\lambda_k) = 0, \tag{3.12}$$

$$\lim_{k\to\infty} \|\lambda_k\|^2 p(x_k,\lambda_k) = a(\tilde{x}). \tag{3.13}$$

Now (2.6), (3.9), (3.11), and (3.12) yield

$$\begin{aligned}\lim_{k\to\infty} \frac{1}{\|\lambda_k\|^2} L_a(x_k,\lambda_k;\epsilon) &= \lim_{k\to\infty} \frac{1}{2\epsilon\|\lambda_k\|^2 p(x_k,\lambda_k)} \|\max\{g(x_k), -\epsilon p(x_k,\lambda_k)\lambda_k\}\|^2 \\ &\quad + \lim_{k\to\infty} \left\| M(x_k)\frac{\lambda_k}{\|\lambda_k\|} \right\|^2 \\ &\le \lim_{k\to\infty} \frac{1}{\|\lambda_k\|^2} L_a(x_0,\lambda_0;\epsilon) = 0,\end{aligned}$$

which, taking into account (3.13), yields

$$\lim_{k\to\infty} \max\{g(x_k), -\epsilon p(x_k,\lambda_k)\lambda_k\} = \max\{g(\tilde{x}), 0\} = 0, \tag{3.14}$$

$$\lim_{k\to\infty} \left\| M(x_k)\frac{\lambda_k}{\|\lambda_k\|} \right\|^2 = \left\| M(\tilde{x})\tilde{\lambda} \right\|^2 = 0. \tag{3.15}$$

From (3.14) we get $\tilde{x} \in \mathcal{F}$ and from (3.15) we get $M(\tilde{x})\tilde{\lambda} = 0$, with $\|\tilde{\lambda}\| = 1$. Hence the matrix $M(\tilde{x})$ should be singular, but, under Assumption A2, this is a contradiction. Therefore we can conclude that for every $\epsilon > 0$, $\Omega(x_0,\lambda_0;\epsilon)$ is bounded.

Then we prove that $\Omega(x_0,\lambda_0;\epsilon)$ is closed. To this aim we show that every limit point $(\tilde{x},\tilde{\lambda})$ of every sequence $\{(x_k,\lambda_k)\} \in \Omega(x_0,\lambda_0;\epsilon)$ belongs to $\Omega(x_0,\lambda_0;\epsilon)$. Suppose by contradiction that $(\tilde{x},\tilde{\lambda}) \notin \Omega(x_0,\lambda_0;\epsilon)$; then by the definition of Ω and by

the continuity of L_a, we have that $\tilde{x} \in \partial\mathcal{P}$ and hence $a(\tilde{x}) = 0$. Then, by recalling that $(x_k, \lambda_k) \in \Omega(x_0, \lambda_0; \epsilon)$, it holds that

$$\lim_{k\to\infty} a(x_k)L_a(x_k, \lambda_k; \epsilon) \leq \lim_{k\to\infty} a(x_k)L_a(x_0, \lambda_0; \epsilon) = 0,$$
$$\lim_{k\to\infty} a(x_k)L_a(x_k, \lambda_k; \epsilon) = \frac{1 + \|\tilde{\lambda}\|^2}{2\epsilon} \|\max\{g(\tilde{x}), -\epsilon p(\tilde{x}, \tilde{\lambda})\tilde{\lambda}\}\|^2 = 0,$$

so that $g(\tilde{x}) \leq 0$; this contradicts the statement $a(\tilde{x}) = 0$. Therefore the level set $\Omega(x_0, \lambda_0; \epsilon)$ is compact. □

Remark 3.5. The fact, shown by Theorem 3.4, that the continuously differentiable function $L_a(x, \lambda; \epsilon)$ has level sets that are compact for every value of the penalty parameter ϵ is quite relevant. It implies, on the one hand, that L_a admits a global minimum point, and hence a stationary point, on $\mathcal{P} \times \mathbb{R}^m$; and on the other hand, that any globally convergent unconstrained minimization algorithm, using first order derivatives of the objective function only, can be employed to compute the stationary points of L_a. □

4. Stationary points of the function L_a. In this section we consider the relationships between KKT pairs of problem (P) and stationary points of $L_a(x, \lambda; \epsilon)$. First we prove that, for any $\epsilon > 0$, every KKT pair of problem (P) is a stationary point of L_a. Then we show that every stationary point $(\bar{x}, \bar{\lambda})$ of L_a such that $\max\{g(\bar{x}), -\epsilon p(\bar{x}, \bar{\lambda})\bar{\lambda}\} = 0$ is a KKT pair of problem (P). Finally we prove that, under Assumptions A2 and A3 and for sufficiently small values of ϵ, every stationary point of L_a is such that $\max\{g(\bar{x}), -\epsilon p(\bar{x}, \bar{\lambda})\bar{\lambda}\} = 0$ and, hence, is a KKT pair of problem (P).

THEOREM 4.1. *Let $(\bar{x}, \bar{\lambda})$ be a KKT pair of problem* (P). *Then, for any $\epsilon > 0$, the pair $(\bar{x}, \bar{\lambda})$ is a stationary point of $L_a(x, \lambda; \epsilon)$.*

Proof. The proof is straightforward using (2.8) and (2.9). □

PROPOSITION 4.2. *Let $(\bar{x}, \bar{\lambda}) \in \mathcal{P} \times \mathbb{R}^m$ be a stationary point for $L_a(x, \lambda; \epsilon)$, and assume that*

$$\max\{g(\bar{x}), -\epsilon p(\bar{x}, \bar{\lambda})\bar{\lambda}\} = 0. \tag{4.1}$$

Then, $(\bar{x}, \bar{\lambda})$ is a KKT pair of problem (P).

Proof. The proof can easily be derived as in [9]; we include it for completeness.

By using (2.9) and the assumption that $(\bar{x}, \bar{\lambda})$ is a stationary point of L_a which satisfies (4.1), we obtain

$$M(\bar{x}) \left[\nabla g(\bar{x})' \nabla_x L(\bar{x}, \bar{\lambda}) + G(\bar{x})^2 \bar{\lambda}\right] = 0.$$

Premultiplying by $[\nabla g(\bar{x})' \nabla_x L(\bar{x}, \bar{\lambda}) + G(\bar{x})^2 \bar{\lambda}]'$ and recalling (2.11), we get

$$\left[\nabla g(\bar{x})' \nabla_x L(\bar{x}, \bar{\lambda}) + G(\bar{x})^2 \bar{\lambda}\right]' \left[\nabla g(\bar{x})' \quad G(\bar{x})\right] \times \begin{bmatrix} \nabla g(\bar{x}) \\ G(\bar{x}) \end{bmatrix} \left[\nabla g(\bar{x})' \nabla_x L(\bar{x}, \bar{\lambda}) + G(\bar{x})^2 \bar{\lambda}\right] = 0,$$

from which

$$\begin{bmatrix} \nabla g(\bar{x}) \\ G(\bar{x}) \end{bmatrix} \left[\nabla g(\bar{x})' \nabla_x L(\bar{x}, \bar{\lambda}) + G(\bar{x})^2 \bar{\lambda}\right] = 0.$$

Premultiplying by $\left[\begin{smallmatrix}\nabla_x L(\bar{x},\bar{\lambda})\\ G(\bar{x})\bar{\lambda}\end{smallmatrix}\right]'$ we get

$$\left[\nabla g(\bar{x})'\nabla_x L(\bar{x},\bar{\lambda}) + G(\bar{x})^2\bar{\lambda}\right] = 0. \tag{4.2}$$

Taking into account that $(\bar{x},\bar{\lambda})$ is a stationary point of L_a satisfying (4.1), and using (2.8) and (4.2), we obtain

$$\nabla_x L(\bar{x},\bar{\lambda}) = 0. \tag{4.3}$$

Now the proof of the proposition is complete by recalling that (4.1) implies

$$g(\bar{x}) \le 0, \qquad \bar{\lambda} \ge 0, \qquad G(\bar{x})\bar{\lambda} = 0. \qquad \square \tag{4.4}$$

The next proposition provides a technical result to be employed in the proof that stationary points of L_a are also KKT pairs of problem (P). Due to the technical nature of the proposition, its proof is given in the appendix.

PROPOSITION 4.3. *For every $\hat{x} \in \mathcal{F}$ such that the gradients $\nabla g_i(\hat{x})$, $i \in I_0(\hat{x})$, are linearly independent, there exist numbers $\epsilon(\hat{x}) > 0$, $\sigma(\hat{x}) > 0$, and $\rho(\hat{x}) > 0$ such that, for all $\epsilon \in (0, \epsilon(\hat{x})]$, for all $(x,\lambda) \in \Omega(x_0,\lambda_0;\epsilon)$ satisfying $\|x - \hat{x}\| \le \sigma(\hat{x})$ and $\|\nabla_\lambda L_a(x,\lambda;\epsilon)\| \le \|\max\{g(x), -\epsilon p(x,\lambda)\lambda\}\|$, the following inequality holds:*

$$\epsilon\|\nabla_x L_a(x,\lambda;\epsilon)\| \ge \rho(\hat{x})\|\max\{g(x), -\epsilon p(x,\lambda)\lambda\}\|. \tag{4.5}$$

By using Proposition 4.3 we can prove the following result, which will be exploited also in order to define an updating rule for the penalty parameter in the algorithmic model described in section 7.

PROPOSITION 4.4. *Suppose that Assumptions* A2 *and* A3 *hold. Then there exists an $\bar{\epsilon} > 0$ such that for all $\epsilon \in (0, \bar{\epsilon}]$ and all $(x,\lambda) \in \Omega(x_0,\lambda_0;\epsilon)$ we have*

$$\|\nabla L_a(x,\lambda;\epsilon)\| \ge \|\max\{g(x), -\epsilon p(x,\lambda)\lambda\}\|. \tag{4.6}$$

Proof. The proof is by contradiction. Suppose that the result is false; then, recalling Theorem 3.3, there exist sequences $\{\epsilon_k\}$ and $\{(x_k,\lambda_k)\}$ such that

$$\epsilon_k \to 0, \tag{4.7}$$

$$(x_k,\lambda_k) \in \Omega(x_0,\lambda_0;\epsilon_k), \tag{4.8}$$

$$x_k \to \tilde{x} \in \mathcal{C}, \tag{4.9}$$

$$\|\nabla L_a(x_k,\lambda_k;\epsilon_k)\| < \|\max\{g(x_k), -\epsilon_k p(x_k,\lambda_k)\lambda_k\}\|. \tag{4.10}$$

From (4.10) we have

$$\|\nabla_\lambda L_a(x_k,\lambda_k;\epsilon_k)\| < \|\max\{g(x_k), -\epsilon_k p(x_k,\lambda_k)\lambda_k\}\|, \tag{4.11}$$

$$\epsilon_k p(x_k,\lambda_k)\|\nabla_x L_a(x_k,\lambda_k;\epsilon_k)\| < \epsilon_k p(x_k,\lambda_k)\|\max\{g(x_k), -\epsilon_k p(x_k,\lambda_k)\lambda_k\}\|. \tag{4.12}$$

By the expression of $p(x_k,\lambda_k)$, the sequences $p(x_k,\lambda_k)$ and $\max\{g(x_k), -\epsilon_k p(x_k,\lambda_k)\lambda_k\}$ are bounded; then, by (4.7), we have

$$\lim_{k\to\infty} \epsilon_k p(x_k,\lambda_k)\|\max\{g(x_k), -\epsilon_k p(x_k,\lambda_k)\lambda_k\}\| = 0,$$

which, by (4.12), implies that

$$\lim_{k\to\infty} \epsilon_k p(x_k,\lambda_k)\|\nabla_x L_a(x_k,\lambda_k;\epsilon_k)\| = 0. \tag{4.13}$$

Again by the expression of $p(x,\lambda)$, the sequences $\{p(x_k,\lambda_k)\nabla g(x_k)\lambda_k\}$ and

$$\left\{p(x_k,\lambda_k)Q(x_k,\lambda_k)\left[\nabla g(x_k)'\nabla L(x_k,\lambda_k)+G(x_k)^2\lambda_k\right]\right\}$$

are also bounded. Then, taking the limit of $\{\epsilon_k p(x_k,\lambda_k)\nabla_x L_a(x_k,\lambda_k;\epsilon_k)\}$ and recalling (2.8), we obtain

$$\begin{aligned}0&=\lim_{k\to\infty}\epsilon_k p(x_k,\lambda_k)\nabla_x L_a(x_k,\lambda_k;\epsilon_k)\\&=\sum_{i=1}^{m}\left[1+\frac{s}{2}\frac{\|\max\{g(\tilde x),0\}\|^2\max\{g_i(\tilde x),0\}^{(s-2)}}{a(\tilde x)}\right]\max\{g_i(\tilde x),0\}\nabla g_i(\tilde x),\end{aligned}$$

which, if Assumption A3(a) holds, yields $\tilde x\in\mathcal{F}$.

On the other hand, if Assumption A3(b) holds, namely, if we assume that $x_0\in\mathcal{F}$, by Proposition 3.1(b) and (c), we have, for any k,

$$\begin{aligned}&f(x_k)-\frac{\epsilon_k\alpha}{2}+\frac{1}{2\epsilon_k a(x_k)}\|\max\{g(x_k),-\epsilon_k p(x_k,\lambda_k)\lambda_k\}\|^2\\&\quad\le L_a(x_k,\lambda_k;\epsilon_k)\le L_a(x_0,\lambda_0;\epsilon_k)\le f(x_0)+\|\nabla g(x_0)'\nabla_x L(x_0,\lambda_0)+G(x_0)^2\lambda_0\|^2.\end{aligned}$$

By taking limits and by the continuity assumptions

$$\begin{aligned}&f(\tilde x)+\limsup_{k\to\infty}\frac{1}{2\epsilon_k a(x_k)}\|\max\{g(x_k),-\epsilon_k p(x_k,\lambda_k)\lambda_k\}\|^2\\&\qquad\le f(x_0)+\|\nabla g(x_0)'\nabla_x L(x_0,\lambda_0)+G(x_0)^2\lambda_0\|^2,\end{aligned}$$

which implies $\max\{g(\tilde x),0\}=0$, so that again $\tilde x\in\mathcal{F}$.

In conclusion, if Assumption A3 holds, the point $\tilde x$ to which the sequence converges is feasible. This fact with (4.7), (4.11), and Proposition 4.3 imply that, for sufficiently large k, we have a contradiction with (4.10). □

By the proof of the preceding proposition we can extract the following result, which will be used in what follows.

PROPOSITION 4.5. *Let $\{\epsilon_k\}$ be a sequence of positive numbers converging to zero and let $\{(x_k,\lambda_k)\}$ be a sequence of points such that $(x_k,\lambda_k)\in\Omega(x_0,\lambda_0;\epsilon_k)$. Assume that Assumption* A3(b) *holds; then every limit point $\tilde x$ of the sequence $\{x_k\}$ is feasible for problem* (P).

By combining Propositions 4.2 and 4.4, we can now establish the following result, which, together with Theorem 4.1, completes the analysis of the correspondences between stationary points of $L_a(x,\lambda;\epsilon)$ and KKT pairs of problem (P).

THEOREM 4.6. *Suppose that Assumptions* A2 *and* A3 *hold. Then there exists an $\bar\epsilon>0$ such that for all $\epsilon\in(0,\bar\epsilon]$, if $(\bar x,\bar\lambda)\in\Omega(x_0,\lambda_0;\epsilon)$ is a stationary point of $L_a(x,\lambda;\epsilon)$, the pair $(\bar x,\bar\lambda)$ is a KKT pair for problem* (P).

In conclusion, we have shown that, under Assumptions A1, A2, and A3, for sufficiently small values of ϵ, there exists a one-to-one correspondence between KKT pairs of the constrained problem (P) and the unconstrained stationary points of L_a in the level set $\Omega(x_0,\lambda_0;\epsilon)$; however, since the pair (x_0,λ_0) is arbitrary in $\mathcal{P}\times\mathbb{R}^m$, the correspondence holds on the whole set $\mathcal{P}\times\mathbb{R}^m$.

5. Optimality results. In this section, we complete the analysis of the exactness properties of $L_a(x,\lambda;\epsilon)$ by establishing the relationships between local or global solutions of problem (P) and local or global unconstrained minimum points of the augmented Lagrangian function.

To this aim, we first recall (see, e.g., [17, p. 46]) that given a set $\mathcal{M}$, a nonempty subset $\mathcal{M}^* \subset \mathcal{M}$ is called an isolated set of $\mathcal{M}$ if there exists a closed set $\mathcal{E}$ such that $\overset{\circ}{\mathcal{E}} \supset \mathcal{M}^*$ and such that, if $x \in \mathcal{E} \backslash \mathcal{M}^*$, then $x \notin \mathcal{M}$. If $\mathcal{M}(\bar{f})$ denotes the set of local minimum points of problem (P) corresponding to the local minimum value $\bar{f}$, then an isolated compact set $\mathcal{M}^*(\bar{f})$ of $\mathcal{M}(\bar{f})$ possesses a property pointed out in [12].

PROPOSITION 5.1. *Let $\mathcal{M}^*(\bar{f})$ be an isolated compact set of local minimum points of problem* (P), *corresponding to the local minimum value $\bar{f}$. Then there exists a compact set $\mathcal{E} \subset \mathcal{P}$ such that $\mathcal{M}^*(\bar{f}) \subset \overset{\circ}{\mathcal{E}}$ and, for any point $x \in \mathcal{F} \cap \mathcal{E}$, if $x \notin \mathcal{M}^*(\bar{f})$, then $f(x) > \bar{f}$.*

Now we can prove that isolated compact sets of local minimizers of problem (P) correspond to unconstrained local minimizers of L_a.

THEOREM 5.2. *Suppose that Assumption* A2 *holds. Let $\mathcal{M}^*(\bar{f})$ be an isolated compact set of local minimum points of problem* (P), *corresponding to the local minimum value $\bar{f}$; then there exists an $\bar{\epsilon} > 0$ such that for all $\epsilon \in (0, \bar{\epsilon}]$, if $\bar{x} \in \mathcal{M}^*(\bar{f})$ and $\bar{\lambda}$ is the associated KKT multiplier, then the pair $(\bar{x}, \bar{\lambda})$ is a local unconstrained minimum point of $L_a(x, \lambda; \epsilon)$.*

Proof. By Proposition 5.1 there exists a compact set $\mathcal{E} \subset \mathcal{P}$ such that $\mathcal{M}^*(\bar{f}) \subset \overset{\circ}{\mathcal{E}}$ and, for any point $x \in \mathcal{F} \cap \mathcal{E}$, if $x \notin \mathcal{M}^*(\bar{f})$, then $f(x) > \bar{f}$. By Theorem 4.1 and point (a) of Proposition 3.1, the pair $(\bar{x}, \bar{\lambda})$ is a stationary point of $L_a(x, \lambda; \epsilon)$, with $L_a(\bar{x}, \bar{\lambda}; \epsilon) = \bar{f}$.

Now, assume that the proposition is false. Then, for any integer k, there must exist an $\epsilon_k \leq 1/k$ and a pair $(\bar{x}_k, \bar{\lambda}_k) \in \mathcal{M}^*(\bar{f}) \times \mathbb{R}^m$ which is not a local unconstrained minimum point for $L_a(x, \lambda; \epsilon_k)$. On the other hand, since $\mathcal{E} \subset \mathcal{P}$, Theorem 3.4 implies that $L_a(x, \lambda; \epsilon_k)$ has a global minimum point (x_k, λ_k) on $\mathcal{E} \times \mathbb{R}^m$; this point satisfies

$$L_a(x_k, \lambda_k; \epsilon_k) < L_a(\bar{x}_k, \bar{\lambda}_k; \epsilon_k) = \bar{f}. \tag{5.1}$$

Since $\mathcal{E}$ is compact, there exists a subsequence, which we relabel $\{(x_k, \lambda_k)\}$, such that

$$\lim_{k \to \infty} x_k = \tilde{x} \in \mathcal{E}.$$

By taking limits, from (3.2) and (5.1), we have

$$f(\tilde{x}) + \limsup_{k \to \infty} \frac{1}{2\epsilon_k a(x_k)} \| \max\{g(x_k), -\epsilon_k p(x_k, \lambda_k)\lambda_k\} \|^2 \leq \limsup_{k \to \infty} L_a(x_k, \lambda_k; \epsilon_k) \leq \bar{f}. \tag{5.2}$$

Taking into account that the term $p(x, \lambda)\lambda$ is bounded in $\mathcal{E} \times \mathbb{R}^m$, (5.2) implies that $\tilde{x} \in \mathcal{F}$ and $f(\tilde{x}) \leq \bar{f}$, so that $\tilde{x} \in \mathcal{M}^*(\bar{f})$.

Since $\tilde{x} \in \overset{\circ}{\mathcal{E}}$ and $x_k \to \tilde{x}$, we must have, for k large enough,

$$\nabla L_a(x_k, \lambda_k; \epsilon_k) = 0. \tag{5.3}$$

Now, if we introduce the level sets $\Omega(x_0, \lambda_0; \epsilon_k)$, where $(x_0, \lambda_0) = (\bar{x}, \bar{\lambda}) \in \mathcal{M}^*(\bar{f}) \times \mathbb{R}^m$, we have that Assumption A3(b) is satisfied and, by (5.1), that $(x_k, \lambda_k) \in \Omega(x_0, \lambda_0; \epsilon_k)$. Therefore Theorem 4.6 and (5.3) imply that, for sufficiently large values of k, (x_k, λ_k) is a KKT pair of problem (P). Then, we have $x_k \in \mathcal{F} \cap \mathcal{E}$ and, by point (a) of Proposition 3.1, $L_a(x_k, \lambda_k; \epsilon_k) = f(x_k)$. Therefore, by (5.1), we obtain $f(x_k) < \bar{f}$, which contradicts the assumption $x_k \in \mathcal{F} \cap \mathcal{E}$. □

By reasoning as in [9] and [14], we can also prove the following converse result.

THEOREM 5.3. *Suppose that Assumptions* A2 *and* A3 *hold. Then, there exists an* $\bar{\epsilon} > 0$ *such that, for all* $\epsilon \in (0, \bar{\epsilon}]$, *if* $(\bar{x}, \bar{\lambda}) \in \Omega(x_0, \lambda_0; \epsilon)$ *is a local unconstrained minimum point of* $L_a(x, \lambda; \epsilon)$, $\bar{x}$ *is a local minimum point of problem* (P) *and* $\bar{\lambda}$ *is the corresponding KKT multiplier.*

Proof. If $(\bar{x}, \bar{\lambda}) \in \Omega(x_0, \lambda_0; \epsilon)$ is a local minimum point of $L_a(x, \lambda; \epsilon)$, then Theorem 4.6 ensures that an $\bar{\epsilon} > 0$ exists such that, for all $\epsilon \in (0, \bar{\epsilon}]$, the pair $(\bar{x}, \bar{\lambda})$ is a KKT pair for problem (P). By point (a) of Proposition 3.1 we have also $f(\bar{x}) = L_a(\bar{x}, \bar{\lambda}; \epsilon)$.

Since $(\bar{x}, \bar{\lambda}) \in \Omega(x_0, \lambda_0; \epsilon)$ is a local unconstrained minimum point of $L_a(x, \lambda; \epsilon)$, there exist neighborhoods $\mathcal{B}_{\bar{x}}$ and $\mathcal{B}_{\bar{\lambda}}$ of $\bar{x}, \bar{\lambda}$ such that

$$f(\bar{x}) = L_a(\bar{x}, \bar{\lambda}; \epsilon) \leq L_a(x, \lambda; \epsilon)$$

for all $x \in \mathcal{B}_{\bar{x}}$ and for all $\lambda \in \mathcal{B}_{\bar{\lambda}}$. By point (b) of Proposition 3.1 we have

$$f(\bar{x}) \leq L_a(x, \lambda; \epsilon) \leq f(x) + \|\nabla g(x)' \nabla_x L(x, \lambda) + G(x)^2 \lambda\|^2 \tag{5.4}$$

for all $x \in \mathcal{B}_{\bar{x}} \cap \mathcal{F}$ and for all $\lambda \in \mathcal{B}_{\bar{\lambda}}$. Recalling that by Assumption A2 the matrix $M(x)$ is nonsingular, we have that, for every point x, the term $\|\nabla g(x)' \nabla_x L(x, \lambda) + G(x)^2 \lambda\|^2 = \|\nabla g(x)' \nabla f(x) + M(x)\lambda\|^2$ is a strictly convex function in λ, whose unique minimum point is given by

$$\lambda(x) = -M(x)^{-1} \nabla g(x)' \nabla f(x).$$

Since $\|\nabla g(\bar{x})' \nabla_x L(\bar{x}, \bar{\lambda}) + G(\bar{x})^2 \bar{\lambda}\|^2 = 0$, we have that $\lambda(\bar{x}) = \bar{\lambda}$. Then there exists a neighborhood $\mathcal{B}'_{\bar{x}} \subseteq \mathcal{B}_{\bar{x}}$ such that $\lambda(x) \in \mathcal{B}_{\bar{\lambda}}$ for all $x \in \mathcal{B}'_{\bar{x}}$. Therefore, (5.4) implies

$$f(\bar{x}) \leq L_a(x, \lambda(x); \epsilon) \leq f(x) + \|\nabla g(x)' \nabla f(x) + M(x)\lambda(x)\|^2 = f(x)$$

for all $x \in \mathcal{B}'_{\bar{x}} \cap \mathcal{F}$, and this shows that $\bar{x}$ is a local minimum point also for problem (P). ∎

Again reasoning as in [14], we can state the one-to-one correspondence between global solutions of problem (P) and global minimizers of L_a.

THEOREM 5.4. *Suppose that the feasible set* $\mathcal{F}$ *is not empty and that Assumption* A2 *holds. Then, there exists an* $\bar{\epsilon} > 0$ *such that, for all* $\epsilon \in (0, \bar{\epsilon}]$, *if* $\bar{x}$ *is a global minimum point of problem* (P) *and* $\bar{\lambda}$ *is the corresponding KKT multiplier, the pair* $(\bar{x}, \bar{\lambda})$ *is a global minimum point of* $L_a(x, \lambda; \epsilon)$ *on* $\mathcal{P} \times \mathbb{R}^m$, *and conversely.*

Proof. If $\bar{x}$ is a global minimum point for problem (P) and $\bar{\lambda}$ is the corresponding KKT multiplier, then $(\bar{x}, \bar{\lambda})$ is a KKT pair, so that, by point (a) of Proposition 3.1, we have $f(\bar{x}) = L_a(\bar{x}, \bar{\lambda}; \epsilon)$.

On the other hand, if $(\bar{x}, \bar{\lambda})$ is global minimum point of $L_a(x, \lambda; \epsilon)$, it is a stationary point of L_a contained in the level set $\Omega(x_0, \lambda_0; \epsilon)$ with $x_0 \in \mathcal{F}$. Recalling Theorem 4.6, we have that there exists $\bar{\epsilon}$ such that for all $\epsilon \in (0, \bar{\epsilon}]$ every stationary point of L_a in $\Omega(x_0, \lambda_0; \epsilon)$ is a KKT pair of problem (P). Therefore, also in this case, by point (a) of Proposition 3.1 we have $f(\bar{x}) = L_a(\bar{x}, \bar{\lambda}; \epsilon)$. We can conclude that the functions f and L_a take the same value at every point that is either a global minimum point for problem (P) or a global minimum point of L_a, and this proves the proposition. ∎

6. Second order analysis. In this section we assume that f and g_i, $i = 1, \ldots, m$, are three times continuously differentiable functions and that $s > 2$. Under these assumptions we perform an analysis of the second order properties of the Lagrangian function L_a. This analysis allows us to prove an additional exactness result, and provides the bases for the definition of algorithms which combine the

global convergence with a superlinear convergence rate, without requiring the strict complementarity assumption (see section 8).

Since L_a is an SC^1 function in $\mathcal{P} \times \mathbb{R}^m$, its generalized Hessian $\partial^2 L_a(x, \lambda; \epsilon)$, in Clarke's sense, can be defined [7]. We recall that the generalized Hessian $\partial^2 L_a(x, \lambda; \epsilon)$ is the set of matrices given by

$$\partial^2 L_a(x, \lambda; \epsilon) = \operatorname{co}\{\partial_B^2 L_a(x, \lambda; \epsilon)\},$$

where

$$\begin{aligned}\partial_B^2 L_a(x, \lambda; \epsilon) = \big\{ & H \in \mathbb{R}^{(n+m)\times(n+m)} : \exists\{(x^k, \lambda^k)\} \to (x, \lambda) \text{ with} \\ & \nabla^2 L_a \text{ differentiable at } (x^k, \lambda^k) \text{ and } \{\nabla^2 L_a(x^k, \lambda^k; \epsilon)\} \to H\big\}.\end{aligned}$$

The generalized Hessian $\partial^2 L_a$ is a nonempty, convex, compact set of symmetric matrices; furthermore, the point-to-set map $(x, \lambda) \mapsto \partial^2 L_a(x, \lambda; \epsilon)$ is bounded on bounded sets [21].

For the Lagrangian function L_a it is possible to describe the structure of the generalized Hessian $\partial^2 L_a$ in a neighborhood of a KKT pair of problem (P). To this aim we consider a partition of the index set $\{1, \dots, m\}$ into the subsets $A \subseteq \{1, \dots, m\}$, $N = \{1, \dots, m\} \setminus A$, and we partition the vectors g and λ according to these index sets: $g = (g_A' \ g_N')'$ and $\lambda = (\lambda_A' \ \lambda_N')'$. Then we introduce the $(n + m) \times (n + m)$ symmetric matrix $H(x, \lambda; \epsilon, A)$ given blockwise by

$$\begin{aligned} H_{xx}(x, \lambda; \epsilon, A) &= \nabla_x^2 L(x, \lambda) \\ &\quad + \frac{1}{\epsilon p(x, \lambda)} \nabla g_A(x) \nabla g_A(x)' + 2\nabla_x^2 L(x, \lambda) \nabla g(x) \nabla g(x)' \nabla_x^2 L(x, \lambda), \end{aligned} \tag{6.1}$$

$$\begin{aligned} H_{x\lambda}(x, \lambda; \epsilon, A) &= \begin{bmatrix} \nabla g_A(x) & 0 \end{bmatrix} \\ &\quad + 2\nabla_x^2 L(x, \lambda) \nabla g(x) \left(\nabla g(x)' \nabla g(x) + \begin{bmatrix} 0 & 0 \\ 0 & G_N(x)^2 \end{bmatrix} \right), \end{aligned} \tag{6.2}$$

$$\begin{aligned} H_{\lambda\lambda}(x, \lambda; \epsilon, A) &= -\epsilon p(x, \lambda) \begin{bmatrix} 0 & 0 \\ 0 & I_N \end{bmatrix} \\ &\quad + 2\left(\nabla g(x)' \nabla g(x) + \begin{bmatrix} 0 & 0 \\ 0 & G_N(x)^2 \end{bmatrix} \right) \left(\nabla g(x)' \nabla g(x) + \begin{bmatrix} 0 & 0 \\ 0 & G_N(x)^2 \end{bmatrix} \right), \end{aligned} \tag{6.3}$$

where I_N is the identity matrix of dimension $|N|$ and 0 is a zero matrix of proper dimensions.

The following proposition, which is proved in the appendix, states that in a neighborhood of a KKT pair of problem (P), the set $\partial_B^2 L_a(x, \lambda; \epsilon)$ can be described almost explicitly.

PROPOSITION 6.1. *For every KKT pair $(\bar{x}, \bar{\lambda})$ of problem* (P) *and every given ϵ there exists a neighborhood $\mathcal{B}$ of $(\bar{x}, \bar{\lambda})$ such that, for all (x, λ) in $\mathcal{B}$, we have*

$$\partial_B^2 L_a(x, \lambda; \epsilon) = \{H(x, \lambda; \epsilon, A) + K(x, \lambda; \epsilon, A) : \ A \in \mathcal{A}\},$$

where

- $\mathcal{A} = \{A : A = I_+(\bar{x}, \bar{\lambda}) \cup J \quad \forall J \subseteq I_0(\bar{x}) \setminus I_+(\bar{x}, \bar{\lambda})\}$;
- $H(x, \lambda; \epsilon, A)$ *is a matrix given blockwise by* (6.1)–(6.3);
- $K(x, \lambda; \epsilon, A)$ *is a matrix such that* $\|K(x, \lambda; \epsilon, A)\| \leq \rho(x, \lambda)$, *with* $\rho(x, \lambda)$ *a nonnegative continuous function such that* $\rho(\bar{x}, \bar{\lambda}) = 0$.

At a KKT pair where strict complementarity holds, we have $I_+(\bar{x},\bar{\lambda}) = I_0(\bar{x})$. In this case $\partial^2 L_a(\bar{x},\bar{\lambda};\epsilon)$ reduces to a singleton, and in a neighborhood of the KKT pair the generalized Hessian can be further characterized.

PROPOSITION 6.2. *For every KKT pair $(\bar{x},\bar{\lambda})$ of problem* (P) *where strict complementarity holds, and for every given ϵ, there exists a neighborhood $\mathcal{B}$ of $(\bar{x},\bar{\lambda})$ such that for all (x,λ) in $\mathcal{B}$, L_a is twice continuously differentiable, with Hessian matrix given by*

$$\nabla^2 L_a(x,\lambda;\epsilon) = H(x,\lambda;\epsilon,A) + K(x,\lambda;\epsilon,A),$$

where $A = I_0(\bar{x})$, and H and K are matrices as in Proposition 6.1.

The next theorem, which can be considered the main result of this section, proves that, for sufficiently small values of ϵ, KKT pairs of problem (P) satisfying the strong second order sufficient condition are strict local minimizers of L_a which satisfy also the second order sufficient optimality condition for SC^1 functions (see [23]).

THEOREM 6.3. *Let $(\bar{x},\bar{\lambda})$ be a KKT pair of problem* (P) *which satisfies the strong second order sufficient condition and assume that at $\bar{x}$ the gradients of the active constraints are linearly independent. Then there exists an $\bar{\epsilon}$ such that, for all $\epsilon \in (0,\bar{\epsilon}]$, $(\bar{x},\bar{\lambda})$ is an isolated local minimum point for $L_a(x,\lambda;\epsilon)$, and all matrices in $\partial^2 L_a(\bar{x},\bar{\lambda};\epsilon)$ are positive definite.*

Proof. By Theorem 4.1, the pair $(\bar{x},\bar{\lambda})$ is a stationary point for $L_a(x,\lambda;\epsilon)$. By Proposition 6.1 and by the Carathéodory theorem, every matrix $\bar{H}(\bar{x},\bar{\lambda};\epsilon)$ in $\partial^2 L_a(\bar{x},\bar{\lambda};\epsilon)$ can be written in the form

$$\bar{H}(\bar{x},\bar{\lambda};\epsilon) = \sum_{i=1}^{t} \beta_i H(\bar{x},\bar{\lambda};\epsilon,A_i),$$

where $t \le (n+m)^2+1$, $\beta_i \ge 0$, $\sum_{i=1}^{t}\beta_i = 1$, and $A_i \in \mathcal{A}$. Now consider the following quadratic form in $(v,w) \in \mathbb{R}^n \times \mathbb{R}^m$:

$$(v',w')\bar{H}(\bar{x},\bar{\lambda};\epsilon)\begin{pmatrix} v \\ w \end{pmatrix} = s(v,w) + \frac{1}{\epsilon}q(v,w) + \epsilon r(v,w), \tag{6.4}$$

where

$$\begin{aligned} s(v,w) &= v'\nabla_x^2 L(\bar{x},\bar{\lambda})v + 2\|\nabla g(\bar{x})'\nabla_x^2 L(\bar{x},\bar{\lambda})v + M(\bar{x})w\|^2 \\ &\quad + 2\sum_{i=1}^{t}\beta_i v'\nabla g_{A_i}(\bar{x})w_{A_i}, \\ q(v,w) &= \frac{1}{p(\bar{x},\bar{\lambda})}\sum_{i=1}^{t}\beta_i\|\nabla g_{A_i}(\bar{x})'v\|^2, \\ r(v,w) &= -p(\bar{x},\bar{\lambda})\sum_{i=1}^{t}\beta_i\|w_{N_i}\|^2. \end{aligned}$$

First we note that $q(v,w) \ge 0$. Then we observe that $q(v,w) = 0$ implies that

$$\nabla g'_{I_+(\bar{x},\bar{\lambda})}v = 0.$$

Therefore, since the pair $(\bar{x},\bar{\lambda})$ satisfies the strong second order sufficient condition

for problem (P), $s(v,w) \leq 0$ and $q(v,w) = 0$ imply that

$$v = 0, \qquad M(\bar{x})w = 0.$$

Since, by the linear independence of the gradients of the active constraints, the matrix $M(\bar{x})$ is nonsingular, this last equality implies that $w = 0$. In conclusion, for every $(v,w) \in \mathbb{R}^n \times \mathbb{R}^m$,

(i) $q(v,w) \geq 0$;
(ii) $q(v,w) = 0$ and $s(v,w) \leq 0$ imply $v = 0$ and $w = 0$.

Recalling known results on the sum of quadratic forms (see, for example, [9]) it is possible to assert that there exists a value $\bar{\epsilon}$ such that $s(v,w) + (1/\epsilon)q(v,w) + \epsilon r(v,w)$ is positive definite for all $\epsilon \in (0, \bar{\epsilon}]$. This implies, by (6.4), that every matrix in $\partial^2 L_a(\bar{x}, \bar{\lambda}; \epsilon)$ is positive definite for every $\epsilon \in (0, \bar{\epsilon}]$. Then the pair $(\bar{x}, \bar{\lambda})$ also satisfies the second order sufficient condition to be an isolated minimum point for L_a; see [23]. □

7. The algorithmic model. In this section we describe an algorithmic model for the solution of problem (P), based on the unconstrained minimization of the function $L_a(x, \lambda; \epsilon)$. The main result in this section is that, under Assumptions A1, A2, and A3 employed to establish the exactness properties of L_a, every algorithm described by the model is globally convergent towards KKT pairs of problem (P), without requiring that the penalty parameter ϵ goes to zero. As is well known, the fact that the penalty parameter is bounded away from zero limits ill-conditioning in the unconstrained minimization of merit functions.

Nevertheless the algorithm is able to extract some information about the original problem even when Assumptions A2 and/or A3 do not hold. Therefore, drawing our inspiration from [5, 6], we will analyze in detail the behavior of the algorithm when only Assumption A1 holds. In this analysis we use the notion of generalized critical point (see, e.g., [22]).

A point $\bar{x} \in \mathcal{F}$ is a *generalized critical point* for problem (P) if there exist multipliers $\bar{\eta} \in \mathbb{R}$ and $\bar{\lambda} \in \mathbb{R}^m$, not both zero, such that

$$\nabla f(\bar{x})\bar{\eta} + \nabla g(\bar{x})\bar{\lambda} = 0, \qquad G(\bar{x})\bar{\lambda} = 0.$$

In particular, a point $\bar{x} \in \mathcal{F}$ is a generalized critical point if there exists a $\bar{\lambda} \neq 0$ such that $\nabla g(\bar{x})\bar{\lambda} = 0$, $G(\bar{x})\bar{\lambda} = 0$.

In the algorithm we make use of an iteration map $T : \mathcal{P} \times \mathbb{R}^m \to \mathcal{P} \times \mathbb{R}^m$ that satisfies the following assumption.

Assumption A4. For every fixed value ϵ and every starting point $(x_0, \lambda_0) \in \mathcal{P} \times \mathbb{R}^m$, the sequence $\{(x_k, \lambda_k)\}$ given by $(x_{k+1}, \lambda_{k+1}) = T\big[(x_k, \lambda_k)\big]$ belongs to the level set $\Omega(x_0, \lambda_0; \epsilon)$, and all its limit points are stationary points of $L_a(x, \lambda; \epsilon)$.

These requirements on the map T can be easily satisfied by any globally convergent algorithm for the unconstrained minimization of $L_a(x, \lambda; \epsilon)$. In fact we can always ensure, by simple devices, that the trial points produced along the search direction remain in $\Omega(x_0, \lambda_0; \epsilon)$. In the next section we give some guidelines in order to define an iteration map T able to provide a superlinear convergence rate.

Now, we can describe the algorithm.

Algorithm ALFA (Augmented Lagrangian Function Algorithm).
Data: $(z_0, \mu_0) \in \mathbb{R}^n \times \mathbb{R}^m$ and $\epsilon_0 > 0$.
Choose $\alpha > 0$ such that $z_0 \in \mathcal{P}$, set $j = 0$, $k = 0$, and $(x_0, \lambda_0) = (z_0, \mu_0)$.
While $\|\nabla L_a(x_0, \lambda_0; \epsilon_j)\| + \|\max\{g(x_0), -\epsilon_j p(x_0, \lambda_0)\lambda_0\}\| \neq 0$ do.
  While $\|\nabla L_a(x_k, \lambda_k; \epsilon_j)\| \geq \|\max\{g(x_k), -\epsilon_j p(x_k, \lambda_k)\lambda_k\}\|$ do.
    Compute $(x_{k+1}, \lambda_{k+1}) = T\big[(x_k, \lambda_k)\big]$, set $k = k + 1$.
    If $\|\nabla L_a(x_k, \lambda_k; \epsilon_j)\| + \|\max\{g(x_k), -\epsilon_j p(x_k, \lambda_k)\lambda_k\}\| = 0$ STOP.
  End while
  Choose $\epsilon_{j+1} \in (0, \epsilon_j)$, set $(z_{j+1}, \mu_{j+1}) = (x_k, \lambda_k)$, $j = j + 1$, and $k = 0$.
  If $L_a(z_0, \mu_0; \epsilon_j) \leq L_a(z_j, \mu_j; \epsilon_j)$ set $(x_0, \lambda_0) = (z_0, \mu_0)$; else set $(x_0, \lambda_0) = (z_j, \mu_j)$.
End while

The algorithm performs an outer iteration and an inner iteration. The outer iteration, indexed by j, monitors the decrease of the penalty parameter and provides a proper starting point (x_0, λ_0) for the inner iteration. The inner iteration, indexed by k, performs an unconstrained minimization of L_a starting from (x_0, λ_0). In particular, the outer iteration produces the sequences $\{\epsilon_j\} \subset \mathbb{R}^+$ and $\{(z_j, \mu_j)\} \subseteq \mathcal{P} \times \mathbb{R}^m$; the inner iteration produces, for a fixed ϵ_j, a sequence $\{(x_k, \lambda_k)\}_j \subseteq \Omega(x_0, \lambda_0; \epsilon_j)$.

Theorem 7.1. *Let $\{\epsilon_j\}$, $\{(z_j, \mu_j)\}$, and $\{\{(x_k, \lambda_k)\}_j\}$ be the sequences produced by Algorithm ALFA. Then the following hold:*

(a) *If the sequence $\{\epsilon_j\}$ is finite, with last element $\epsilon_{\bar{j}}$, then*
  (a1) *in the case that the sequence $\{\lambda_k\}_{\bar{j}}$ is bounded, every limit point $(\tilde{x}, \tilde{\lambda})$ of the sequence $\{(x_k, \lambda_k)\}_{\bar{j}}$ is a KKT pair for problem* (P);
  (a2) *in the case that the sequence $\{\lambda_k\}_{\bar{j}}$ is not bounded, the sequence $\{x_k\}_{\bar{j}}$ has a limit point $\tilde{x}$ which is a generalized critical point for problem* (P).

(b) *If the sequence $\{\epsilon_j\}$ is infinite, then*
  (b1) *every limit point $\tilde{z}$ of the sequence $\{z_j\}$ such that $\tilde{z} \notin \mathcal{F}$ violates Assumption* A3(a), *that is,*

$$\sum_{i:g_i(\tilde{z})>0} c_i(\tilde{z}) \nabla g_i(\tilde{z}) = 0, \tag{7.1}$$

  where the positive coefficients $c_i(\tilde{z})$ are given by (2.13);
  (b2) *every limit point $\tilde{z}$ of the sequence $\{z_j\}$ such that $\tilde{z} \in \mathcal{F}$ is a generalized critical point for problem* (P).

Proof. (a) Let $\epsilon_{\bar{j}}$ be the last element of the finite sequence $\{\epsilon_j\}$. The properties of the iteration map T ensure that every point (x_k, λ_k), produced with $\epsilon = \epsilon_{\bar{j}}$, belongs to the level set $\Omega(x_0, \lambda_0; \epsilon_{\bar{j}})$.

Now, if the sequence $\{\lambda_k\}_{\bar{j}}$ is bounded, Theorem 3.3 implies that the sequence $\{(x_k, \lambda_k)\}_{\bar{j}}$ is bounded. Furthermore, by Assumption A4, every limit point $(\tilde{x}, \tilde{\lambda})$ of the sequence $\{(x_k, \lambda_k)\}_{\bar{j}}$ is a stationary point of the function $L_a(x, \lambda; \epsilon_{\bar{j}})$. Due to the condition in the inner while-instruction, we have that $\nabla_x L_a(\tilde{x}, \tilde{\lambda}; \epsilon_{\bar{j}}) = 0$ implies $\max\{g(\tilde{x}), -\epsilon_{\bar{j}} p(\tilde{x}, \tilde{\lambda})\tilde{\lambda}\} = 0$; then, point (a1) follows from Proposition 4.2.

If the sequence $\{\lambda_k\}_{\bar{j}}$ is not bounded, we can repeat the same arguments of the proof of Theorem 3.4 and we can conclude (see (3.14) and (3.15)) that there exists a

limit point $(\tilde{x}, \tilde{\lambda})$ of the sequence $\{(x_k, \lambda_k/\|\lambda_k\|)\}_{\bar{j}}$ such that

$$\max\{g(\tilde{x}), 0\} = 0, \tag{7.2}$$

$$M(\tilde{x})\tilde{\lambda} = 0, \tag{7.3}$$

$$\|\tilde{\lambda}\| = 1. \tag{7.4}$$

Now (7.3) yields

$$\tilde{\lambda}' M(\tilde{x})\tilde{\lambda} = \tilde{\lambda}' \begin{pmatrix} \nabla g(\tilde{x})' & G(\tilde{x}) \end{pmatrix} \begin{pmatrix} \nabla g(\tilde{x}) \\ G(\tilde{x}) \end{pmatrix} \tilde{\lambda} = 0,$$

which, in turn, implies that

$$\nabla g(\tilde{x})\tilde{\lambda} = 0, \tag{7.5}$$

$$G(\tilde{x})\tilde{\lambda} = 0. \tag{7.6}$$

Now the proof of point (a2) follows from (7.2), (7.4), (7.5), and (7.6).

(b) Consider the sequence $\{(z_j, \mu_j)\}$. For $j > 0$, the points (z_j, μ_j) are produced because it happens that

$$\begin{aligned} \|\nabla_x L_a(z_{j+1}, \mu_{j+1}; \epsilon_j)\|^2 + \|\nabla_\lambda L_a(z_{j+1}, \mu_{j+1}; \epsilon_j)\|^2 \\ < \|\max\{g(z_{j+1}, -\epsilon_j p(z_{j+1}, \mu_{j+1})\mu_{j+1})\}\|^2. \end{aligned} \tag{7.7}$$

Now, we observe that, by the if-instruction of the outer iteration, we have

$$(z_{j+1}, \mu_{j+1}) \in \Omega(z_0, \mu_0; \epsilon_j)$$

for all j; then Theorem 3.3 ensures that the sequence $\{z_j\}$ is bounded. Let $\tilde{z}$ be any limit point of $\{z_j\}$; (7.7) implies that

$$\lim_{j\to\infty} \epsilon_j p(z_{j+1}, \mu_{j+1}) \nabla_x L_a(z_{j+1}, \mu_{j+1}; \epsilon_j) = 0.$$

By the expression of the term $p(x, \lambda)$, the sequences

$$\begin{aligned} &\{p(z_{j+1}, \mu_{j+1})\}, \\ &\{p(z_{j+1}, \mu_{j+1}) \nabla g(z_{j+1}) \mu_{j+1}\}, \\ &\{p(z_{j+1}, \mu_{j+1}) Q(z_{j+1}, \mu_{j+1}) \left[\nabla g(z_{j+1})' \nabla_x L(z_{j+1}, \mu_{j+1}) + G(z_{j+1})^2 \mu_{j+1}\right]\} \end{aligned}$$

are bounded. Then, we obtain

$$\begin{aligned} &\lim_{j\to\infty} \epsilon_j p(z_{j+1}, \mu_{j+1}) \nabla_x L_a(z_{j+1}, \mu_{j+1}; \epsilon_j) \\ &\quad = \sum_{i=1}^{m} \left[1 + \frac{s}{2} \frac{\|\max\{g(\tilde{z}), 0\}\|^2}{a(\tilde{z})} \max\{g_i(\tilde{z}), 0\}^{s-2}\right] \max\{g_i(\tilde{z}), 0\} \nabla g_i(\tilde{z}) = 0, \end{aligned}$$

and this proves point (b1) if $\tilde{z} \notin \mathcal{F}$.

If $\tilde{z} \in \mathcal{F}$ and the gradients $\nabla g_i(\tilde{z})$, for $i \in I_0(\tilde{z})$, are linearly dependent, there exists $\tilde{\mu} \neq 0$ such that $\nabla g(\tilde{z})\tilde{\mu} = 0$ and $G(\tilde{z})\tilde{\mu} = 0$. Therefore, $\tilde{z}$ is a generalized critical point for problem (P). On the other hand, if the gradients $\nabla g_i(\tilde{z})$ for $i \in I_0(\tilde{z})$ were linearly independent we would get a contradiction between Proposition 4.3 and the fact that the points of the sequence $\{(z_j, \mu_j)\}$ do not satisfy the test in the inner while-instruction; thus, point (b2) is also proved. □

Remark 7.2. It can be verified that, if case (b1) occurs, $\tilde{z}$ is a stationary point of the function

$$\phi(x) = \frac{1}{a(x)} \| \max\{g(x), 0\} \|^2,$$

where $a(x)$ is given by (2.1). The function ϕ is a weighted measure of the constraint violations. In particular, if case (b1) occurs with $s = 2$ in (2.1), the point $\tilde{z}$ becomes also a stationary point of the distance function

$$\operatorname{dist}[g(x) | \mathbb{R}^m_-] = \inf_y \{ \|g(x) - y\|; y \leq 0 \in \mathbb{R}^m \} = \| \max\{g(x), 0\} \|;$$

therefore, if problem (P) is nonfeasible and $g(x)$ is convex, Algorithm ALFA provides a point that is as close to feasibility as possible (see [25]). □

If Assumption A2 holds in addition to Assumption A1, then only the cases (a1) and (b1) can occur for the sequences generated by Algorithm ALFA. This is easily seen by looking at the proof of Theorem 7.1. In fact, the point $\tilde{x}$ produced in the case (a2) would be feasible by (7.2) and would violate Assumption A2 due to (7.4), (7.5), (7.6); the feasible point $\tilde{z}$ produced in the case (b2) would again violate Assumption A2. On the other hand, if Assumption A3 holds in addition to Assumption A1, then the case (b1) cannot occur. In fact, if Assumption A3(b) holds, then Proposition 4.5 ensures that every accumulation point of $\{x_k\}$ should be a feasible point; if Assumption A3(a) holds, then every point satisfying (7.1) should be feasible.

Therefore, we can conclude with the following main result.

THEOREM 7.2. *Assume that Assumptions* A1, A2, *and* A3 *hold, and let* $\{\epsilon_j\}$, $\{\{(x_k, \lambda_k)\}_j\}$ *be the sequences produced by Algorithm ALFA. Then the sequence* $\{\epsilon_j\}$ *is finite; if* $\epsilon_{\bar{j}}$ *is the final value of the penalty parameter, every limit point of the sequence* $\{(x_k, \lambda_k)\}_{\bar{j}}$ *is a KKT pair for problem* (P).

8. Remarks on the iteration map T. In this section we give some guidelines on the construction of the iteration map T for the unconstrained minimization of L_a.

We confine ourselves to a linesearch approach. In this case the sequence $\{(x^k, \lambda^k)\}$ given by the map T is described by the iteration

$$x^{k+1} = x^k + \theta^k d^k_x, \tag{8.1}$$

$$\lambda^{k+1} = \lambda^k + \theta^k d^k_\lambda, \tag{8.2}$$

where $\theta^k \in \mathbb{R}$ is the steplength and $d^k = (d^k_x, d^k_\lambda) \in \mathbb{R}^n \times \mathbb{R}^m$ is the search direction. As is well known, a proper choice for the search direction (d^k_x, d^k_λ) guarantees a superlinear convergence rate in the unconstrained minimization of L_a. However, the use of the Newton direction is not suitable for the function L_a due to the fact that, by construction, L_a is not twice continuously differentiable everywhere in $\mathcal{P} \times \mathbb{R}^m$, and due also to the fact that the evaluation of the Hessian matrix $\nabla^2 L_a$, where it exists, requires the evaluation of the third order derivatives of the problem functions f and g.

Here we assume, for analytical purposes, that the problem functions are three times continuously differentiable, and that $s > 2$ in (2.1). However, we will describe some directions which can be used to produce sequences $\{(x^k, \lambda^k)\}$ which are locally convergent with a superlinear rate of convergence without requiring the evaluation of third order derivatives. In particular, we consider search directions which satisfy the following assumption.

Assumption A5. The direction $d^k \in \mathbb{R}^{n+m}$ satisfies a system of the kind

$$\tilde{H}(x^k, \lambda^k; \epsilon) d = -\nabla L_a(x^k, \lambda^k; \epsilon), \tag{8.3}$$

where the matrix $\tilde{H}(x^k, \lambda^k; \epsilon)$ has the property that if the sequence $\{(x^k, \lambda^k)\}$ converges to a KKT pair $(\bar{x}, \bar{\lambda})$ for problem (P), then

$$\lim_{k\to\infty} \operatorname{dist}\big[\tilde{H}(x^k, \lambda^k; \epsilon) \big| \partial_B^2 L_a(x^k, \lambda^k; \epsilon)\big] = 0. \tag{8.4}$$

Search directions which satisfy Assumption A5 play a fundamental role in defining efficient iteration maps T, as is shown by the following proposition.

PROPOSITION 8.1. *Let $(\bar{x}, \bar{\lambda})$ be a KKT pair which satisfies the strong second order sufficient condition for problem* (P), *and assume that at $\bar{x}$ the gradients of the active constraints are linearly independent. If the vectors in the sequence $\{d^k\}$ satisfy Assumption* A5, *then there exists an $\bar{\epsilon}$ such that, for all $\epsilon \in (0, \bar{\epsilon}]$,*

(a) *there exists a neighborhood $\mathcal{B}(\bar{x}, \bar{\lambda})$ of $(\bar{x}, \bar{\lambda})$ such that, for all $(x^k, \lambda^k) \in \mathcal{B}(\bar{x}, \bar{\lambda})$,*

- *the search direction d^k satisfies the conditions*

$$\nabla L_a(x^k, \lambda^k; \epsilon)' d^k \le -c\|\nabla L_a(x^k, \lambda^k; \epsilon)\|^2,$$
$$c\|d^k\| \le \|\nabla L_a(x^k, \lambda^k; \epsilon)\|,$$

  where c is a positive constant;
- *an Armijo-type linesearch accepts the unit stepsize;*

(b) *if the sequence $\{(x^k, \lambda^k)\}$ given by* (8.1) *and* (8.2) *with $\theta^k = 1$ converges to $(\bar{x}, \bar{\lambda})$, then the rate of convergence is superlinear.*

Proof. Point (a) follows from Assumption A5, Theorem 6.3, and the results given in [15]. Point (b) follows again from Assumption A5, Theorem 6.3, and [28, Theorem 2]. □

In particular, point (a) of Proposition 8.1 shows that standard globalization techniques will eventually produce x_{k+1} by performing a unit stepsize along the direction d^k. This fact and point (b) of the same proposition show that the use of search directions satisfying Assumption A5 allows us to define algorithms for solving problem (P) which combine the global convergence with a superlinear rate of convergence.

The proposal of a specific globally and superlinearly convergent algorithm for problem (P) is beyond the scope of this paper. Here we limit ourselves to indicating, as examples, some directions which satisfy Assumption A5.

To this aim, we consider at the point (x^k, λ^k) the estimates of the sets of the active and nonactive constraints $A(x^k, \lambda^k; \epsilon)$ and $N(x^k, \lambda^k; \epsilon)$ given by

$$A(x^k, \lambda^k; \epsilon) = \{i : g_i(x^k) \ge -\epsilon p(x^k, \lambda^k)\lambda_i^k\}, \tag{8.5}$$
$$N(x^k, \lambda^k; \epsilon) = \{i : g_i(x^k) < -\epsilon p(x^k, \lambda^k)\lambda_i^k\} \tag{8.6}$$

and denoted by the short notations A^k and N^k; we recall that in [16] it has been shown that, for every value of the penalty parameter ϵ, there exists a neighborhood $\mathcal{B}$ of a KKT point $(\bar{x}, \bar{\lambda})$ such that for all $(x^k, \lambda^k) \in \mathcal{B}$ the set A^k satisfies

$$I_+(\bar{x}, \bar{\lambda}) \subseteq A^k \subseteq I_0(\bar{x}). \tag{8.7}$$

The first search direction that we consider has been proposed in [10]. At each iteration, this direction is computed by solving the linear system

$$H(x^k, \lambda^k; \epsilon, A^k) d = -\nabla L_a(x^k, \lambda^k; \epsilon), \tag{8.8}$$

where the matrix $H(x^k, \lambda^k; \epsilon, A^k)$ is given blockwise by (6.1)–(6.3). By Theorem 6.3, system (8.8) is well defined in a neighborhood of a KKT pair which satisfies the strong

second order sufficient condition and where the gradients of the active constraints are linearly independent. Then, by Proposition 6.1 and (8.7), it is immediate to verify that the direction given by solving system (8.8) satisfies Assumption A5.

As a second example we consider the search direction proposed in [10] and [1] (and further studied in [16]). At each iteration, this direction depends again on the estimates A^k and N^k of the index sets of the active and nonactive constraints given by (8.5) and (8.6). Given the vectors p^k and z^k obtained by solving the linear system

$$\begin{bmatrix} \nabla_x^2 L(x^k, \lambda^k) & \nabla g_{A^k}(x^k) \\ \nabla g_{A^k}(x^k)' & 0 \end{bmatrix} \begin{bmatrix} p \\ z \end{bmatrix} = - \begin{bmatrix} \nabla f(x^k) \\ g_{A^k}(x^k) \end{bmatrix}, \tag{8.9}$$

the search direction $d^k = (d_x^k, d_\lambda^k)$ is defined in the following way:

$$d_x^k = p^k, \tag{8.10}$$

$$d_{\lambda_{A^k}}^k = z^k - \lambda_{A^k}^k, \tag{8.11}$$

$$d_{\lambda_{N^k}}^k = -\lambda_{N^k}^k. \tag{8.12}$$

In the next proposition we state that the direction d^k given by (8.10)–(8.12) satisfies Assumption A5. The proof is given in the appendix.

PROPOSITION 8.2. *If the linear system* (8.9) *is well defined at the kth iteration, then the direction* d^k *given by* (8.10)–(8.12) *satisfies Assumption* A5.

For the direction d^k given by (8.10)–(8.12), it is possible to state a stronger convergence rate result. In fact in [16] the following proposition is proved.

PROPOSITION 8.3. *Let* $(\bar{x}, \bar{\lambda})$ *be a KKT pair for problem* (P) *which satisfies the strong second order sufficient condition, and assume that in* $\bar{x}$ *the gradients of the active constraints are linearly independent. Then, for any* $\epsilon > 0$, *there exists a neighborhood* $\mathcal{B}(\bar{x}, \bar{\lambda})$ *of* $(\bar{x}, \bar{\lambda})$ *such that if* $(x^0, \lambda^0) \in \mathcal{B}(\bar{x}, \bar{\lambda})$, *the system* (8.9) *is well defined, and the sequence* $\{(x^k, \lambda^k)\}$ *produced by* (8.1)–(8.2), *with* $\theta^k = 1$ *and* d^k *given by* (8.10)–(8.12), *satisfies* $(x^k, \lambda^k) \in \mathcal{B}(\bar{x}, \bar{\lambda})$ *for all k, converges to* $(\bar{x}, \bar{\lambda})$, *and has a rate of convergence which is quadratic* (*superlinear for the sequence* $\{x^k\}$).

Finally we may consider a direction given by the usual SQP approach. In this case the direction d_x^k is the solution of the quadratic subproblem

$$\begin{aligned} \min_d \ & \frac{1}{2} d' \nabla^2 L(x^k, \lambda^k) d + \nabla f(x^k)' d \\ \text{s.t.} \ & \nabla g(x^k)' d + g(x^k) \le 0, \end{aligned} \tag{8.13}$$

and

$$d_\lambda^k = \mu^k - \lambda^k, \tag{8.14}$$

where μ^k is the KKT multiplier associated with the solution d_x^k of subproblem (8.13).

The local behavior of the SQP iteration has been widely studied (see, e.g., [20, 30, 18, 3]). Here we report the following proposition which shows that, under suitable assumptions, the search direction produced by an SQP approach also satisfies Assumption A5. The proof of this proposition is in the appendix.

PROPOSITION 8.4. *Let* $(\bar{x}, \bar{\lambda})$ *be a KKT pair for problem* (P) *that satisfies the strict complementarity assumption and assume that there exists a neighborhood of* $(\bar{x}, \bar{\lambda})$ *where the solution of subproblem* (8.13) *exists and is continuous. Then the direction* d^k, *obtained from* (8.13)–(8.14), *satisfies Assumption* A5.

Appendix.

Proof of Proposition 4.3. By the linear independence of the gradients of the active constraints, we have that the matrix $M(\hat{x})$ is positive definite. Let $\mathcal{B}$ be a neighborhood of $\hat{x}$ where the matrix $M(x)$ is nonsingular. If $\|\nabla_\lambda L_a(x,\lambda;\epsilon)\| \leq \|\max\{g(x), -\epsilon p(x,\lambda)\lambda\}\|$ by (2.9) we can write

$$\Big\| \max\{g(x), -\epsilon p(x,\lambda)\lambda\} + \frac{1}{\epsilon a(x)} \|\max\{g(x), -\epsilon p(x,\lambda)\lambda\}\|^2 \lambda + 2M(x)\left(\nabla g(x)' \nabla_x L(x,\lambda) + G(x)^2\lambda\right) \Big\| \leq \|\max\{g(x), -\epsilon p(x,\lambda)\lambda\}\|,$$

and if $x \in \mathcal{B}$, we can obtain

$$\|\nabla g(x)' \nabla_x L(x,\lambda) + G(x)^2\lambda\| \leq \|M(x)^{-1}\| \left\{1 + \frac{1}{2\epsilon a(x)} \|\lambda\| \|\max\{g(x), -\epsilon p(x,\lambda)\lambda\}\|\right\} \|\max\{g(x), -\epsilon p(x,\lambda)\lambda\}\|. \tag{A.1}$$

Now, omitting the arguments to simplify the notation, and recalling that

$$\mathrm{Max}\{g, -\epsilon p\lambda\} = \mathrm{diag}(\max\{g_i, \epsilon p \lambda_i\}),$$

by (2.8) we can write

$$\begin{aligned} \|\nabla g' \nabla_x L_a\| = \Bigg\| & \nabla g' \nabla_x L + G^2\lambda - \left[G\Lambda - \frac{1}{\epsilon p}\big(G^2 - G\ \mathrm{Max}\{g, -\epsilon p\lambda\}\big)\right] \max\{g, -\epsilon p\lambda\} \\ & + \frac{1}{\epsilon p} \nabla g' \nabla g \max\{g, -\epsilon p\lambda\} \\ & + \frac{s}{2\epsilon a p} \sum_{i=1}^{m} \nabla g' \nabla g_i \max\{g_i, 0\}^{(s-1)} \|\max\{g, -\epsilon p\lambda\}\|^2 \\ & + \nabla g' Q\big[\nabla g' \nabla_x L + G^2\lambda\big] \Bigg\|, \end{aligned} \tag{A.2}$$

where we have used the equality

$$G\lambda = \Lambda \max\{g, -\epsilon p\lambda\} + \frac{1}{\epsilon p}\big[\mathrm{Max}\{g, -\epsilon p\lambda\} - G\big] \max\{g, -\epsilon p\lambda\}; \tag{A.3}$$

hence we can write

$$\epsilon p \|\nabla g' \nabla_x L_a\| = \big\|\Gamma \max\{g, -\epsilon p\lambda\} + \epsilon p\big(I + \nabla g' Q\big)\big[\nabla g' \nabla_x L + G^2\lambda\big]\big\|,$$

where the matrix $\Gamma(x,\lambda;\epsilon)$ is given by

$$\begin{aligned} \Gamma(x,\lambda;\epsilon) = & \big[\nabla g' \nabla g + \big(G^2 - G\ \mathrm{Max}\{g, -\epsilon p\lambda\}\big)\big] - \epsilon p G\Lambda \\ & + \frac{s}{2a} \sum_{i=1}^{m} \nabla g' \nabla g_i \max\{g_i, 0\}^{(s-1)} \max\{g, -\epsilon p\lambda\}'. \end{aligned}$$

Therefore, employing (A.1), we have

$$\epsilon p \|\nabla g' \nabla_x L_a\| \geq \|\Gamma \max\{g, -\epsilon p\lambda\}\| - \epsilon p \|\big(I + \nabla g' Q\big)\| \|M^{-1}\| \left\{1 + \frac{1}{2\epsilon a} \|\lambda\| \|\max\{g, -\epsilon p\lambda\}\|\right\} \|\max\{g, -\epsilon p\lambda\}\|,$$

from which we obtain

$$\epsilon\eta\|\nabla_x L_a\| \geq \epsilon p\|\nabla g'\nabla_x L_a\| \tag{A.4}$$
$$\geq \left[\sigma_m(\Gamma) - \|\big(I+\nabla g'Q\big)\|\|M^{-1}\|\left\{\epsilon p + \frac{1}{2a}\|p\lambda\|\|\max\{g,-\epsilon p\lambda\}\|\right\}\right]\|\max\{g,-\epsilon p\lambda\}\|,$$

where $\sigma_m(\Gamma)$ is the smallest singular value of Γ, and

$$\eta > \max_{x\in\mathcal{C}} |a(x)|\|\nabla g(x)\|,$$

with $\mathcal{C}$ defined in Theorem 3.3. Now we note that, if $\hat{x} \in \mathcal{F}$, then for all $\lambda \in \mathbb{R}^m$ we have

$$\Gamma(\hat{x},\lambda;0) = \left[\nabla g(\hat{x})'\nabla g(\hat{x}) + G(\hat{x})^2\right] = M(\hat{x}),$$

so that the matrix $\Gamma(\hat{x},\lambda;0)$ is positive definite and $\sigma_m(\Gamma(\hat{x},\lambda;0))$ is strictly positive. Moreover the term

$$\left\{\epsilon p(x,\lambda) + \frac{1}{2a}\|p(x,\lambda)\lambda\|\|\max\{g(x),-\epsilon p(x,\lambda)\lambda\}\|\right\} \tag{A.5}$$

in (A.4) vanishes for $\epsilon = 0$ and $\hat{x} \in \mathcal{F}$.

By Theorem 3.3, the points x such that $(x,\lambda) \in \Omega(x_0,\lambda_0;\epsilon)$ belong to a compact set $\mathcal{C}$ which does not depend on ϵ. This and the expression of $p(x,\lambda)$ imply that both $p(x,\lambda)$ and $p(x,\lambda)\lambda$ are bounded for all $(x,\lambda) \in \mathcal{C}\times R^m$.

Then we can see that, for all "sufficiently small" values of ϵ, for all x in a "sufficiently small" neighborhood of $\hat{x} \in \mathcal{F}$, and for all $\lambda \in R^m$, the term $\sigma_m(\Gamma)$ is "sufficiently positive" and the term (A.5) is "sufficiently small" so as to make the term in square brackets in (A.4) strictly positive. More formally we can say that there exist numbers $\epsilon(\hat{x}) > 0$, $\sigma(\hat{x}) > 0$, and $\rho(\hat{x}) > 0$ such that for all $\epsilon \in (0,\epsilon(\hat{x})]$ and for all $(x,\lambda) \in \Omega(x_0,\lambda_0;\epsilon)$ satisfying $\|x-\hat{x}\| \leq \sigma(\hat{x})$ and $\|\nabla_\lambda L_a(x,\lambda;\epsilon)\| \leq \|\max\{g(x),-\epsilon p(x,\lambda)\lambda\}\|$, it holds that

$$\frac{1}{\eta}\left[\sigma_m(\Gamma) - \|\big(I+\nabla g'Q\big)\|\|M^{-1}\|\left\{\epsilon p + \frac{1}{2a}\|p\lambda\|\|\max\{g,-\epsilon p\lambda\}\|\right\}\right] \tag{A.6}$$
$$\geq \rho(\hat{x}) > 0.$$

By (A.4) and (A.6) we get (4.5), so that the proposition is proved. □

Proof of Proposition 6.1. Let $(\bar{x},\bar{\lambda})$ be a KKT pair of problem (P). Consider a point (x,λ) in a neighborhood $\mathcal{B}$ of $(\bar{x},\bar{\lambda})$ and a sequence $\{(x^k,\lambda^k)\}$ converging to (x,λ) such that the Hessian of L_a exists in (x^k,λ^k). By looking at (2.8)–(2.9) we note that ∇L_a is differentiable in (x^k,λ^k) either if we have $g_i(x^k) \neq -\epsilon p(x^k,\lambda^k)\lambda_i^k$ for all i or if for every i such that $g_i(x^k) = -\epsilon p(x^k,\lambda^k)\lambda_i$ we also have $-\epsilon\nabla_x p(x^k,\lambda^k)\lambda_i = \nabla_x g_i(x^k)$ and $-\epsilon\nabla_\lambda\left(p(x^k,\lambda^k)\lambda_i\right) = 0$. Let A^k and N^k indicate the sets given by

$$A^k = \{i : g_i(x^k) \geq -\epsilon p(x^k,\lambda^k)\lambda_i^k\},$$
$$N^k = \{i : g_i(x^k) < -\epsilon p(x^k,\lambda^k)\lambda_i^k\};$$

by partitioning the vectors g and λ according to the index sets A^k and N^k, ∇L_a can also be written in the following way:

$$
\begin{aligned}
\nabla_x L_a(x^k,\lambda^k;\epsilon) &= \nabla_x L(x^k,\lambda^k) + \frac{1}{\epsilon p(x^k,\lambda^k)}\nabla g_{A^k}(x^k) g_{A^k}(x^k) - \nabla g_{N^k}(x^k)\lambda^k_{N^k} \\
&+ \frac{s}{2\epsilon a(x^k) p(x^k,\lambda^k)}\Bigg[\|g_{A^k}(x^k)\|^2 \\
&\qquad + \epsilon^2 p(x^k,\lambda^k)^2 \|\lambda^k_{N^k}\|^2\Bigg] \sum_{i=1}^{m} \nabla g_i(x^k)\max\{g_i(x^k),0\}^{s-1} \\
&+ Q(x^k,\lambda^k)\Big[\nabla g(x^k)'\nabla_x L(x^k,\lambda^k) + G(x^k)^2\lambda^k\Big], \qquad \text{(A.7)} \\
\nabla_\lambda L_a(x^k,\lambda^k;\epsilon) &= \begin{bmatrix} g_{A^k}(x^k) \\ 0 \end{bmatrix} - \epsilon p(x^k,\lambda^k)\begin{bmatrix} 0 \\ \lambda^k_{N^k} \end{bmatrix} \\
&+ \frac{1}{\epsilon a(x^k)}\Big[\|g_{A^k}(x^k)\|^2 + \epsilon^2 p(x^k,\lambda^k)^2\|\lambda^k_{N^k}\|^2\Big]\lambda^k \\
&+ 2\Big[\nabla g(x^k)'\nabla g(x^k) + G(x^k)^2\Big]\Big[\nabla g(x^k)'\nabla_x L(x^k,\lambda^k) + G(x^k)^2\lambda^k\Big]. \qquad \text{(A.8)}
\end{aligned}
$$

Then the Hessian of L_a in (x^k,λ^k) can be obtained by differentiating (A.7)–(A.8), and this yields

$$\nabla^2 L_a(x^k,\lambda^k;\epsilon) = H(x^k,\lambda^k;\epsilon,A^k) + K(x^k,\lambda^k;\epsilon,A^k),$$

where $H(x^k,\lambda^k;\epsilon,A^k)$ is given by (6.1)–(6.3) and $K(x^k;\lambda^k;\epsilon,A^k)$ represents the sum of matrices whose terms contain as a factor either a component of $g_{A^k}(x^k)$, a component $\lambda^k_{N^k}$, or a component of $\nabla_x L(x^k,\lambda^k)$. Now, for sufficiently large values of k (see [16]),

$$I_+(\bar{x},\bar{\lambda}) \subseteq A^k \subseteq I_0(\bar{x}), \qquad \text{(A.9)}$$

which implies both that $g_{A^k}(x^k)$ and $\lambda^k_{N^k}$ go to 0 as $(x^k,\lambda^k)\to(\bar{x},\bar{\lambda})$ and that, for sufficiently large k, $A^k \in \mathcal{A}$. These facts, recalling the definition of $\partial^2 L_a(x,\lambda;\epsilon)$, imply that

$$\partial^2_B L_a(x,\lambda;\epsilon) \subseteq \{H(x,\lambda;\epsilon,A) + K(x,\lambda;\epsilon,A) : A \in \mathcal{A}\}.$$

Now to prove that the two sets of the above inclusion coincide we show that for every index set $A \in \mathcal{A}$ it is possible to find a sequence $\{(x^k,\lambda^k)\}$, converging to $(\bar{x},\bar{\lambda})$, such that

$$
\begin{aligned}
&\{i : g_i(x^k) > -\epsilon p(x^k,\lambda^k)\lambda^k_i\} = A, \\
&\{i : g_i(x^k) < -\epsilon p(x^k,\lambda^k)\lambda^k_i\} = \{1,\dots,m\}\setminus A
\end{aligned}
$$

and such that

$$\nabla^2 L_a(x^k,\lambda^k;\epsilon) = H(x^k,\lambda^k;\epsilon,A) + K(x^k;\lambda^k;\epsilon,A).$$

Let us denote by N the index set $\{1,\dots,m\}\setminus A$; recalling the definition of A we can write

$$
\begin{aligned}
A &= A_1 \cup A_2, \\
N &= N_1 \cup N_2,
\end{aligned}
$$

where

$$A_1 = \{i \in A : g_i(\bar{x}) = 0, \bar{\lambda}_i > 0\},$$
$$A_2 = \{i \in A : g_i(\bar{x}) = 0, \bar{\lambda}_i = 0\},$$
$$N_1 = \{i \in N : \bar{\lambda}_i = 0, g_i(\bar{x}) < 0\},$$
$$N_2 = \{i \in N : \bar{\lambda}_i = 0, g_i(\bar{x}) = 0\}.$$

Now, for all (x^k, λ^k) sufficiently close to $(\bar{x}, \bar{\lambda})$ we have

$$\{i : g_i(x^k) > -\epsilon p(x^k, \lambda^k)\lambda_i^k\} \supseteq A_1, \tag{A.10}$$
$$\{i : g_i(x^k) < -\epsilon p(x^k, \lambda^k)\lambda_i^k\} \supseteq N_1. \tag{A.11}$$

Then, to conclude the proof, we show that it is possible to refine further the choice of the points (x^k, λ^k) so that the following inclusions also hold:

$$\{i : g_i(x^k) > -\epsilon p(x^k, \lambda^k)\lambda_i^k\} \supseteq A_2, \tag{A.12}$$
$$\{i : g_i(x^k) < -\epsilon p(x^k, \lambda^k)\lambda_i^k\} \supseteq N_2. \tag{A.13}$$

Let $\delta > 0$ be a number such that $|\bar{\lambda}_i| \leq \delta$ for all $i = 1, \ldots, m$. Since $g_i(\bar{x}) = 0$ for all $i \in A_2 \cup N_2$, we can find a sequence $\{x^k\}$ such that, for all k and for all $i \in A_2 \cup N_2$, we have

$$\frac{2|g_i(x^k)|}{\epsilon a(x^k)} \leq \frac{\delta}{1 + m\delta^2}. \tag{A.14}$$

In correspondence to the sequence $\{x^k\}$, we consider a sequence $\{\lambda^k\}$ converging to $\bar{\lambda}$ such that

$$|\lambda_i^k| \leq \delta, \quad i \in A_1 \cup N_1, \tag{A.15}$$
$$\lambda_i^k = \max\left\{\frac{2|g_i(x^k)|(1 + m\delta^2)}{\epsilon a(x^k)}, \frac{\delta}{k}\right\}, \quad i \in A_2, \tag{A.16}$$
$$\lambda_i^k = -\max\left\{\frac{2|g_i(x^k)|(1 + m\delta^2)}{\epsilon a(x^k)}, \frac{\delta}{k}\right\}, \quad i \in N_2. \tag{A.17}$$

Recalling (A.14), $|\lambda_i^k| \leq \delta$ for all $i = 1, \ldots, m$, and this implies that

$$\|\lambda^k\|^2 \leq m\delta^2. \tag{A.18}$$

Now, by using (A.16), (A.18), and the definition of $p(x^k, \lambda^k)$, we have for all $i \in A_2$

$$\begin{aligned} -\lambda_i^k &= -\max\left\{\frac{2|g_i(x^k)|(1 + m\delta^2)}{\epsilon a(x^k)}, \frac{\delta}{k}\right\} \leq -\max\left\{\frac{2|g_i(x^k)|(1 + \|\lambda^k\|^2)}{\epsilon a(x^k)}, \frac{\delta}{k}\right\} \\ &< -\frac{|g_i(x^k)|(1 + \|\lambda^k\|^2)}{\epsilon a(x^k)} \leq \frac{g_i(x^k)(1 + \|\lambda^k\|^2)}{\epsilon a(x^k)} = \frac{g_i(x^k)}{\epsilon p(x^k, \lambda^k)}, \end{aligned}$$

which proves (A.12). In a similar way, by using (A.17), we have for all $i \in N_2$

$$\begin{aligned} -\lambda_i^k &= \max\left\{\frac{2|g_i(x^k)|(1 + m\delta^2)}{\epsilon a(x^k)}, \frac{\delta}{k}\right\} \geq \max\left\{\frac{2|g_i(x^k)|(1 + \|\lambda^k\|^2)}{\epsilon a(x^k)}, \frac{\delta}{k}\right\} \\ &> \frac{|g_i(x^k)|(1 + \|\lambda^k\|^2)}{\epsilon a(x^k)} \geq \frac{g_i(x^k)(1 + \|\lambda^k\|^2)}{\epsilon a(x^k)} = \frac{g_i(x^k)}{\epsilon p(x^k, \lambda^k)}, \end{aligned}$$

which proves (A.13). Hence we have shown that there exists a sequence $\{(x^k, \lambda^k)\}$, converging to $(\bar{x}, \bar{\lambda})$, such that

$$\{i : g_i(x^k) > -\epsilon p(x^k, \lambda^k)\lambda_i^k\} \supseteq A_1 \cup A_2 = A, \tag{A.19}$$

$$\{i : g_i(x^k) < -\epsilon p(x^k, \lambda^k)\lambda_i^k\} \supseteq N_1 \cup N_2 = N. \tag{A.20}$$

Noting that $\{A, N\}$ constitutes a partition of the set $\{1, \dots, m\}$ we must have

$$\{i : g_i(x^k) > -\epsilon p(x^k, \lambda^k)\lambda_i^k\} = A,$$
$$\{i : g_i(x^k) < -\epsilon p(x^k, \lambda^k)\lambda_i^k\} = N,$$

and this completes the proof of the proposition. □

Proof of Proposition 8.2. First we note that (8.9)–(8.12) yield (omitting the arguments and the index k)

$$\begin{aligned} \nabla_x^2 L d_x + \nabla g_A d_{\lambda_A} + \nabla g_N d_{\lambda_N} &= -\nabla_x L, \\ \nabla g_A d_x &= -g_A, \\ d_{\lambda_N} &= -\lambda_N. \end{aligned} \tag{A.21}$$

From these equalities and recalling (A.7)–(A.8) we have

$$\begin{aligned} -\nabla_x L_a = {} & \nabla_x^2 L d_x + \nabla g_A d_{\lambda_A} + \frac{1}{\epsilon p} \nabla g_A \nabla g_A' d_x \\ & + \frac{s}{2\epsilon a p} \sum_{i=1}^{m} \nabla g_i \max\{g_i, 0\}^{s-1} (g_A' \nabla g_A' d_x + \epsilon^2 p^2 \lambda_N' d_{\lambda_N}) \\ & + 2 \left[\nabla_x^2 L \nabla g + \sum_{i=1}^{m} \nabla_x^2 g_i \nabla_x L e_i' + 2 \nabla g G \Lambda \right] \\ & \times \left[\nabla g' \left(\nabla_x^2 L d_x + \nabla g d_\lambda \right) + \begin{pmatrix} G_A \Lambda_A \nabla g_A' d_x \\ G_N^2 d_{\lambda_N} \end{pmatrix} \right], \end{aligned} \tag{A.22}$$

$$\begin{aligned} -\nabla_\lambda L_a = {} & \begin{bmatrix} \nabla g_A' \\ 0 \end{bmatrix} d_x - \epsilon p \begin{bmatrix} 0 \\ I_N \end{bmatrix} d_{\lambda_N} \\ & + \frac{1}{\epsilon a} \left(\begin{bmatrix} \lambda_A g_A' \nabla g_A' \\ 0 \end{bmatrix} d_x + \|g_A\|^2 \begin{bmatrix} 0 \\ I_N \end{bmatrix} d_{\lambda_N} + \epsilon^2 p^2 \lambda \lambda_N' d_{\lambda_N} \right) \\ & + 2 \left[\nabla g' \nabla g + G^2 \right] \left[\nabla g' \left(\nabla_x^2 L d_x + \nabla g d_\lambda \right) + \begin{pmatrix} G_A \Lambda_A \nabla g_A' d_x \\ G_N^2 d_{\lambda_N} \end{pmatrix} \right]. \end{aligned} \tag{A.23}$$

Reordering the terms in (A.22)–(A.23) we obtain

$$-\nabla L_a(x, \lambda; \epsilon) = \Big[H(x, \lambda; \epsilon, A(x, \lambda; \epsilon)) + R(x, \lambda; \epsilon) \Big] d, \tag{A.24}$$

where the matrix $H(x, \lambda; \epsilon, A(x, \lambda; \epsilon))$ is given by (6.1)–(6.3) and the matrix R satisfies $\|R(x, \lambda; \epsilon)\| \le \hat{\rho}(x, \lambda)$, with $\hat{\rho}(x, \lambda)$ a positive continuous function such that $\hat{\rho}(\bar{x}, \bar{\lambda}) = 0$ at every KKT pair $(\bar{x}, \bar{\lambda})$. Then the proof of the proposition follows from Proposition 6.1 and (8.7). □

Proof of Proposition 8.4. The proof follows from the proof of Proposition 8.2 by noting that, under the assumptions stated and for any ϵ, there exists a neighborhood of $(\bar{x}, \bar{\lambda})$ where we have

$$A(x, \lambda; \epsilon) = \{i : \nabla g_i(x) d_x = -g_i(x)\},$$
$$N(x, \lambda; \epsilon) = \{i : (d_\lambda)_i = -\lambda_i\}. \qquad \square$$

Acknowledgments. We wish to thank three anonymous referees and the editor of the journal for their careful reading of the paper and for their constructive comments that contributed greatly to improving the paper.

REFERENCES

[1] D. P. Bertsekas, *Constrained Optimization and Lagrange Multipliers Methods*, Academic Press, New York, 1982.
[2] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed., Athena Scientific, Belmont, MA, 1999.
[3] J. F. Bonnans, *Rates of convergence of Newton type methods for variational inequalities and nonlinear programming*, Appl. Math. Optim., 29 (1994), pp. 161–186.
[4] D. Boukari and A. V. Fiacco, *Survey of penalty, exact-penalty and multiplier methods from* 1968 *to* 1993, Optimization, 32 (1995), pp. 301–334.
[5] J. V. Burke, *A sequential quadratic programming method for potentially infeasible mathematical programs*, J. Math. Anal. Appl., 139 (1989), pp. 319–351.
[6] J. V. Burke and S. P. Han, *A robust sequential quadratic programming method*, Math. Program., 43 (1989), pp. 277–303.
[7] F. H. Clarke, *Optimization and Nonsmooth Analysis*, John Wiley, New York, 1983.
[8] G. Di Pillo, *Exact penalty methods*, in Algorithms for Continuous Optimization: The State of the Art, E. Spedicato, ed., Kluwer, Boston, 1994, pp. 203–253.
[9] G. Di Pillo and L. Grippo, *An augmented Lagrangian for inequality constraints in nonlinear programming problems*, J. Optim. Theory Appl., 36 (1982), pp. 495–519.
[10] G. Di Pillo and L. Grippo, *A class of continuously differentiable exact penalty function algorithms for nonlinear programming problems*, in System Modelling and Optimization, E. P. Toft-Christensen, ed., Springer-Verlag, Berlin, 1984, pp. 246–256.
[11] G. Di Pillo and L. Grippo, *A continuously differentiable exact penalty function for nonlinear programming problems with inequality constraints*, SIAM J. Control Optim., 23 (1985), pp. 72–84.
[12] G. Di Pillo and L. Grippo, *An exact penalty method with global convergence properties*, Math. Programming, 36 (1986), pp. 1–18.
[13] G. Di Pillo and L. Grippo, *Exact penalty functions in constrained optimization*, SIAM J. Control Optim., 27 (1989), pp. 1333–1360.
[14] G. Di Pillo and S. Lucidi, *On exact augmented Lagrangian functions in nonlinear programming*, in Nonlinear Optimization and Applications, G. Di Pillo and F. Giannessi, eds., Plenum Press, New York, 1996, pp. 85–100.
[15] F. Facchinei, *Minimization of SC^1 functions and the Maratos effect*, Oper. Res. Lett., 17 (1995), pp. 131–137.
[16] F. Facchinei and S. Lucidi, *Quadratically and superlinearly convergent algorithms for the solution of inequality constrained minimization problems*, J. Optim. Theory Appl., 85 (1995), pp. 265–289.
[17] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley, New York, 1968.
[18] R. Fletcher, *Practical Methods of Optimization*, John Wiley, New York, 1987.
[19] T. Glad and E. Polak, *A multiplier method with automatic limitation of penalty growth*, Math. Programming, 17 (1979), pp. 140–155.
[20] S. P. Han, *Superlinearly convergent variable metric algorithms for general nonlinear programming problems*, Math. Programming, 11 (1976), pp. 263–282.
[21] J. B. Hiriart-Urruty, J. J. Strodiot, and V. H. Nguyen, *Generalized Hessian matrix and second-order optimality conditions for problems with $C^{1,1}$ data*, Appl. Math. Optim., 11 (1984), pp. 43–56.
[22] J. Guddat, F. Guerra Vasquez, and H. Th. Jongen, *Parametric Optimization: Singularities, Pathfollowing and Jumps*, John Wiley, Chichester, UK, 1990.
[23] D. Klatte and K. Tammer, *On second-order sufficient optimality conditions for $C^{1,1}$-optimization problems*, Optimization, 19 (1988), pp. 169–179.
[24] S. Lucidi, *New results on a class of exact augmented Lagrangians*, J. Optim. Theory Appl., 58 (1988), pp. 259–282.
[25] S. Lucidi, *New results on a continuously differentiable exact penalty function*, SIAM J. Optim., 2 (1992), pp. 558–574.
[26] O. L. Mangasarian and S. Fromovitz, *The Fritz John necessary optimality conditions in the presence of equality constraints*, J. Math. Anal. Appl., 17 (1967), pp. 34–47.
[27] J. S. Pang, *Serial and parallel computation of Karush-Kuhn-Tucker points via nonsmooth equations*, SIAM J. Optim., 4 (1994), pp. 872–893.

[28] J. S. Pang and L. Qi, *Nonsmooth equations: Motivation and algorithms*, SIAM J. Optim., 3 (1993), pp. 443–465.
[29] L. Qi and J. Sun, *A nonsmooth version of Newton's method*, Math. Programming, 58 (1993), pp. 353–367.
[30] S. M. Robinson, *Perturbed Kuhn-Tucker points and rates of convergence for a class of nonlinear-programming algorithms*, Math. Programming, 7 (1974), pp. 1–16.

SIAM J. OPTIM.
Vol. 12, No. 2, pp. 407–420

AN APPROPRIATE SUBDIFFERENTIAL FOR QUASICONVEX FUNCTIONS*

ARIS DANIILIDIS†, NICOLAS HADJISAVVAS‡, AND JUAN-ENRIQUE MARTÍNEZ-LEGAZ§

Abstract. In this paper we introduce and study a subdifferential that is related to the quasiconvex functions, much as the Fenchel–Moreau subdifferential is related to the convex ones. It is defined for any lower semicontinuous function, through an appropriate combination of an abstract subdifferential and the normal cone to sublevel sets. We show that this "quasiconvex" subdifferential is always a cyclically quasimonotone operator that coincides with the Fenchel–Moreau subdifferential whenever the function is convex, and that under mild assumptions, the density of its domain in the domain of the function is equivalent to the quasiconvexity of the function. We also show that the "quasiconvex" subdifferential of a lower semicontinuous function contains the derivatives of its differentiable quasiaffine supports. As a consequence, it contains the subdifferential introduced by Martínez-Legaz and Sach in a recent paper [*J. Convex Anal.*, 6 (1999), pp. 1–12]. Several other properties and calculus rules are also established.

Key words. subdifferential, quasiconvex function, nonsmooth analysis, quasimonotone operator

AMS subject classifications. 26B25, 26E15, 90C26, 49J52

PII. S1052623400371399

1. Introduction. In the last thirty years, several notions of subdifferentials for quasiconvex functions have been proposed. The oldest ones are the Greenberg–Pierskalla subdifferential [6] and the tangential introduced by Crouzeix [4]. These two subdifferentials have in common that they are convex cones, and are therefore too large to give enough information on the function. The lower subdifferential of Plastria [13] is smaller but still unbounded, as are the related α-lower subdifferentials [10]. All of these subdifferentials arise in the context of some quasiconvex conjugation scheme. Of a different nature is the weak lower subdifferential [9], which is more in the spirit of nonsmooth analysis in that its support function partially coincides with the directional derivative; however, this set is not quite satisfactory either, as it is even bigger than the lower subdifferential of Plastria. Trying to remedy this drawback, Martínez-Legaz and Sach [11] recently introduced the Q-subdifferential. Given that it is a subset of the Greenberg–Pierskalla subdifferential, it shares with all other quasiconvex subdifferentials the property that its nonemptiness on the domain of a lower semicontinuous function implies quasiconvexity of the function, which justifies the claim that it is a quasiconvex subdifferential; on the other hand, unlike all other subdifferentials previously introduced in quasiconvex analysis, it can be regarded as

*Received by the editors April 17, 2000; accepted for publication (in revised form) April 11, 2001; published electronically November 13, 2001.

http://www.siam.org/journals/siopt/12-2/37139.html

†Laboratoire de Mathématiques Appliquées, Université de Pau et des Pays de l'Adour, Avenue de l'Université, 64000 Pau, France (aris.daniilidis@univ-pau.fr). This author's research was supported by the TMR postdoctoral grant ERBFMBI CT 983381.

‡Department of Mathematics, University of the Aegean, Karlovassi 83200, Samos, Greece (nhad@aegean.gr).

§CODE and Departament d'Economia i d'Història Economica, Universitat Autònoma de Barcelona, Bellaterra 08193, Spain (jemartinez@selene.uab.es). This author's research was supported by the DGICYT (Spain), project PB98-0867, and by the Comissionat per Universitats i Recerca de la Generalitat de Catalunya, grant 1998SGR-00062. Part of this work was completed while this author was visiting the Department of Mathematics of the University of the Aegean (November 1999), to which he is grateful for the support received.

a small set, as it is contained in the Fréchet subdifferential. But this advantage is, at the same time, the main drawback of this subdifferential, as one has to impose rather strong assumptions on a quasiconvex function to ensure the nonemptiness of its Q-subdifferential on a dense subset of the domain.

In view of all these considerations, one can reasonably say that the problem of defining a sufficiently good subdifferential for quasiconvex functions is still open. To solve it, one has first to set the standards that such a concept should meet. In this sense, we can formulate the general principle that a quasiconvex subdifferential should be related to quasiconvex functions in a way similar to the classical Fenchel–Moreau subdifferential's relation to convex functions. Let us be more precise. The Fenchel–Moreau subdifferential is well defined for an arbitrary function, while, under mild conditions, its nonemptiness on a dense subset of the domain of a lower semicontinuous function is equivalent to convexity of the function. Similarly, a quasiconvex subdifferential should be defined for arbitrary functions, but its nonemptiness on the domain of a lower semicontinuous function should be equivalent (under mild assumptions) to quasiconvexity of the function. Another desirable property of any (quasiconvex) subdifferential is that it should reduce to the Fenchel–Moreau subdifferential in the case of convex functions. As we shall prove below, the quasiconvex subdifferential introduced in this paper satisfies all these requirements. Moreover, it is smaller than all previously defined quasiconvex subdifferentials (except the Q-subdifferential), as it is contained in the upper Dini subdifferential.

The new subdifferential is defined through an appropriate combination of an abstract subdifferential (in the sense of the axiomatic scheme of Aussel–Corvellec–Lassonde [2]) and geometrical considerations based on the notion of the normal cone to sublevel sets, in such a way that it retains important properties from both. For instance, for the class of quasiconvex functions our subdifferential is identical (under mild conditions) to the abstract subdifferential, so that it inherits the same calculus rules; on the other hand, for any continuous function f, the existence of a nonzero element of the subdifferential at x_0 implies that f is "quasiconvex with respect to x_0," in the sense that if $x_0 = \lambda x + (1-\lambda)y$, with $0 \leq \lambda \leq 1$, then $f(x_0) \leq \max\{f(x), f(y)\}$.

The rest of the paper is organized as follows. Section 2 establishes the notation and some preliminaries related to the abstract subdifferentials upon which our quasiconvex subdifferential is built. The central part of the paper is section 3, where the quasiconvex subdifferential is introduced and compared with other subdifferentials, and its main properties are discussed.

2. Notation and preliminaries. In what follows, $X \neq \{0\}$ will denote a Banach space and X^* its dual. For any $x \in X$ and $x^* \in X^*$ we denote by $\langle x^*, x\rangle$ the value of x^* at x. For $x \in X$ and $\varepsilon > 0$ we denote by $B_\varepsilon(x)$ the closed ball centered at x with radius $\varepsilon > 0$, while for $x,\ y \in X$ we denote by $[x, y]$ the closed segment $\{tx+(1-t)y : t \in [0,1]\}$. The segments $]x, y]$, $[x, y[$, and $]x, y[$ are defined analogously.

Throughout this article we shall deal with proper functions $f : X \to \mathbb{R} \cup \{+\infty\}$ (i.e., functions for which $dom(f) := \{x \in X : f(x) < +\infty\}$ is nonempty). For any $a \in \mathbb{R}$ the sublevel (resp., strict sublevel) set of f corresponding to a is the set $S_a(f) = \{x \in X : f(x) \leq a\}$ (resp., $S_a^<(f) = \{x \in X : f(x) < a\}$). We shall use S_a and $S_a^<$ if there is no risk of confusion.

The Fenchel–Moreau subdifferential $\partial^{FM} f(x)$ of f at any $x \in dom(f)$ is defined by the formula

$$\partial^{FM} f(x) := \{x^* \in X^* : f(y) \geq f(x) + \langle x^*, y - x\rangle \quad \forall y \in X\}. \tag{2.1}$$

(If $x \notin dom(f)$, then we set $\partial^{FM} f(x) = \emptyset$.)

Another useful subdifferential is the Greenberg–Pierskalla subdifferential $\partial^{GP} f$, given by

$$\partial^{GP} f(x) = \{x^* \in X^* : \langle x^*, y - x\rangle \geq 0 \Rightarrow f(y) \geq f(x)\}. \tag{2.2}$$

Given a set $C \subseteq X$ and $x \in X$, the normal cone to C at x is by definition the cone

$$N_C(x) = \{x^* \in X^* : \forall y \in C, \langle x^*, y - x\rangle \leq 0\}.$$

Let $N_f(x) := N_{S_{f(x)}}(x)$ (resp., $N_f^<(x) := N_{S^<_{f(x)}}(x)$) be the normal cone to the sublevel (resp., strict sublevel) set corresponding to the value $f(x)$. The following equivalencies are straightforward:

$$x^* \in N_f(x) \Longleftrightarrow (\forall y \in X, \langle x^*, y - x\rangle > 0 \Rightarrow f(y) > f(x)); \tag{2.3}$$

$$x^* \in N_f^<(x) \Longleftrightarrow (\forall y \in X, \langle x^*, y - x\rangle > 0 \Rightarrow f(y) \geq f(x)). \tag{2.4}$$

Combining the above relations it follows that

$$\partial^{GP} f(x) \subseteq N_f^<(x) \text{ and } N_f(x) \subseteq N_f^<(x).$$

Besides ∂^{FM} and ∂^{GP}, one can define other subdifferentials which, unlike the former ones, depend only on the local properties of the function f. Such is the Fréchet subdifferential $\partial^F f(x)$, defined by

$$\partial^F f(x) := \{x^* \in X^* : f(y) \geq f(x) + \langle x^*, y - x\rangle + o(y - x) \quad \forall y \in X\},$$

where $o : X \to R$ is some real valued function satisfying

$$\lim_{x \to 0} \frac{o(x)}{\|x\|} = 0.$$

Another "local" subdifferential is the upper Dini subdifferential $\partial^{D^+} f$, defined as follows:

$$\partial^{D^+} f(x) = \begin{cases} \{x^* \in X^* : \langle x^*, d\rangle \leq f^{D+}(x, d), \forall d \in X\} & \text{if } x \in dom(f), \\ \emptyset & \text{if } x \notin dom(f), \end{cases}$$

where

$$f^{D^+}(x, d) = \limsup_{t \searrow 0^+} \frac{1}{t}(f(x + td) - f(x)). \tag{2.5}$$

Both the upper Dini and the Fréchet subdifferential belong to a larger class of subdifferentials defined axiomatically. We recall from [2, Definition 2.1] the relevant definition.

DEFINITION 1. *A subdifferential ∂ is an operator that associates to any lower semicontinuous (lsc) function $f : X \to \mathbb{R} \cup \{+\infty\}$ and any $x \in X$ a subset $\partial f(x)$ of X^* so that the following properties are satisfied:*

(P1) $\partial f(x) = \partial^{FM} f(x)$, *whenever f is convex;*

(P2) $0 \in \partial f(x)$, *whenever f has a local minimum at x; and*

(P3) $\partial(f + g)(x) \subseteq \partial f(x) + \partial g(x)$

for all convex continuous functions g for which both $\partial g(x)$ and $\partial(-g)(x)$ are nonempty. (Such functions are called ∂-differentiable at x.)

Other subdifferentials satisfying the above properties are the Gâteaux, Hadamard, and Clarke–Rockafellar subdifferentials [2].

Remark 2. Let us observe, in relation to Property (P1), that

$$\partial^{FM} f \subseteq \partial f \tag{2.6}$$

for any lsc function f. Indeed, take any $x_0 \in X$ and any $x^* \in \partial^{FM} f(x_0)$. Then relation (2.1) guarantees that the function

$$g(x) = f(x) - \langle x^*, x - x_0 \rangle$$

has a minimum at x_0, which yields in view of (P2) that $0 \in \partial g(x_0)$. Using Properties (P3) and (P1) we now conclude

$$0 \in \partial f(x_0) + \partial(\langle -x^*, \cdot - x_0 \rangle) = \partial f(x_0) - x^*,$$

i.e., $x^* \in \partial f(x_0)$.

For the purposes of the present paper we shall always use a subdifferential ∂ such that $\partial \subseteq \partial^{D^+}$.

We further recall from [2, Definition 2.2] the following definition.

DEFINITION 3. *A norm $\|.\|$ on X is said to be ∂-smooth if the functions of the form $x \mapsto \sum_n \mu_n \|x - v_n\|^2$ are ∂-differentiable, where the sequence (v_n) converges in X, $\mu_n \geq 0$, and the series $\sum_n \mu_n$ is convergent.*

We shall always assume that the space X admits a ∂-smooth renorming. (Note that this condition is automatically satisfied if ∂ is the Clarke–Rockafellar subdifferential; also, all reflexive Banach spaces admit a ∂^F-smooth renorming.) In such a case, the following mean value theorem holds [2, Theorem 4.1].

THEOREM 4. *Let f be lsc and ∂ be a subdifferential. If $x, y \in X$ and $f(y) > f(x)$, then there exist $z \in [x, y[$ and sequences $(x_n) \subseteq dom(f)$, $(x_n^*) \subseteq X^*$, such that $x_n \to z$, $x_n^* \in \partial f(x_n)$, and*

$$\langle x_n^*, z + t(y - x) - x_n \rangle > 0 \quad \forall t > 0.$$

In particular, $dom(\partial f)$ is dense in $dom(f)$.

Subdifferentials can be used to characterize lsc quasiconvex functions. We recall that a function $f : X \to \mathbb{R} \cup \{+\infty\}$ is called *quasiconvex* if its sublevel sets S_α are convex subsets of X for all $\alpha \in \mathbb{R}$. In [1] it has been shown that a function f is quasiconvex if and only if the following property is true:

$$\text{if } x^* \in \partial f(x) \text{ and } \langle x^*, y - x \rangle > 0, \text{ then } f(z) \leq f(y) \quad \forall z \in [x, y]. \tag{2.7}$$

An easy consequence of (2.7) is the following property of lsc quasiconvex functions (for $\partial f \subseteq \partial^{D^+} f$):

$$\text{if } x^* \in \partial f(x) \text{ and } \langle x^*, y - x \rangle > 0, \text{ then } f(y) > f(x). \tag{2.8}$$

Indeed, $x^* \in \partial f(x)$ and $\langle x^*, y - x \rangle > 0$ yield $f^{D^+}(x, y - x) > 0$; hence for some $t > 0$ (suitably small) we have $f(x) < f(x + t(y - x))$. From (2.7) it follows that $f(x + t(y - x)) \leq f(y)$; hence the result.

Next let $T : X \rightrightarrows X^*$ be a multivalued operator. Following [5] we say that T is *cyclically quasimonotone* if for any $n \geq 1$ and any $x_1, x_2, \ldots, x_n \in X$ there exists $i \in \{1, 2, \ldots, n\}$ such that

$$\langle x_i^*, x_{i+1} - x_i \rangle \leq 0 \quad \forall x_i^* \in T(x_i) \tag{2.9}$$

(where $x_{n+1} := x_1$). If we restrict n in (2.9) to $n = 2$, then T is called *quasimonotone*.

3. The "quasiconvex" subdifferential ∂^q. In this section we introduce the "quasiconvex" subdifferential ∂^q whose definition depends on both local and global properties of the function. We show that this subdifferential seems completely adapted in quasiconvex analysis (as far as one considers that the Fenchel–Moreau subdifferential ∂^{FM} is apt in convex analysis). In subsection 3.1 we compare the subdifferential ∂^q with the one defined recently in [11], while in subsection 3.2 we present some interesting properties of ∂^q.

Given an abstract subdifferential ∂ (according to Definition 1) contained in ∂^{D^+}, we introduce below the "quasiconvex" subdifferential ∂^q.

DEFINITION 5. *The quasiconvex subdifferential $\partial^q f : X \rightrightarrows X^*$ of f is defined for all $x \in dom(f)$ as follows:*

$$\partial^q f(x) = \begin{cases} \partial f(x) \cap N_f(x) & \text{if } N_f^<(x) \neq \{0\}, \\ \emptyset & \text{if } N_f^<(x) = \{0\}. \end{cases}$$

If $x \notin dom f$, then we set $\partial^q f(x) = \emptyset$.

We present some fundamental properties of ∂^q in the following propositions.

PROPOSITION 6. *For every proper function f, the operator $\partial^q f$ is cyclically quasimonotone.*

Proof. It is sufficient to show that the operator N_f (relation (2.3)) is cyclically quasimonotone. The proof follows exactly the same pattern as the proof of quasimonotonicity of N_f in [12]. If $x_i \in X$, $i = 1, 2, \ldots, n$, and $x_i^* \in N_f(x_i)$ are such that $\langle x_i^*, x_{i+1} - x_i \rangle > 0$ for all i (where $x_{n+1} \equiv x_1$), then (2.8) implies that $f(x_{i+1}) > f(x_i)$ for all i. By transitivity we conclude $f(x_1) > f(x_1)$; hence we have a contradiction. □

PROPOSITION 7. *Let f be a radially continuous function (that is, the restriction of f on line segments is continuous). Then*

(i) *for all $x \in dom(f)$ we have*

$$\partial^q f(x) = \begin{cases} \partial f(x) \cap N_f(x) & \text{if } \partial^{GP} f(x) \neq \emptyset, \\ \emptyset & \text{if } \partial^{GP} f(x) = \emptyset. \end{cases}$$

In particular for any $x \in X$, if $\partial^q f(x) \neq \emptyset$, then $\partial^{GP} f(x) \neq \emptyset$.

(ii) $\partial^q f(x) \setminus \{0\} \subseteq \partial^{GP} f(x)$.

Proof. (i) If $0 \in \partial^{GP} f(x)$, then $\partial^{GP} f(x) = X^*$. Hence, if $\partial^{GP} f(x) \neq \emptyset$, then $N_f^<(x) \neq \{0\}$. So we have only to prove that if $\partial^{GP} f(x) = \emptyset$, then $N_f^<(x) = \{0\}$. Note that from (2.4) we always have $0 \in N_f^<(x)$. Let us show that $N_f^<(x) \setminus \{0\} \subseteq \partial^{GP} f(x)$. To this end, let $x^* \in N_f^<(x) \setminus \{0\}$ and suppose that $\langle x^*, y - x \rangle \geq 0$. Choose $d \in X$ such that $\langle x^*, d \rangle > 0$. For any $t > 0$ one has $\langle x^*, y + td - x \rangle > 0$; hence $f(y + td) \geq f(x)$. Letting $t \to 0$ and using radial continuity we get $f(y) \geq f(x)$, that is, $x^* \in \partial^{GP} f(x)$.

(ii) The second assertion follows from the following inclusions:

$$\partial^q f(x) \setminus \{0\} \subseteq N_f(x) \setminus \{0\} \subseteq N_f^<(x) \setminus \{0\} \subseteq \partial^{GP} f(x).$$

The proof is complete. □

PROPOSITION 8. *Suppose that f is lsc and satisfies one of the following conditions:*

(i) *f is convex;*

(ii) *f is quasiconvex and for all $a > \inf f$ the sublevel sets $S_a(f)$ have nonempty interior.*

Then

$$\partial f = \partial^q f.$$

Proof. It follows directly from Definition 5 that $\partial^q f \subseteq \partial f$. To show that equality holds, consider any $x^* \in \partial f(x)$. Suppose first that $x^* \neq 0$. Then (2.8) and (2.3) entail that $x^* \in N_f(x)$; hence $x^* \in \partial^q(x)$. If now $x^* = 0$, then obviously $x^* \in \partial f(x) \cap N_f(x)$. According to Definition 5 it suffices to ensure that $N_f^<(x) \neq \{0\}$. Indeed, if x is a global minimum, then $N_f^<(x) = X^*$. If x is not a global minimum, then f cannot be convex; hence assumption (ii) holds. It follows that the convex set $S^<_{f(x)}$ has a nonempty interior. Thus by the Hahn–Banach theorem there exists $y^* \in X^* \setminus \{0\}$ such that $\langle y^*, x\rangle \geq \langle y^*, x'\rangle$ for all $x' \in S^<_{f(x)}$. We now conclude that $y^* \in N_f^<(x)$, i.e., $N_f^<(x) \neq \{0\}$. □

Remark. The same proof shows that Proposition 8 (ii) holds without any assumption on the sublevel sets, in the case of X finite-dimensional.

Note that if f is lsc, quasiconvex, and radially continuous, then S_a has a nonempty interior for all $a > \inf f$. This is a direct consequence of the following proposition.

PROPOSITION 9. *If f is quasiconvex, lsc, and radially continuous, then it is continuous.*

Proof. Since f is lsc, it suffices to show that $S_a^<$ is open. For any $x \in S_a^<$, let b be such that $f(x) < b < a$. Since f is radially continuous, for any $y \in X$ we can find $\varepsilon > 0$ such that $]x - \varepsilon y, x + \varepsilon y[\subseteq S_b$. Hence $x \in \operatorname{alg\,int} S_b$. For closed convex sets in Banach spaces the algebraic and the topological interior coincide (e.g., [7, p. 139]). It follows that $x \in \operatorname{int} S_b \subseteq \operatorname{int} S_a^<$. Hence $S_a^<$ is open. □

The following lemma is in the same spirit.

LEMMA 10. *Let $K \subseteq X$ be closed. If $\operatorname{alg\,int} K \neq \emptyset$, then $\operatorname{int} K \neq \emptyset$.*

Proof. Let $x \in \operatorname{algint} K$. Then obviously

$$\bigcup_{n\in\mathbb{N}} n(K - x) = X.$$

By Baire's lemma, there exists $n_0 \in \mathbb{N}$ such that $\operatorname{int}(n_0(K - x)) \neq \emptyset$. We conclude that $\operatorname{int} K \neq \emptyset$. □

We are now ready to state the following result.

PROPOSITION 11. *Let f be lsc, and suppose that either f is radially continuous, or $\operatorname{dom}(f)$ is convex and S_a has nonempty interior for all $a > \inf f$.*

(i) *If the set $\{x \in X : N_f^<(x) \neq \{0\}\}$ is dense in $\operatorname{dom}(f)$, then f is quasiconvex.*

(ii) *f is quasiconvex if and only if the domain of $\partial^q f$ is dense in $\operatorname{dom}(f)$.*

Proof. (i) To show that f is quasiconvex, it suffices to show that S_a is convex for all a with $\inf f < a < +\infty$. For this it is sufficient to show that any $x \in X \setminus S_a$ can be strictly separated from S_a by means of a closed hyperplane. By Lemma 10, both assumptions imply that $\operatorname{int} S_a \neq \emptyset$. Choose any $y \in \operatorname{int} S_a$.

Case 1. Suppose that f is radially continuous. Then the restriction of f on the line segment $[x, y]$ takes all the values between $f(x)$ and $f(y)$. Hence there exists

$z \in]x, y[$ such that $a < f(z) < +\infty$. In particular, $z \in dom(f)$, so (by assumption) we can find $c^* \in N_f^<(c) \setminus \{0\}$, where c is as close to z as we wish. Since f is lsc we may assume that $f(c) > a$ and $c \in]x, y'[$ for some $y' \in \operatorname{int} S_a$. Using (2.4) we now obtain

$$\langle c^*, d\rangle > 0 \Rightarrow f(c+d) \geq f(c).$$

For all $w \in S_a$ we have $\langle c^*, w - c\rangle \leq 0$ (otherwise we would have $f(w) \geq f(c) > a$). In particular, $\langle c^*, w - c\rangle \leq 0$ for all $w \in y' + B_\varepsilon(y')$ for a suitable $\varepsilon > 0$. It follows easily that $\langle c^*, y' - c\rangle < 0$, hence $\langle c^*, x - c\rangle > 0$. Summarizing,

$$\langle c^*, w\rangle \leq \langle c^*, c\rangle < \langle c^*, x\rangle \quad \forall w \in S_a.$$

Consequently, c^* separates strictly S_a and x.

Case 2. Suppose that $dom(f)$ is convex. If $x \notin \overline{dom}(f)$, then we can strictly separate x and $\overline{dom}(f)$ by means of a closed hyperplane. In particular, the same hyperplane strictly separates x and S_a.

If $x \in \overline{dom}(f)$, then $[y, x[\subseteq \operatorname{int} dom(f)$. Since S_a is closed and $x \notin S_a$, there exists $z \in]y, x[$ such that $a < f(z) < +\infty$. As in Case 1, it now follows that x and S_a can be strictly separated.

(ii) If f is quasiconvex, then by Proposition 8 we conclude $\partial^q f = \partial f$. Hence (by Theorem 4) $dom(\partial^q f)$ is dense in $dom(f)$. Conversely, if $dom(\partial^q f)$ is dense in $dom(f)$, then the set $\{z \in dom(f) : N_f^<(z) \neq \{0\}\}$ is dense in $dom(f)$; hence by (i) the function f is quasiconvex. □

Combining Proposition 8, Proposition 11, and Theorem 4, we obtain the following corollary.

COROLLARY 12. *Let f be an lsc radially continuous function (respectively, f is an lsc function with convex domain and its sublevel sets have nonempty interior). Then the following are equivalent:*

(i) *f is quasiconvex;*

(ii) *$\partial^q f = \partial f$;*

(iii) *$\partial^q f$ satisfies the conclusion of Theorem 4 (mean value theorem);*

(iv) *$dom(\partial^q f)$ is dense in $dom(f)$.*

3.1. Comparison of ∂^q with other subdifferentials. We start with the following result.

PROPOSITION 13. *For any lsc function f,*

$$\partial^{FM} f \subseteq \partial^q f \subseteq \partial f. \tag{3.1}$$

Proof. The second inclusion follows directly from Definition 5. To prove the first inclusion, consider any $x^* \in \partial^{FM} f(x)$. It is straightforward from (2.3) that $x^* \in N_f(x) \subseteq N_f^<(x)$. Note also that $N_f^<(x) \neq \{0\}$ (if $x^* = 0$, then (2.1) implies that $N_f^<(x) = X^*$). Hence (3.1) follows from Remark 2. □

Remark 14. In view of Proposition 8, the inclusion $\partial^q f \subseteq \partial f$ becomes an equality if the function f is quasiconvex and continuous, while both inclusions in (3.1) become equalities if the function f is convex.

We shall further compare ∂^q with the subdifferential ∂^Q introduced recently in [11, Definition 2.1]. Before recalling the definition of the latter, we provide a result concerning the representation of lsc quasiconvex functions by means of *quasiaffine* functions. We recall that a function f is called quasiaffine if it is both quasiconvex

and quasiconcave. In contrast to the rest of the paper, in the next proposition we allow the functions to take the value $-\infty$.

PROPOSITION 15. *A function $f : X \to \mathbb{R}\cup\{+\infty,-\infty\}$ is lsc quasiconvex if and only if it satisfies*

$$f(x) = \sup_{q\in Q} q(x),$$

where Q is the set of continuous quasiaffine minorants $q : X \to \mathbb{R}\cup\{+\infty,-\infty\}$ that are differentiable on $q^{-1}(\mathbb{R})$.

Proof. The "if" part of the statement is obvious, since all continuous quasiaffine functions are lsc quasiconvex, and this class is closed under pointwise suprema. To prove the "only if" part, let $f : X \to \mathbb{R}\cup\{+\infty,-\infty\}$ be lsc quasiconvex and define $g : X \to \mathbb{R}\cup\{+\infty\}$ by $g(x) = e^{f(x)}$ (using the conventions $e^{+\infty} = +\infty$ and $e^{-\infty} = 0$). It follows that g is quasiconvex and nonnegative. Combining [8, Theorem 5.15] with implication (ii)⇒(i) in [8, Theorem 5.1], we conclude that g is the pointwise supremum of the collection of its real valued, differentiable, quasiaffine minorants with bounded derivatives. It follows that g is also the supremum of a collection of continuous nonnegative quasiaffine functions, which are differentiable at all points where their value is positive. Let us observe that $f(x) = \ln g(x)$ (with the conventions $\ln 0 = -\infty$ and $\ln +\infty = +\infty$) and that the logarithmic function

$$\ln : [0,+\infty] \to \mathbb{R}\cup\{+\infty,-\infty\}$$

is continuous, differentiable on $]0,+\infty[$, and increasing. The proposition follows from the observation that the composition $q = \ln\circ\, r$ of ln with a continuous quasiaffine function r which is differentiable at all points x such that $r(x) \in]0,+\infty[$ yields a continuous quasiaffine function q differentiable on $q^{-1}(\mathbb{R})$. □

Given an lsc function $f : X \to \mathbb{R}\cup\{+\infty\}$, let us recall the definition of the subdifferential $\partial^Q f$ given in [11], as follows. The subdifferential $\partial^Q f(x)$ of f at $x \in dom(f)$ is the set of all $x^* \in X^*$ such that for some nondecreasing differentiable function $\varphi : \mathbb{R} \to \mathbb{R}$ (depending on x^*), with $\varphi(0) = 0$ and $\varphi'(0) = 1$, the following relation holds:

$$f(y) \geq f(x) + \varphi(\langle x^*, y-x\rangle) \quad \forall y \in X. \tag{3.2}$$

Let us observe that the right-hand part of the above inequality defines a differentiable quasiaffine support function of f at x (i.e., a differentiable quasiaffine function g satisfying $f \geq g$ and $f(x) = g(x)$). Therefore $\partial^Q f(x)$ is contained in the set of the derivatives at x of the differentiable quasiaffine supports of f at x.

PROPOSITION 16. *Let $f : X \to \mathbb{R}\cup\{+\infty\}$ be lsc, and suppose that $\partial^F f \subseteq \partial f$.*

(i) *If x^* is the derivative of a continuous quasiaffine support of f at x differentiable at x, then $x^* \in \partial^q f(x)$.*

(ii) $\partial^Q f(x) \subseteq \partial^q f(x)$.

Proof. (i) From Theorem 2.31 of [8] it follows that a continuous function $h : X \to \mathbb{R}$ is quasiaffine if and only if there exist $y^* \in X^*$ and a nondecreasing continuous function $\psi : \mathbb{R} \to \mathbb{R}$ such that $h = \psi\circ y^*$. Thus if h is a quasiaffine support of f at x, and x^* is the derivative of h at x, then $x^* = \psi'(\langle y^*, x\rangle)y^*$. Since h is a support of f at x, we obviously have $x^* \in \partial^F f(x)$; thus $x^* \in \partial f(x)$.

Let us first assume that $x^* \neq 0$. Let $y \in X$ be such that $\langle x^*, y-x\rangle > 0$. Since $x^* \in \partial f(x)$ and h is quasiconvex, using (2.8) we conclude that $f(y) \geq h(y) >$

$h(x) = f(x)$. Thus $y \notin S^<_{f(x)}(f)$, which proves that $x^* \in N_f(x) \subseteq N^<_f(x)$. Hence $x^* \in \partial f(x) \cap N_f(x) = \partial^q f(x)$.

Suppose now that $x^* = 0$. Then obviously $x^* \in \partial f(x) \cap N_f(x)$; hence it suffices to show that $N^<_f(x) \neq \{0\}$. This certainly holds if x is a global minimum of f. If this is not the case, then $y^* \neq 0$. Let us prove that, in this case, $y^* \in N^<_f(x)$. Indeed, for $y \in S^<_{f(x)}(f)$ one has $\psi(\langle y^*, y\rangle) \leq f(y) < f(x) = \psi(\langle y^*, x\rangle)$, whence, as ψ is nondecreasing, $\langle y^*, y\rangle < \langle y^*, x\rangle$.

(ii) This portion of the proof follows directly from (i) and (3.2). □

3.2. Other properties of the subdifferential ∂^q. In this section we establish calculus rules for the quasiconvex subdifferential ∂^q. Let us first remark that inside the class of lsc quasiconvex functions whose sublevel sets have nonempty interior, the quasiconvex subdifferential ∂^q inherits calculus rules from the abstract subdifferential ∂; see Corollary 12. On the other hand, for any lsc function f, Definition 5 yields the following necessary condition for global optimality:

$$f \text{ has a global minimum at } x_0 \implies 0 \in \partial^q f(x_0). \tag{3.3}$$

Remark. Thanks to Proposition 8, relation (3.3) holds true also for local minima whenever f is lsc quasiconvex, and for all $a > \inf f$ the sublevel sets $S_a(f)$ have nonempty interior.

Let us further show a calculus rule based on the "supremum," an operation important in quasiconvex analysis.

PROPOSITION 17. *Suppose that ∂ is either the upper Dini subdifferential ∂^{D^+} or the Fréchet subdifferential ∂^F. Let $\{f_i\}_{i\in I}$ be a family of lsc functions on X, and set $f = \sup_{i\in I} f_i$. Then for every $x_0 \in X$*

$$\overline{\text{co}}^{w^*}\left(\bigcup_{i\in I(x_0)} \partial^q f_i(x_0)\right) \subseteq \partial^q f(x_0), \tag{3.4}$$

where $I(x_0) := \{i \in I : f_i(x_0) = f(x_0)\}$ and $\overline{\text{co}}^{w^*}(K)$ denotes the w-closed convex hull of K.*

Proof. Let $x_0 \in X$. If $x_0 \notin dom(f)$, then for all $i \in I(x_0)$, $f_i(x_0) = f(x_0) = +\infty$ and $\partial^q f(x_0) = \partial^q f_i(x_0) = \emptyset$. Hence we may suppose that $x_0 \in dom(f)$. Let us observe that $\partial^q f(x_0)$ is a w^*-closed and convex subset of X^*. Thus it suffices to show that if $x^* \in \bigcup_{i\in I(x_0)} \partial^q f_i(x_0)$, then $x^* \in \partial^q f(x_0)$. To do so, let $i \in I(x_0)$ and $x^* \in \partial^q f_i(x_0)$. Since $\partial^q f_i(x_0) \neq \emptyset$, we deduce that $N^<_{f_i}(x_0) \neq \{0\}$. Using the fact that $f(x_0) = f_i(x_0)$ and $f(x) \geq f_i(x)$ for all $x \in X$, we obtain $N^<_f(x_0) \neq \{0\}$. Thus it remains to show (see Definition 5) that $x^* \in \partial^{D^+} f(x_0) \cap N_f(x_0)$ (resp., $x^* \in \partial^F f(x_0) \cap N_f(x_0)$). But this follows easily from the fact that $N_{f_i}(x_0) \subset N_f(x_0)$ and $\partial^{D^+} f_i(x_0) \subset \partial^{D^+} f(x_0)$ (resp., $\partial^F f_i(x_0) \subset \partial^F f(x_0)$). □

Remark. (i) Relation (3.4) holds true whenever ∂ is an abstract subdifferential satisfying $\partial f(x_0) \subset \partial g(x_0)$, whenever $f(x_0) = g(x_0)$ and $f \leq g$.

(ii) Equality in (3.4) is generally not true, even if f is the supremum of two continuous quasiconvex functions. Indeed, let

$$f_1(x) = \begin{cases} \sqrt{-x} & \text{if } x \leq 0, \\ -\sqrt{x} & \text{if } x > 0, \end{cases}$$

and $f_2 = -f_1$. Then $f(x) = \max\{f_1(x), f_2(x)\} = \sqrt{|x|}$ and $\partial^q f(0) = \mathbb{R}$, while $\partial^q f_1(0) = \partial^q f_2(0) = \emptyset$.

Let us give a special case where (3.4) holds with equality. Suppose that $\{f_1, f_2, \dots, f_k\}$ is a finite family of locally Lipschitz quasiconvex functions on X that are regular (resp., strongly regular) at x_0; that is, $\partial^{D^+} f_i(x_0) = \partial^o f_i(x_0)$ (resp., $\partial^F f_i(x_0) = \partial^o f_i(x_0)$), where $\partial^o f_i(x_0)$ stands for the Clarke subdifferential of f_i at x_0 [3]. If $f = \max f_i$ and $x^* \in \partial^q f(x_0)$, then obviously $x^* \in \partial^o f(x_0)$; hence by [3, Proposition 2.3.12] $x^* \in \mathrm{co}(\bigcup_{i\in I(x_0)} \partial^o f_i(x_0))$. Thanks to Corollary 12(ii) and the regularity (resp., strong regularity) of each f_i, we infer that $\partial^o f_i(x_0) = \partial^q f_i(x_0)$, so equality in (3.4) follows.

A more general result is given in the following proposition.

PROPOSITION 18. *Let $f = \max_{i\in I} f_i$, where $\{f_i\}_{i\in I}$ is a finite set of lsc quasiconvex functions such that for all $a > \inf f_i$ the sublevel sets $S_a(f_i)$ have nonempty interior, and let $x_0 \in X$. Further, let ∂ be the upper Dini subdifferential, and assume that for all $i \in I$ and $d \in X$*

$$f_i^{D^+}(x_0, d) = \sup\{\langle x^*, d\rangle : x^* \in \partial f_i(x_0)\}. \tag{3.5}$$

(This condition is in particular satisfied whenever f is regular, or (Pshenichnyi) quasidifferentiable at x_0 with nonempty subdifferential.) Then

$$\overline{co}^{w^*}\left(\bigcup_{i\in I(x_0)} \partial^q f_i(x_0)\right) = \partial^q f(x_0), \tag{3.6}$$

where $I(x_0) := \{i \in I : f_i(x_0) = f(x_0)\}$.

Proof. Thanks to Proposition 17, we have only to show the right-hand side inclusion "$\supseteq$". Let us suppose, in seeking a contradiction, that there exists

$$x^* \in \partial^q f(x_0) \setminus \overline{co}^{w^*}\left(\bigcup_{i\in I(x_0)} \partial^q f_i(x_0)\right).$$

Then by the Hahn–Banach theorem there exist $d \in X$ and $\varepsilon > 0$ such that for all $z^* \in \overline{co}^{w^*}(\bigcup_{i\in I(x_0)} \partial^q f_i(x_0))$ we have $\langle x^*, d\rangle > \langle z^*, d\rangle + \varepsilon$. Since I is finite, it can be easily shown that there exists $i \in I$ such that $f^{D^+}(x_0, d) \le f_i^{D^+}(x_0, d)$. Our assumptions imply (see Proposition 8(ii)) that $\partial f_i(x_0) = \partial^q f_i(x_0)$. Since $\partial^q f(x_0) \subseteq \partial f(x_0)$, we get $x^* \in \partial f(x_0)$; that is,

$$f_i^{D^+}(x_0, d) \ge f^{D^+}(x_0, d) \ge \langle x^*, d\rangle > \langle z^*, d\rangle + \varepsilon \quad \forall z^* \in \partial f_i(x_0).$$

This clearly contradicts (3.5). ∎

Note that whenever X is finite-dimensional, the assumption on the sublevel sets is superfluous (see the remark after Proposition 8). The following example shows that the assumption that the family is finite cannot be overcome, even if all f_i are convex and the supremum is actually a maximum at each point.

Example. Let $f : \mathbb{R} \to \mathbb{R}$ be the convex function

$$f(x) = \begin{cases} 0 & \text{if } x \le 0, \\ x + x^2 & \text{if } 0 < x. \end{cases}$$

For each $n \in \mathbb{N}$, let $g_n(x)$ be the equation of the straight line which is tangent to the graph of f at $(1/n, f(1/n))$, and let $x_n \in]0, 1/n[$ be the intersection of this tangent with the x-axis. Let us define

$$f_n(x) = \begin{cases} 0 & \text{if } x \leq x_n, \\ g_n(x) & \text{if } x_n < x \leq \frac{1}{n}, \\ f(x) & \text{if } \frac{1}{n} < x. \end{cases}$$

Then f_n is convex, $f(x) = \max_{n \geq 1} f(x)$ for each $x \in \mathbb{R}$, and $\partial^q f_n(0) = \{0\}$ while $\partial^q f(0) = [0, 1]$. Hence (3.6) does not hold.

In what follows, we shall show that ∂^q obeys a chain rule. We start with the corresponding rule for classical subdifferentials.

PROPOSITION 19. *Suppose that ∂ is either ∂^{D^+} or ∂^F, let $f : X \to \mathbb{R} \cup \{+\infty\}$, and suppose that $g : \mathbb{R} \cup \{+\infty\} \to \mathbb{R} \cup \{+\infty\}$ is nondecreasing.*

(i) *If g is differentiable at $f(x_0)$ for some $x_0 \in dom(f)$, then*

$$g'(f(x_0))\, \partial f(x_0) \subseteq \partial (g \circ f)(x_0). \tag{3.7}$$

(ii) *If, moreover, f is convex and $g'(f(x_0)) > 0$, then* (3.7) *holds with equality.*

Proof. (i) Assume first that $\partial = \partial^{D^+}$. Let $a < f^{D^+}(x_0, d)$. It follows from (2.5) that for any $\delta > 0$ there exists $0 < t < \delta$ satisfying

$$\frac{f(x_0 + td) - f(x_0)}{t} > a.$$

Hence $f(x_0 + td) > f(x_0) + at$ and $g(f(x_0 + td)) \geq g(f(x_0) + at)$. Since g is differentiable at $f(x_0)$ it follows that

$$g(f(x_0) + at) = g(f(x_0)) + g'(f(x_0))\, at + o(at),$$

where $\lim_{t \to 0} \frac{o(t)}{t} = 0$. Hence

$$\frac{g(f(x_0 + td)) - g(f(x_0))}{t} \geq a g'(f(x_0)) + \frac{o(at)}{t},$$

which yields $(g \circ f)^{D^+}(x_0, d) \geq a g'(f(x_0))$. Consequently,

$$g'(f(x_0))\, f^{D^+}(x_0, d) \leq (g \circ f)^{D^+}(x_0, d);$$

hence (3.7) holds.

Assume now that $\partial = \partial^F$ and take any $x^* \in \partial^F f(x_0)$. Then

$$\liminf_{\|u\| \searrow 0} \frac{f(x_0 + u) - f(x_0) - \langle x^*, u \rangle}{\|u\|} \geq 0.$$

Let $a < 0$. Then there exists $\delta > 0$ such that for all $u \in X$ with $\|u\| < \delta$

$$\frac{f(x_0 + u) - f(x_0) - \langle x^*, u \rangle}{\|u\|} > a.$$

Since g is nondecreasing, the previous inequality implies

$$g(f(x_0 + u)) \geq g(f(x_0) + \langle x^*, u \rangle + a\|u\|),$$

and since g is differentiable at $f(x_0)$,

$$g(f(x_0+u)) \geq g(f(x_0)) + g'(f(x_0))(\langle x^*, u\rangle + a\|u\|) + o(\langle x^*, u\rangle + a\|u\|),$$

where $\lim_{t\to 0}\frac{o(t)}{t} = 0$. Since $(\|x^*\| - a)\|u\| \geq |\langle x^*, u\rangle + a\|u\||$, it follows that

$$\liminf_{\|u\|\searrow 0} \frac{(g\circ f)(x_0+u) - (g\circ f)(x_0) - g'(f(x_0))\langle x^*, u\rangle}{\|u\|} \geq a g'(f(x_0)). \tag{3.8}$$

Since the above relation is true for all $a < 0$, the left-hand side is nonnegative. This implies that $g'(f(x_0))\ x^* \in \partial^F(g\circ f)(x_0)$; hence (3.7) holds.

(ii) Suppose now that f is convex. Then the function $t \to f(x_0+td)$ is right differentiable; hence the same holds also for the function $t \to (g\circ f)(x_0+td)$. It follows from the usual chain rule for differentiable functions that

$$g'(f(x_0))\, f^{D^+}(x_0, d) = (g\circ f)^{D^+}(x_0, d). \tag{3.9}$$

Hence if $\partial = \partial^{D^+}$, then (3.7) holds with equality.

Suppose now that $\partial = \partial^F$. It is sufficient to show that if $x^* \notin \partial^F f(x_0)$, then $g'(f(x_0))x^* \notin \partial^F(g\circ f)(x_0)$. Since f is convex we have $\partial^F f = \partial^{FM} f$; hence from (2.1) there exists $u \in X$ such that $f(x_0+u) - f(x_0) < \langle x^*, u\rangle$. Choose $a < 0$ such that

$$f(x_0+u) - f(x_0) < \langle x^*, u\rangle + a\|u\|. \tag{3.10}$$

Convexity of f guarantees that the function $t \to \frac{f(x_0+tu)-f(x_0)}{t}$ is nondecreasing for all $t \geq 0$. Thus for any $0 < t < 1$ we infer from (3.10) that

$$f(x_0+tu) - f(x_0) < (\langle x^*, u\rangle + a\|u\|)\, t.$$

Since g is nondecreasing we obtain

$$g(f(x_0+tu)) \leq g(f(x_0) + t\langle x^*, u\rangle + ta\|u\|),$$

and, since g is differentiable at $f(x_0)$,

$$g(f(x_0+tu)) \leq g(f(x_0)) + tg'(f(x_0))(\langle x^*, u\rangle + a\|u\|) + o(t\langle x^*, u\rangle + ta\|u\|),$$

where $\lim_{t\to 0}\frac{o(t)}{t} = 0$. Dividing by $t\|u\|$ and letting $t \to 0$ we deduce

$$\liminf_{t\searrow 0} \frac{(g\circ f)(x_0+tu) - (g\circ f)(x_0) - g'(f(x_0))\langle x^*, tu\rangle}{\|tu\|} \leq a g'(f(x_0)).$$

Since $a < 0$ and $g'(f(x_0)) > 0$, it follows that the left-hand side of (3.8) is negative. Hence $g'(f(x_0))\, x^* \notin \partial^F(g\circ f)(x_0)$. □

PROPOSITION 20. *Let $f : X \to \mathbb{R}\cup\{+\infty\}$ be lsc and $g : \mathbb{R}\cup\{+\infty\} \to \mathbb{R}\cup\{+\infty\}$ be nondecreasing. Assume that the subdifferential ∂ satisfies assertions* (i) *and* (ii) *of Proposition* 19 *(for instance, $\partial = \partial^F$ or ∂^{D^+}). If g is differentiable at $f(x_0)$ with $g'(f(x_0)) > 0$ for some $x_0 \in dom(f)$, then*

$$g'(f(x_0))\, \partial^q f(x_0) \subseteq \partial^q(g\circ f)(x_0); \tag{3.11}$$

the above inclusion becomes an equality whenever f is convex.

Proof. Since g is nondecreasing and $g'(f(x_0)) > 0$, we can easily deduce that

$$N_f^<(x_0) = N_{g\circ f}^<(x_0) \tag{3.12}$$

and

$$N_f(x_0) = N_{g\circ f}(x_0). \tag{3.13}$$

Thus, if $x^* \in \partial^q f(x_0)$, then (3.12) yields $N_{g\circ f}^<(x_0) \neq \emptyset$. Since $\partial^q f \subseteq \partial f$, we infer from (3.7) that

$$g'(f(x_0))\, x^* \in \partial(g\circ f)(x_0).$$

Besides, since $x^* \in N_f(x_0)$ and $N_{g\circ f}(x_0)$ is a cone, (3.13) implies

$$g'(f(x_0))\, x^* \in N_{g\circ f}(x_0).$$

Hence (3.11) holds.

If now f is convex, then, by Proposition 8, $\partial^q f = \partial^{FM} f = \partial f$. Hence, in order to show the equality in (3.11), we have to show that $\partial^q(g\circ f)(x_0) = \partial(g\circ f)(x_0)$. It suffices to show that if $x^* \in \partial(g\circ f)(x_0)$, then $x^* \in \partial^q(g\circ f)(x_0)$. Since (3.7) holds with equality, we have

$$\frac{x^*}{g'(f(x_0))} \in \partial f(x_0) = \partial^q f(x_0).$$

Hence $N_{g\circ f}^<(x_0) = N_f^<(x_0) \neq \{0\}$ and (since $N_f(x_0)$ is a cone) $x^* \in N_f(x_0) = N_{g\circ f}(x_0)$. It follows that $x^* \in \partial^q(g\circ f)(x_0)$. □

Let $C \subseteq X$ and let us define the (upper Dini tangent) cone $T_{D^+}(C, x_0)$ of C at $x_0 \in C$ as follows:

$$T_{D^+}(C, x_0) = \{u \in X : \exists \delta > 0\colon\ \forall t \in]0, \delta[,\ x_0 + tu \in C\}.$$

We have the following proposition.

PROPOSITION 21. *Let $f : X \to \mathbb{R}\cup\{+\infty\}$ and $x_0 \in f^{-1}(\mathbb{R})$. Then*

$$\begin{aligned}&\{x^* \in X^* : (x^*, -1) \in N_{\operatorname{epi} f}(x_0, f(x_0))\} \subseteq \partial^q f(x_0)\\ &\subseteq \{x^* \in X^* : (x^*, -1) \in (T_{D^+}(\operatorname{epi} f, (x_0, f(x_0))))^o\}.\end{aligned}$$

Proof. The first inclusion follows from (3.1) and the observation that

$$\partial^{FM} f(x_0) = \{x^* \in X^* : (x^*, -1) \in N_{\operatorname{epi} f}(x_0, f(x_0))\}.$$

To prove the second inclusion, since $\partial^q \subseteq \partial \subseteq \partial^{D^+}$ it suffices to show that

$$\partial^{D^+} f(x_0) = \{x^* \in X^* : (x^*, -1) \in (T_{D^+}(\operatorname{epi} f, (x_0, f(x_0))))^o\}.$$

To this end, let $x^* \in \partial^{D^+} f(x_0)$. For any $(u, v) \in T_{D^+}(\operatorname{epi} f, (x_0, f(x_0)))$ there exists $\delta > 0$ such that

$$f(x_0 + tu) \leq f(x_0) + tv$$

for all $t \in]0, \delta[$. It follows that

$$\langle x^*, u\rangle \leq \limsup_{t\searrow 0} \frac{f(x_0 + tu) - f(x_0)}{t} \leq v,$$

i.e., $(x^*,-1)\in (T_{D^+}(\operatorname{epi} f,(x_0,f(x_0))))^o$.

Conversely, let $x^*\in X^*$ be such that $(x^*,-1)\in (T_{D^+}(\operatorname{epi} f,(x_0,f(x_0))))^o$. For each $u\in X$, set $v=f^{D^+}(x_0,u)$. Then for any $\lambda\in]v,+\infty[$ we can find $\delta>0$ such that for all $t\in]0,\delta[$

$$\frac{f(x_0+tu)-f(x_0)}{t}\leq\lambda.$$

It follows that $(u,\lambda)\in T_{D^+}(\operatorname{epi} f,(x_0,f(x_0)))$, and hence $\langle x^*,u\rangle\leq\lambda$. Since this is true for all $\lambda\in]v,+\infty[$, we deduce that $\langle x^*,u\rangle\leq v$; hence $x^*\in\partial^{D^+}f(x_0)$. □

Let us finally state the following corollary.

COROLLARY 22. *Let $A\subseteq X$ and denote by $\delta_A:X\to\mathbb{R}\cup\{+\infty\}$ the indicator function of A defined by*

$$\delta_A(x)=\begin{cases}0 & \text{if } x\in A,\\ +\infty & \text{if } x\notin A.\end{cases}$$

For all $x_0\in A$ we have

$$\partial^q\delta_A(x_0)=N_A(x_0).$$

Proof. We have the following equivalencies:

$$\begin{aligned}x^*\in\partial^{FM}\delta_A(x_0)&\Leftrightarrow\forall x\in X,\ \langle x^*,x-x_0\rangle\leq\delta_A(x)-\delta_A(x_0)\\&\Leftrightarrow\forall x\in A,\ \langle x^*,x-x_0\rangle\leq 0\Leftrightarrow x^*\in N_A(x_0).\end{aligned}$$

Hence (3.1) implies that $N_A(x_0)\subseteq\partial^q\delta_A(x_0)$. Conversely, if $x^*\in\partial^q\delta_A(x_0)$, then $x^*\in N_{\delta_A}(x_0)$. It is very easy to see that $N_{\delta_A}(x_0)=N_A(x_0)$, and the corollary follows. □

REFERENCES

[1] D. AUSSEL, *Subdifferential properties of quasiconvex and pseudoconvex functions: Unified approach*, J. Optim. Theory Appl., 97 (1998), pp. 29–45.

[2] D. AUSSEL, J. N. CORVELLEC, AND M. LASSONDE, *Mean value property and subdifferential criteria for lower semicontinuous functions*, Trans. Amer. Math. Soc., 347 (1995), pp. 4147–4161.

[3] F. H. CLARKE, *Optimization and Nonsmooth Analysis*, Wiley Interscience, New York, 1983.

[4] J. P. CROUZEIX, *Contributions a l'Étude des Fonctions Quasiconvexes*, Ph.D. thesis, Université de Clermont-Ferrand II, Aubière cedex, France, 1977.

[5] A. DANIILIDIS AND N. HADJISAVVAS, *On generalized cyclically monotone operators and proper quasimonotonicity*, Optimization, 47 (2000), pp. 123–135.

[6] H. P. GREENBERG AND W. P. PIERSKALLA, *Quasi-conjugate functions and surrogate duality*, Cahiers Centre Études Recherche Opér., 15 (1973), pp. 437–448.

[7] R. HOLMES, *Geometric Functional Analysis and Its Applications*, Springer, New York, 1975.

[8] J. E. MARTÍNEZ-LEGAZ, *Quasiconvex duality theory by generalized conjugation methods*, Optimization, 19 (1988), pp. 603–652.

[9] J. E. MARTÍNEZ-LEGAZ, *Weak lower subdifferentials and applications*, Optimization, 21 (1990), pp. 321–341.

[10] J. E. MARTÍNEZ-LEGAZ AND S. ROMANO-RODRÍGUEZ, *α-lower subdifferentiable functions*, SIAM J. Optim., 3 (1993), pp. 800–825.

[11] J. E. MARTÍNEZ-LEGAZ AND P. H. SACH, *A new subdifferential in quasiconvex analysis*, J. Convex Anal., 6 (1999), pp. 1–12.

[12] J. P. PENOT, *Are generalized derivatives useful for generalized convex functions?*, in Generalized Convexity, Generalized Monotonicity, J.-P. Crouzeix, J.-E. Martínez-Legaz, and M. Volle, eds., Kluwer, Dordrecht, The Netherlands, 1998, pp. 3–59.

[13] F. PLASTRIA, *Lower subdifferentiable functions and their minimization by cutting planes*, J. Optim. Theory Appl., 46 (1985), pp. 37–53.

SIAM J. OPTIM.
Vol. 12, No. 2, pp. 421–435

SUFFICIENT CONDITIONS FOR ERROR BOUNDS*

ZILI WU† AND JANE J. YE†

Abstract. For a lower semicontinuous (l.s.c.) inequality system on a Banach space, it is shown that error bounds hold, provided every element in an abstract subdifferential of the constraint function at each point outside the solution set is norm bounded away from zero. A sufficient condition for a global error bound to exist is also given for an l.s.c. inequality system on a real normed linear space. It turns out that a global error bound closely relates to metric regularity, which is useful for presenting sufficient conditions for an l.s.c. system to be regular at sets. Under the generalized Slater condition, a continuous convex system on R^n is proved to be metrically regular at bounded sets.

AMS subject classifications. 49J52, 90C26, 90C31

Key words. abstract subdifferentials, inequality systems, error bounds, metrical regularity, generalized Slater condition

PII. S1052623400371557

1. Introduction.

Let X be a real normed linear space and C a nonempty closed subset of X. Let $f_i, |g_j| : X \to (-\infty, +\infty]$ be lower semicontinuous (l.s.c.) for each $i = 1, \ldots, r$ and $j = 1, \ldots, s$. Denote the solution set of an l.s.c. (inequality) system by

$$S := \{x \in C : f_1(x) \le 0, \ldots, f_r(x) \le 0; g_1(x) = 0, \ldots, g_s(x) = 0\},$$

which is assumed to be nonempty. *The distance function* $d_S : X \to R$ is defined by

$$d_S(x) = \inf\{\|x - s\| : s \in S\}.$$

The set S is said to have a *global error bound* if there exists a constant $\mu > 0$ such that

$$d_S(x) \le \mu(\, \|F(x)_+\| + \|G(x)\|\,) \quad \forall x \in C,$$

where $F(x)_+ = (f_1(x)_+, \ldots, f_r(x)_+) \in R^r$ with $f_i(x)_+ := \max\{f_i(x), 0\}$ for $i = 1, \ldots, r$, $G(x) = (g_1(x), \ldots, g_s(x)) \in R^s$ and $\|\cdot\|$ is the usual Euclidean norm. The set S is said to have a *local error bound* if there exist constants $\mu > 0$ and $\delta > 0$ such that

$$d_S(x) \le \mu(\, \|F(x)_+\| + \|G(x)\|\,) \quad \forall x \in C \text{ with } \|(\, F(x)_+, G(x)\,)\| < \delta.$$

Apparently if the set S has a global (local) error bound, then functions involved provide a global (local) error estimate for the distance from any point x to the solution set S. Because this kind of estimation has many important applications in optimization, many sufficient conditions for error bounds to exist have been given since Hoffman [10] proved that a global error bound always holds for any linear inequality systems on R^n. The reader is referred to [1, 2, 4, 5, 6, 9, 11, 13, 14, 15, 16, 17, 18, 21, 22, 23] and the references therein for the results on error bounds.

*Received by the editors April 25, 2000; accepted for publication (in revised form) April 12, 2001; published electronically December 14, 2001.

http://www.siam.org/journals/siopt/12-2/37155.html

†Department of Mathematics and Statistics, University of Victoria, Victoria, BC, Canada V8W 3P4 (ziliwu@joymail.com, janeye@math.uvic.ca). The work of the second author was supported by NSERC and a University of Victoria internal research grant.

It is worth pointing out that there are two important classes of conditions in these results. One contains the Slater condition (explicitly or implicitly), which closely relates to the points inside the solution set S of a system, while the other is expressed by various subdifferentials of functions at the points outside S. Since the latter includes subdifferentials, one can use the knowledge of nonsmooth analysis to study this issue more effectively.

To the authors' knowledge, it is Ioffe [11] who first used Ekeland's variational principle and the sum rule to prove the existence of a global error bound (as well as metric regularity at a point) for a Lipschitz continuous equality system under the condition that any element in the Clarke subdifferential of the constraint function at each point outside the solution set be norm bounded away from zero. Using Ioffe's method, Ye [22] and Jourani [13] have sharpened the result of Ioffe by replacing the Clarke subdifferential with the limiting subdifferential in R^n and a partial subdifferential in a general Banach space, respectively. In a Hilbert space, Clarke et al. [4, Theorem 3.3.1] have weakened Ioffe's condition using the proximal subdifferential instead of the Clarke subdifferential (see also Ye [23, Claim]). Since the proximal subdifferential does not satisfy the sum rule, the result of Clarke et al. was proved not by Ioffe's method but by the decrease principle. We note that Wu [21] has used a different tool, that is, the fuzzy sum rule (instead of the sum rule), to prove that the Clarke subdifferential in Ioffe's condition can be replaced by the proximal subdifferential for an l.s.c. system on R^n. This method is in fact suitable for various subdifferentials on a Banach space no matter whether they satisfy the fuzzy sum rule or the sum rule since the latter always implies the former. Therefore in this paper, we introduce an abstract subdifferential which satisfies the fuzzy sum rule and then take advantage of this method to show that the Clarke subdifferential in Ioffe's condition can really be replaced by such an abstract subdifferential which includes many subdifferentials in the nonsmooth analysis literature. These results unify and extend those stated in this paragraph. In fact, for an l.s.c. system they have provided sufficient conditions not only for a global error bound but also for a local error bound as well as for metric regularity.

This paper is organized as follows. In section 2, we introduce the concept of ∂_ω-subdifferential and show that several common subdifferentials in nonsmooth analysis are ∂_ω-subdifferentials. In section 3, we use the ∂_ω-subdifferential to present sufficient conditions for error bounds to exist for l.s.c. inequality systems on Banach spaces. Section 4 is devoted to a sufficient condition for a global error bound to hold for a general inequality system on a real normed linear space. With this result we extend those of Deng [5, 6] to an l.s.c. convex system on a real normed linear space. In section 5, relations between error bound and metrical regularity are revealed, and some sufficient conditions are given for a continuous convex system to be metrically regular at a nonempty set. In particular, we prove that a generalized Slater condition is sufficient for a continuous convex system to be metrically regular at any bounded sets in R^n.

Throughout this paper, B and $\overline{B}$, respectively, denote the open unit ball and its closure of X, while B^* and $\overline{B^*}$ are, respectively, the open unit ball and its closure of the dual space X^*. For a nonempty closed subset C of X, ψ_C and $N_C(x)$ denote the indicator function of C and the (Clarke) normal cone to C at $x \in C$, respectively. For an extended real-valued function f defined on X, its epigraph is written as

$$epi\, f := \{(x, r) \in dom\, f \times R : f(x) \le r\},$$

where $dom\, f := \{x \in X : f(x) < +\infty\}$.

2. ∂_ω-subdifferentials. Here we introduce the concept of ∂_ω-subdifferentials for l.s.c. functions, which unifies that of several common subdifferentials in the nonsmooth analysis literature.

DEFINITION 2.1. *Let X be a Banach space and $f : X \to (-\infty, +\infty]$ be l.s.c. at $x \in dom\, f$. A subset of X^*, denoted by $\partial_\omega f(x)$, is called a ∂_ω-*subdifferential of f at x if it has the following properties:

(ω_1) $\partial_\omega g(x) = \partial_\omega f(x)$ *if $g = f$ in a neighborhood of x.*

(ω_2) $0 \in \partial_\omega f(x)$ *when f attains a local minimum at x.*

(ω_3) *If f is convex and Lipschitz of rank L near x, then $\partial_\omega f(x) \subseteq L\overline{B^*}$.*

(ω_4) *The fuzzy sum rule holds; that is, if $g : X \to (-\infty, +\infty]$ is Lipschitz near x, then for any $\xi \in \partial_\omega(f+g)(x)$ and any $\delta > 0$, there exist $x_1, x_2 \in x + \delta B$ such that*

$$f(x_1) \in f(x) + \delta B_1,\ g(x_2) \in g(x) + \delta B_1,\ \text{and } \xi \in \partial_\omega f(x_1) + \partial_\omega g(x_2) + \delta B^*,$$

where $B_1 = (-1, 1)$.

The following commonly used subdifferentials turn out to be ∂_ω-subdifferentials.

Example 2.1. Let X be a Banach space and $f : X \to (-\infty, +\infty]$ be l.s.c. at $x \in dom\, f$. The *Clarke–Rockafellar generalized derivative* of f at x in the direction $v \in X$ is defined as follows:

$$f^\circ(x; v) := \lim_{\epsilon \to 0^+} \limsup_{\substack{y \xrightarrow{f} x \\ t \to 0^+}} \inf_{w \in v + \epsilon B} \frac{f(y + tw) - f(y)}{t},$$

where $y \xrightarrow{f} x$ signifies that y and $f(y)$ converge to x and $f(x)$, respectively. The *generalized gradient* of f at x is the subset of X^* given by

$$\partial f(x) = \begin{cases} \{\xi \in X^* : f^\circ(x; v) \geq \langle \xi, v \rangle \quad \forall v \in X\} & \text{if } f^\circ(x; 0) \neq -\infty; \\ \emptyset & \text{if } f^\circ(x; 0) = -\infty. \end{cases}$$

By the above definition (see Clarke [3, Proposition 2.1.2 (a) and Corollary 1 of Theorem 2.9.8]), $\partial f(x)$ satisfies properties (ω_1)–(ω_4).

Example 2.2. Let X be an Asplund space, i.e., a Banach space such that every continuous convex function is Fréchet differentiable at each point of some G_δ dense subset of this space (which includes all reflexive Banach spaces). Let $f : X \to (-\infty, +\infty]$ be l.s.c. at $x \in dom\, f$. The *Fréchet subdifferential* of f at x, denoted by $\partial_F f(x)$, is the set

$$\left\{ \xi \in X^* : \liminf_{\|h\| \to 0} \frac{f(x + h) - f(x) - \langle \xi, h \rangle}{\|h\|} \geq 0 \right\}.$$

Based on the definition, Ioffe [12, Proposition 1], and Fabian [8, Theorem 3], $\partial_F f(x)$ is a ∂_ω-subdifferential of f at x.

Example 2.3. Let H be a Hilbert space and $f : H \to (-\infty, +\infty]$ be l.s.c. at $x \in dom\, f$. A vector $\xi \in H^*$ is called a *proximal subgradient* of f at x provided that there exist positive numbers M and δ such that

$$f(y) \geq f(x) + \langle \xi, y - x \rangle - M\|y - x\|^2 \quad \forall y \in x + \delta B.$$

The set of all such ξ is denoted by $\partial^\pi f(x)$ and is referred to as the *proximal subdifferential* of f at x. It follows that $\partial^\pi f(x)$ satisfies properties (ω_1)–(ω_4) from the above inequality and Clarke et al. [4, Theorems 1.7.3 and 1.8.3].

Remark 2.1. (i) For a convex function, all subdifferentials in Examples 2.1–2.3 coincide with the subdifferentials in the sense of convex analysis.

(ii) The Fréchet subdifferential contains only the Fréchet derivative whenever a function is Fréchet differentiable, and the proximal subdifferential includes only the Fréchet derivative when a function is Fréchet differentiable and its Fréchet derivative is locally Lipschitz continuous.

(iii) In an Asplund space, one has

$$\partial_F f(x) \subseteq \partial f(x),$$

while in a Hilbert space, the following inclusions hold:

$$\partial^\pi f(x) \subseteq \partial_F f(x) \subseteq \partial f(x).$$

3. Sufficient conditions for an l.s.c. system. Consider a simple inequality system

$$f(x) \leq 0,$$

where f is a locally Lipschitz function defined on R. If the solution set $S := \{x \in R : f(x) \leq 0\}$ is nonempty, then the inequality $d_S(x) \leq \mu f(x)_+$ holds automatically for any $x \in S$ and any $\mu > 0$. To look for a sufficient condition for this inequality to hold for some $\mu > 0$ and any point $x \in R\backslash S$, we can take one point $x_0 \in S$ such that $f(x_0) = 0$ and $f(y) > 0$ for any $y \in (x_0, x] = \{tx_0 + (1-t)x : t \in [0,1)\}$. By the Lebourg mean-value theorem [3, Theorem 2.3.7], there exist $z \in (x_0, x]$ and $\xi \in \partial f(z)$ such that

$$f(x) - f(x_0) = \xi \cdot (x - x_0),$$

from which $f(x)_+ = \|\xi\| \cdot \|x - x_0\| \geq \|\xi\| d_S(x)$. Therefore if $\|\xi\| \geq \mu^{-1}$ holds for some $\mu > 0$ and any $\xi \in \partial f(x)$ for each $x \in R\backslash S$, then $d_S(x) \leq \mu f(x)_+$ holds for any $x \in R$.

For an l.s.c. function f defined on a Banach space X, will the existence of a positive constant μ such that

$$\|\xi\| \geq \mu^{-1} \quad \forall \xi \in \partial_\omega f(x) \qquad \forall x \in X\backslash S$$

also imply the existence of a global error bound? The following theorem gives an affirmative answer.

THEOREM 3.1. *Let $f : X \to (-\infty, +\infty]$ be an l.s.c. function on a Banach space X. Suppose that $x_0 \in S := \{x \in X : f(x) \leq 0\}$ and there exist $\mu > 0$ and $0 < \epsilon \leq \infty$ such that*

$$\|\xi\|_* \geq \mu^{-1} \quad \forall \xi \in \partial_\omega f(x)$$

for any x with $0 < f(x) < \epsilon$ (or $\|x - x_0\| < \epsilon$ and $0 < f(x) < +\infty$). Then we have

$$d_S(x) \leq \mu f(x)_+ \quad \forall x \in X \text{ with } f(x) < \epsilon/2 \text{ (or } \|x - x_0\| < \epsilon/2).$$

Proof. We need only to prove the conclusion for the case where $0 < \epsilon < +\infty$, since for the case $\epsilon = +\infty$ we can obtain the corresponding result by taking the limit from the former one.

Suppose that there were u such that $f(u) < \epsilon/2$ (or $u \in x_0 + (\epsilon/2)B$) and

$$d_S(u) > \mu f(u)_+.$$

Then $u \notin S$ and hence $0 < f(u) < +\infty$. Besides, we can choose $\alpha > 0$ and $t > 1$ such that

$$f(u) \le \frac{\epsilon}{2+\alpha} < \frac{\epsilon}{2} \left(\text{or } \|u - x_0\| \le \frac{\epsilon}{2+\alpha} < \frac{\epsilon}{2}\right) \text{ and } d_S(u) > t\mu f(u) := \gamma. \tag{3.1}$$

Thus $f(u)_+ = f(u) = \gamma(t\mu)^{-1}$ and hence

$$f(u)_+ \le \inf_{v \in X} f(v)_+ + \gamma(t\mu)^{-1}.$$

Note that the function $f(\cdot)_+$ is l.s.c. and bounded below. Applying Ekeland's variational principle [7] to $f(\cdot)_+$ with $\sigma = \gamma(t\mu)^{-1}$ and $\lambda = \gamma$, we find $x \in X$ satisfying

$$\|x - u\| \le \gamma, \tag{3.2}$$

$$f(v)_+ + (t\mu)^{-1}h(v) \ge f(x)_+ \quad \forall v \in X, \tag{3.3}$$

where $h(v) := \|v - x\|$.

From (3.1), (3.2), and (3.3), we have

$$x \in X, x \notin S \quad \text{and} \quad 0 < f(x) < +\infty. \tag{3.4}$$

On the other hand, (3.3) implies that the function

$$f(v)_+ + (t\mu)^{-1}h(v)$$

attains its minimum on X at x. Hence by property (ω_2) in Definition 2.1,

$$0 \in \partial_\omega[f(x)_+ + (t\mu)^{-1}h(x)]. \tag{3.5}$$

Since f is l.s.c. and $0 < f(x)$, there exists $\delta_1 > 0$ such that

$$0 < f(y) \quad \forall y \in x + \delta_1 B.$$

Thus by property (ω_1) in Definition 2.1 and (3.5),

$$0 \in \partial_\omega(f + (t\mu)^{-1}h)(x). \tag{3.6}$$

Let $\delta := \min\{f(x), (1 - t^{-1})\mu^{-1}, \delta_1, \alpha\epsilon(2+\alpha)^{-1}\}$. Then by property (ω_4) in Definition 2.1 and (3.6), there exist x_1 and x_2 both in $x + \delta B$ such that

$$f(x) - \delta < f(x_1) < f(x) + \delta$$

and

$$0 \in \partial_\omega f(x_1) + \partial_\omega((t\mu)^{-1}h)(x_2) + \delta B^*.$$

The inequalities mean that $x_1 \in x + \delta B$ and $0 < f(x_1) < +\infty$. The inclusion, by property (ω_3) in Definition 2.1, implies that there exists

$$\xi \in \partial_\omega f(x_1)$$

such that

$$\|\xi\|_* < (t\mu)^{-1} + \delta \le (t\mu)^{-1} + (1 - t^{-1})\mu^{-1} = \mu^{-1},$$

which contradicts the assumption since

$$\begin{aligned} 0 < f(x_1) &< f(x) + \delta \le f(u)_+ + (t\mu)^{-1}\|u - x\| + \delta \\ &\le f(u) + (t\mu)^{-1}\gamma + \delta = 2f(u) + \delta \\ &\le \frac{2\epsilon}{2+\alpha} + \frac{\alpha\epsilon}{2+\alpha} = \epsilon \end{aligned}$$

$$\begin{aligned} \Big(\text{or } \|x_1 - x_0\| &\le \|x_1 - x\| + \|x - u\| + \|u - x_0\| < \delta + \gamma + \frac{\epsilon}{2+\alpha} \\ &\le \frac{\alpha\epsilon}{2+\alpha} + d_S(u) + \frac{\epsilon}{2+\alpha} \le \frac{(1+\alpha)\epsilon}{2+\alpha} + \|u - x_0\| \\ &\le \frac{(1+\alpha)\epsilon}{2+\alpha} + \frac{\epsilon}{2+\alpha} = \epsilon\Big). \qquad \square \end{aligned}$$

Remark 3.1. In terms of the proximal subdifferential in a Hilbert space, Clarke et al. [4, Theorem 3.3.1] indicates that the inequality $d_S(x) \le \mu f(x)_+$ holds if x is sufficiently near x_0 and $0 < f(x)$ is sufficiently small. (For more discussion about Clarke et al. [4, Theorem 3.3.1], see Ye [23, Claim].) Theorem 3.1 guarantees the inequality to be true if x is sufficiently near x_0 (or $0 < f(x)$ is sufficiently small).

If X is an Asplund space and f is Fréchet differentiable, the Fréchet subdifferential can be taken as $\partial_\omega f$. Theorem 3.1 applied in this case gives the following corollary. Note that a Fréchet differentiable function may not be Lipschitz continuous. The result cannot obtained by Ioffe [11, Theorem 1 or Corollary 1.1].

COROLLARY 3.2. *Let $f : X \to (-\infty, +\infty]$ be l.s.c. on an Asplund space X. Assume that $x_0 \in S := \{x \in X : f(x) \le 0\}$ and that there exist $\mu > 0$ and $0 < \epsilon \le \infty$ such that f is Fréchet differentiable at any x with $0 < f(x) < \epsilon$ (or $\|x - x_0\| < \epsilon$ and $0 < f(x) < +\infty$), and*

$$\|\nabla f(x)\|_* \ge \mu^{-1}.$$

Then we have

$$d_S(x) \le \mu f(x)_+ \quad \forall x \in X \text{ with } f(x) < \epsilon/2 \text{ (or } \|x - x_0\| < \epsilon/2).$$

The result in Theorem 3.1 for a single inequality system can easily be extended to a system including equalities, inequalities, and an abstract constraint $x \in C$ as follows.

THEOREM 3.3. *Let C be a closed subset of X and each $f_i, |g_j| : X \to (-\infty, +\infty]$ be l.s.c. for $i = 1, \ldots, r$ and $j = 1, \ldots, s$. Assume that*

$$x_0 \in S := \{x \in C : f_1(x) \le 0, \ldots, f_r(x) \le 0;\ g_1(x) = 0, \ldots, g_s(x) = 0\},$$

and denote

$$f(x) = \max\{f_1(x), \ldots, f_r(x);\ |g_1(x)|, \ldots, |g_s(x)|\}.$$

Suppose that there exist $\mu > 0$ and $0 < \epsilon \le \infty$ such that

$$\|\xi\|_* \ge \mu^{-1}$$

whenever $\xi \in \partial_\omega(f + \psi_C)(x)$ for any $x \in C$ with $0 < f(x) < \epsilon$ (or $\|x - x_0\| < \epsilon$ and $0 < f(x) < +\infty$). Then we have

$$d_S(x) \le \mu f(x)_+ \le \mu(\|F(x)_+\| + \|G(x)\|)$$

for any $x \in C$ *with* $f(x) < \epsilon/2$ (*or* $\|x - x_0\| < \epsilon/2$).

Proof. By Theorem 3.1, it suffices to check that f is l.s.c.

For any $x \in X$, denote $F_i(x) = f_i(x)$ for $i = 1, \ldots, r$, and $F_i(x) = |g_{i-r}(x)|$ for $i = r+1, \ldots, r+s$. Then for each $1 \le i \le r+s$, $F_i(x)$ is l.s.c.,

$$\begin{aligned}\liminf_{y \to x} f(y) &= \liminf_{y \to x} \max\{F_i(y) : 1 \le i \le r+s\} \\ &\ge \liminf_{y \to x} F_i(y) \ge F_i(x),\end{aligned}$$

and hence

$$\liminf_{y \to x} f(y) \ge f(x) \quad \forall x \in X,$$

which implies that f is l.s.c. □

Remark 3.2. (i) We have proved Theorem 3.3 based on Theorem 3.1, while Theorem 3.1 can be obtained from Theorem 3.3 by taking $C = X$, $r = 1$, and $s = 0$ in it. Therefore they are equivalent to each other. Besides, for the cases $\epsilon = +\infty$ and $\epsilon < +\infty$, Theorems 3.1 and 3.3 both give the corresponding sufficient conditions for global error bounds and local error bounds, respectively.

(ii) Theorem 3.3 has extended Ioffe [11, Theorem 1 and Corollary 1.1] from a Lipschitz equality system to an l.s.c. inequality system. It is also an extension of Wu [21, Theorem 4.19] in which $X = R^n, r = 1, s = 0, \epsilon = +\infty$, and $\partial_\omega = \partial^\pi$.

Theorem 3.3 is stated in terms of any ∂_ω-subdifferentials; however, to simplify checking the conditions, we often try to use smaller ∂_ω-subdifferentials (such as the proximal subdifferential in a Hilbert space and the Fréchet subdifferential in an Asplund space) or some ∂_ω-subdifferentials with better properties (for example, the Clarke subdifferential). Besides, in Theorem 3.3, only $|g_i|$ is required to be l.s.c. no matter whether g is. These points are illustrated in the following example.

Example 3.1. Consider the function $g : R \to R$ given by

$$g(x) = \begin{cases} 1 - |x| & \text{if } x \text{ is a rational number;} \\ -1 + |x| & \text{if } x \text{ is a irrational number.} \end{cases}$$

Take $C = R$. Then $S = \{x \in R : g(x) = 0\} = \{-1, 1\}, \psi_C(x) = 0$, and

$$|g(x)| = |1 - |x|| = \begin{cases} 1 - |x| & \text{if } |x| \le 1; \\ |x| - 1 & \text{if } |x| > 1 \end{cases}$$

is l.s.c. (in fact it is Lipschitz of rank 1). It is easy to find

$$\begin{aligned}&\partial^\pi |g(x)| = \{-1\} \quad \text{for} \quad x < -1 \quad \text{or} \quad 0 < x < 1, \\ &\partial^\pi |g(x)| = \{1\} \quad \text{for} \quad -1 < x < 0 \quad \text{or} \quad 1 < x, \quad \text{and} \\ &\partial^\pi |g(0)| = \emptyset.\end{aligned}$$

For any $x \in C$ with $g(x) \ne 0$, since

$$\partial^\pi(|g| + \psi_C)(x) = \partial^\pi |g(x)| \subseteq \{-1, 1\},$$

we have $\|\xi\| = 1$ for any $\xi \in \partial^\pi(|g| + \psi_C)(x)$. Thus, by Theorem 3.3,

$$d_S(x) \le |g(x)| = |1 - |x|| \quad \forall x \in R.$$

Remark 3.3. Note $d_S(0) = 1 = |g(0)|$ in this example. Thus $\mu = 1$ is the smallest constant such that the above inequality holds for any x in R. Besides, to use Theorem 3.3 to find a global error bound, we cannot use the Clarke subdifferential since if we choose it as a ∂_ω-subdifferential, then $\partial_\omega g(0) = \partial g(0) = [-1, 1]$ and it is impossible to find a μ to satisfy the condition in Theorem 3.3.

Let Y be a real normed linear space and $F : X \times Y \to (-\infty, +\infty]$ be l.s.c. For any fixed $y \in Y$, the partial subdifferential $\partial_x F(x, y)$ at $(x, y) \in X \times Y$ in x defined in Jourani [13] is in fact a ∂_ω-subdifferential of $F(x, y)$ at x (denoted by $\partial^x_\omega F(x, y)$) when $F(x, y)$ is considered as a function of the first variable. Since we use the fuzzy sum rule in the definition of ∂_ω-subdifferential instead of the sum rule as in that of the partial subdifferential, ∂_ω-subdifferentials include more subdifferentials in nonsmooth analysis than partial subdifferentials. For example, for the case $F(x, y) = f(x)\ \forall y \in Y$ the proximal subdifferential $\partial^\pi F(x, y) = \partial^\pi f(x)$ is a ∂_ω-subdifferential but not a partial subdifferential.

Now applying Theorem 3.3 to a function F defined on $X \times Y$, we have the following result of which Jourani [13, Theorem 2.4] is a special case when we take $C = X \times Y$ and $\epsilon = +\infty$.

THEOREM 3.4. *Let $F : X \times Y \to (-\infty, +\infty]$ satisfy that for each $y \in Y$ the function $F(\cdot, y)$ is l.s.c. Let C be a nonempty closed subset of $X \times Y$. Assume that for $y \in Y$ the set*

$$S(y) := \{x \in X : (x, y) \in C \text{ and } F(x, y) \le 0\}$$

is nonempty and that there exist $\mu > 0$ and $0 < \epsilon \le \infty$ such that

$$\|\xi\|_* \ge \mu^{-1} \quad \forall \xi \in \partial^x_\omega (F + \psi_C)(x, y)$$

for any $x \in X$ with $(x, y) \in C$ and $0 < F(x, y) < \epsilon$. Then we have

$$d_{S(y)}(x) \le \mu F(x, y)_+ \quad \forall x \in X \text{ with } (x, y) \in C \text{ and } F(x, y) < \epsilon/2.$$

Proof. For $y \in Y$ in the assumption, denote

$$f(\cdot) := F(\cdot, y) \text{ and } C(y) := \{x \in X : (x, y) \in C\}.$$

Upon applying Theorem 3.3 to the solution set

$$S(y) = \{x \in C(y) : f(x) \le 0\}$$

we obtain the inequality desired. □

4. Sufficient conditions for a general system. In this section we suppose that X is a real normed linear space. Motivated by a note of a referee of Deng [6, Corollary 2], we present the following condition to guarantee the existence of a global error bound for a general inequality system.

THEOREM 4.1. *Let f be an extended real-valued function on a subset C of X and $S = \{x \in C : f(x) \le 0\}$ be nonempty. Suppose that there exist a unit vector u in X and a constant $\mu > 0$ such that for any $\lambda > 0$,*

$$x + \lambda u \in C \text{ and } \sup_{\lambda > 0} \frac{f(x + \lambda u) - f(x)}{\lambda} \le -\mu^{-1} \tag{4.1}$$

for any $x \in C \backslash S$ with $f(x) < +\infty$. Then

$$d_S(x) \le \mu f(x)_+ \quad \forall x \in C.$$

Proof. It suffices to show that the inequality holds for $x \in C \backslash S$ with $f(x) < +\infty$. Now for such an x, $0 < f(x) < +\infty$, $x + \lambda u \in C$, and $f(x + \lambda u) \le f(x) - \mu^{-1}\lambda$ for any $\lambda > 0$, so by taking $\lambda = \mu f(x)$, we have $f(x + \lambda u) \le 0$, i.e., $x + \lambda u \in S$. Thus $d_S(x) \le \|\lambda u\| = \mu f(x)$. □

Remark 4.1. It is easy to see that C in Theorem 4.1 must be unbounded since for $x \in C$ with $f(x) < +\infty$ and any $\lambda > 0$, $x + \lambda u$ must be in C.

Recall that for a nonempty closed convex subset C of X, *the recession cone* of C, denoted by C^∞, is the set

$$C^\infty = \left\{ x \in X : \exists \{\mu_i\} \subseteq (0, +\infty) \ \& \ \{x_i\} \subseteq C \text{ s.t. } \lim_{i\to\infty} \mu_i = 0 \text{ and } \lim_{i\to\infty} \mu_i x_i = x \right\}.$$

According to Rockafellar [19, Theorem 2A(c)], C^∞ can equivalently be expressed as

$$C^\infty = \{x \in X : C + \{x\} \subseteq C\}.$$

For an l.s.c. and proper convex function $f : X \to (-\infty, +\infty]$, since its epigraph is a closed convex subset of $X \times R$, one can use the recession cone of *epi* f to define *the recession function of* f, denoted by f^∞, i.e.,

$$epi(f^\infty) = (epi\, f)^\infty.$$

We refer to [20] for examples of recession functions.

Similar to Deng [5, 6], we use the recession function to give the following sufficient condition for a global error bound.

COROLLARY 4.2. *Let C be a closed convex subset of X and each $f_i : X \to (-\infty, +\infty]$ be l.s.c. proper convex for $i \in I = \{1, \ldots, r\}$. Assume that $S = \{x \in C : f_i(x) \le 0, i \in I\}$ is nonempty and denote $f(x) := \max\{f_i(x) : i \in I\}$. Suppose that there exist a unit vector $u \in C^\infty$ and a constant $\mu > 0$ such that $f_i^\infty(u) \le -\mu^{-1}$ for each $i \in I$. Then for any $1 \le p \le +\infty$,*

$$d_S(x) \le \mu f(x)_+ \le \mu \|F(x)_+\|_p \quad \forall x \in C,$$

where $\|\cdot\|_p$ denotes the p-norm on R^r.

Proof. Since $S = \{x \in C : f(x) \le 0\}$, we need only to check that the conditions in Theorem 4.1 are satisfied for C and f.

First, by Rockafellar [19, Theorem 2A(a)], the inclusion $u \in C^\infty$ implies that $x + \lambda u$ must be in C for each $x \in C$ and any $\lambda \ge 0$. Besides, according to Rockafellar [19, Corollary 3C(a)], for each $i \in I$,

$$f_i^\infty(u) = \sup_{\lambda>0} \frac{f_i(x + \lambda u) - f_i(x)}{\lambda} \quad \forall x \in dom\, f_i.$$

So if $f_i^\infty(u) \le -\mu^{-1}$, then for any $\lambda > 0$,

$$f_i(x + \lambda u) \le f_i(x) - \lambda\mu^{-1} \quad \forall x \in dom\, f_i.$$

Hence for any $x \in dom\, f$ and any $\lambda > 0$,

$$f(x + \lambda u) \le f(x) - \lambda\mu^{-1}.$$

In particular, for any $x \in C \backslash S$ with $f(x) < +\infty$,

$$\sup_{\lambda>0} \frac{f(x + \lambda u) - f(x)}{\lambda} \le -\mu^{-1}.$$

Therefore, for any $1 \le p \le +\infty$, by Theorem 4.1,

$$d_S(x) \le \mu f(x)_+ \le \mu \|F(x)_+\|_p \quad \forall x \in C. \qquad \square$$

Remark 4.2. Note that each f_i in Corollary 4.2 is an l.s.c. and convex function on a real normed linear space. So it is an improvement on Deng [5, Theorem 2.3], in which X is a reflexive Banach space and each f_i is a continuous and convex function for $i = 1, \ldots, r$. Besides, Deng [6, Corollary 2] can be obtained as a special case of Corollary 4.2 where $p = 1$ and f_1 is a continuous and convex function on a Banach space X. Furthermore, Corollary 4.2 not only extends Jourani [13, Theorem 3.3] but also proves that condition (i) in it is redundant.

5. Global error bounds and metric regularity. In Deng [6] close relations between global error bounds and metric regularity are revealed for a continuous and convex inequality system. Most of them turn out to be true for an l.s.c. convex inequality system, and some of them can further be refined. To show this we recall the concept of metric regularity and introduce that of uniformly metric regularity.

DEFINITION 5.1. *Let f be an extended real-valued function on X, C be a subset of X, and $S = \{x \in C : f(x) \le 0\}$ be nonempty. The system*

$$f(x) \le 0, \ x \in C, \tag{5.1}$$

is said to be metrically regular *at a nonempty set $S_1 \subseteq S$ if there exist constants $\delta > 0$ and $\mu(\delta) > 0$ such that*

$$d_S(x) \le \mu(\delta) f(x)_+ \quad \forall x \in C \text{ with } d_{S_1}(x) \le \delta.$$

When $S_1 = \{z\} \subseteq S$, we simply say that the system (5.1) *is* metrically regular *at z. In particular, the system* (5.1) *is said to be* uniformly metrically regular *at S if it is metrically regular at each point of S with the same $\delta > 0$ and $\mu(\delta) > 0$.*

Obviously for any $\emptyset \ne S_1 \subseteq S_2$ we have $d_{S_1}(x) \ge d_{S_2}(x)$ for any $x \in X$, so if the system (5.1) is metrically regular at S_2, then it must also be metrically regular at S_1.

As the referees of this paper pointed out, the notion of metric regularity is related to moving sets, and the equivalence between error bound and (the very definition of) metric regularity usually fails to hold. The following result states the relations between global error bounds and metric regularity for an l.s.c. inequality system.

THEOREM 5.2. *Let f be an l.s.c. extended real-valued function on X and $S = \{x \in X : f(x) \le 0\}$ be nonempty. Consider the following statements:*

(a) *There is a constant $\mu > 0$ such that $d_S(x) \le \mu f(x)_+$ for any $x \in X$.*
(b) *The system* (5.1) *is metrically regular at any nonempty set $S_1 \subseteq S$.*
(c) *The system* (5.1) *is metrically regular at S.*
(d) *The system* (5.1) *is uniformly metrically regular at S.*
(e) *The system* (5.1) *is metrically regular at each point of S.*

Then the following implications hold:

(i) (a) $\Rightarrow$ (b) $\Leftrightarrow$ (c) $\Leftrightarrow$ (d) $\Rightarrow$(e).
(ii) *If f is convex,* (a) $\Leftrightarrow$ (b) $\Leftrightarrow$ (c) $\Leftrightarrow$ (d).
(iii) *If f is convex and S is compact,* (a) $\Leftrightarrow$ (b) $\Leftrightarrow$ (c) $\Leftrightarrow$ (d) $\Leftrightarrow$ (e).

Proof. Since the implications (a) $\Rightarrow$ (b) $\Rightarrow$ (c) $\Rightarrow$ (d) $\Rightarrow$ (e) in (i) are obvious, it suffices to show (d) $\Rightarrow$ (b) for (i), (d) $\Rightarrow$ (a) for (ii), and (e) $\Rightarrow$ (a) for (iii). But since the last implication was proved in Deng [6, Corollary 4] (assuming that $X = R^n$, f is continuous and convex, and S is bounded) and the proof is still valid with the

hypothesis in this theorem, it remains to prove (d) ⇒ (b) for (i) and (d) ⇒ (a) for (ii).

(d) ⇒ (b) for (i): We suppose that statement (d) is true. Then there are constants $\delta > 0$ and $\mu(\delta) > 0$ such that for each $z \in S$,

$$d_S(x) \le \mu(\delta) f(x)_+ \text{ whenever } \|x - z\| \le \delta.$$

Hence for any nonempty subset S_1 of S, we have $d_S(x) \le \mu(\delta) f(x)_+$ for any x with $d_{S_1}(x) \le \delta/2$ since for such an x we can find a point $x_1 \in S_1$ such that $\|x - x_1\| \le \delta$. This proves that statement (b) holds.

(d) ⇒(a) for (ii): Suppose that f is an l.s.c. proper convex function, that (d) holds, and that $\delta > 0$ is the constant in the definition of uniformly metric regularity. Then S is closed, $d_S(x) > 0$ for any $x \in X \backslash S$. Thus for any fixed $x \in X \backslash S$ and any $\epsilon > 0$, there exists $\overline{x} \in S$ such that $\|x - \overline{x}\| \le d_S(x) + \epsilon$. If $\|x - \overline{x}\| \le \delta$, we already have the inequality $d_S(x) \le \mu(\delta) f(x)_+$. If $\|x - \overline{x}\| > \delta$, taking $\lambda := \delta / \|x - \overline{x}\|$ and $y = \lambda x + (1 - \lambda)\overline{x}$, we have

$$\|y - \overline{x}\| = \lambda \|x - \overline{x}\| = \delta,$$

which implies $d_S(y) \le \mu(\delta) f(y)_+$. Besides, by the convexity of f, $f(y) \le \lambda f(x)$. Hence

$$\begin{aligned} d_S(x) &\le \|x - \overline{x}\| = \|y - \overline{x}\|/\lambda = [\|x - \overline{x}\| - \|y - x\|]/\lambda \\ &\le [d_S(x) + \epsilon - \|y - x\|]/\lambda \le [d_S(y) + \epsilon]/\lambda \\ &\le [\mu(\delta) f(y)_+ + \epsilon]/\lambda \le [\mu(\delta)\lambda f(x) + \epsilon]/\lambda \\ &= \mu(\delta) f(x)_+ + \epsilon/\lambda = \mu(\delta) f(x)_+ + \epsilon[d_S(x) + \epsilon]/\delta. \end{aligned}$$

This explains that statement (a) is true since $\epsilon > 0$ and x are arbitrary. □

Remark 5.1. Deng [6] proved the implications (a) ⇔ (b) ⇔ (c) for a continuous convex system on a Banach space, and the implication (e) ⇒ (a) when $X = R^n$ and S is bounded. Theorem 5.2 has extended these results to an l.s.c. system and contains the new equivalent statement (d). Furthermore Theorem 5.2 is allowed to be applied to an l.s.c. extended real-valued function f defined on a closed convex subset C of X to obtain an equivalent result whose statement is the same as that of Theorem 5.2 with the set $\{x \in X : f(x) \le 0\}$ and the inequality "$d_S(x) \le \mu f(x)_+$ for any $x \in X$" replaced by $\{x \in C : f(x) \le 0\}$ and "$d_S(x) \le \mu f(x)_+$ for any $x \in C$," respectively.

In the rest of this paper, we use Theorems 3.1 and 5.2 to give some sufficient conditions for l.s.c. systems to be metrically regular at sets.

PROPOSITION 5.3. *Let $f : X \to R$ be l.s.c. Assume that $S = \{x \in X : f(x) \le 0\}$ is nonempty and that there exist $\mu > 0$ and $0 < \varepsilon \le \infty$ such that for each $z \in S$,*

$$\|\xi\|_* \ge \mu^{-1}$$

whenever $\xi \in \partial_\omega f(x)$ for any $x \in X$ with $0 < f(x)$ and $\|x - z\| < \varepsilon$. Then the system (5.1) *is metrically regular at S. If f is in addition convex, then there is a constant $\mu > 0$ such that*

$$d_S(x) \le \mu f(x)_+ \quad \forall x \in X.$$

Proof. According to Theorem 5.2, it suffices to show that the system (5.1) is metrically regular at S.

By Theorem 3.1, the inequality

$$d_S(x) \le \mu f(x)_+$$

holds for each $z \in S$ and any $x \in X$ with $\|x - z\| < \varepsilon/2$, i.e., the system (5.1) is uniformly metrically regular at S. Hence by implication (i) of Theorem 5.2, the system (5.1) is metrically regular at S. □

The following proposition indicates that if the solution set is compact and contains no stationary points for ∂_ω-subdifferentials with some limiting property, then the system is metrically regular at the solution set.

PROPOSITION 5.4. *Let $f : X \to R$ be continuous. Assume that*

$$S = \{x \in X : f(x) \le 0\}$$

is nonempty and compact and that for each $z \in S$, $0 \notin \partial_\omega f(z)$ and $\partial_\omega f$ satisfies that $\xi \in \partial_\omega f(z)$ if $x_n \to z$, $\xi_n \in \partial_\omega f(x_n)$, and $\xi_n \to \xi$. Then the system (5.1) is metrically regular at S, and hence there is a constant $\mu > 0$ such that

$$d_S(x) \le \mu f(x)_+ \quad \forall x \in X.$$

Proof. Based on relation (iii) in Theorem 5.2, we need only to prove statement (e) in Theorem 5.2. Let $z \in S$ be fixed. Then by Theorem 3.1 it is enough to show that there exist $\mu > 0$ and $\epsilon > 0$ such that

$$\|\xi\|_* \ge \mu^{-1} \quad \forall \xi \in \partial_\omega f(x)$$

for any x with $\|x - z\| < \epsilon$ and $0 < f(x) < +\infty$. In fact, if this were not true, then there would exist sequences $\{x_n\}$ and $\{\xi_n\}$ such that $x_n \to z$, $\xi_n \in \partial_\omega f(x_n)$, and $\xi_n \to 0$. But this would lead to $0 \in \partial_\omega f(z)$, which contradicts the assumption. □

We now consider a convex system which also includes an abstract constraint. In the following proposition we prove that the generalized Slater condition is sufficient for metric regularity.

PROPOSITION 5.5. *Let $f_i : X \to R$ be locally Lipschitz and convex for $i \in I = \{1, \dots, r\}$, and let C be a closed and convex subset of X. Let $N \cup L$ be a partition of the index set I such that f_i is linear for each $i \in L$. Denote*

$$f(x) = \max\{f_i(x), |f_j(x)| : i \in N, j \in L\}.$$

Suppose that there exist

$$x^*, x_0 \in S := \{x \in C : f_i(x) \le 0, i \in N; f_j(x) = 0, j \in L\}$$

such that $f_i(x^*) < 0$ for each $i \in N$ and $\{-\nabla f_i(x_0) : i \in L\}$ is C-linearly independent, i.e.,

$$-\sum_{i \in L} \lambda_i \nabla f_i(x_0) \in N_C(x_0) \text{ implies } \quad \lambda_i = 0 \quad \forall i \in L.$$

Then there exist positive numbers δ and μ such that

(i) $\|\xi\|_* \ge \mu^{-1} \quad \forall \xi \in \partial f(x) + N_C(x)$ *for any $x \in C$ with $\|x - x_0\| < \delta$ and $0 < f(x)$;*

(ii) $d_S(x) \le \mu f(x)_+$ *for any $x \in C$ with $\|x - x_0\| < \delta/2$, i.e., the system (5.1) is metrically regular at x_0.*

Moreover, if $X = R^n$ *and* $\{-\nabla f_i(x) : i \in L\}$ *is* C*-linearly independent for each* $x \in S$*, then for any bounded subset* $\Omega \subseteq R^n$ *there exist* $\delta > 0$ *and* $\mu > 0$ *such that*

$$d_S(x) \leq \mu f(x)_+ \text{ for any } x \in C \cap (\Omega + \delta\overline{B}),$$

i.e., the system (5.1) *is metrically regular at* Ω.

Proof. Since f is Lipschitz near x and ψ_C is finite at x and both functions are convex, by Clarke [3, Corollary 1 of Theorem 2.9.8 and Proposition 2.4.12],

$$\partial(f + \psi_C)(x) = \partial f(x) + \partial\psi_C(x) = \partial f(x) + N_C(x).$$

Hence by applying Theorem 3.1 to the function $f + \psi_C$, statement (ii) follows from statement (i). So for statements (i) and (ii), it suffices to prove statement (i).

Suppose that statement (i) were not true. Then there would exist sequences $\{x_k\} \subseteq C$ and $\xi_k \in \partial f(x_k) + N_C(x_k)$ such that $x_k \to x_0$, $\xi_k \to 0$, and $0 < f(x_k)$ for each k. By Clarke [3, Proposition 2.3.12 and Theorem 2.3.9], for each x_k there exists a set of numbers $\lambda_i^{(k)}$ such that

$$\begin{gathered}
\lambda_i^{(k)} \geq 0 \quad \forall i \in N,\ \sum_{i\in N} \lambda_i^{(k)} + \sum_{i\in L} |\lambda_i^{(k)}| = 1, \\
\xi_k \in \sum_{i\in N} \lambda_i^{(k)} \partial f_i(x_k) \ + \sum_{i\in L} \lambda_i^{(k)} \nabla f_i(x_k) + N_C(x_k), \text{ and} \\
\lambda_i^{(k)}(f_i(x_k) - f(x_k)) = 0 \quad \forall i \in N,\ \lambda_i^{(k)}(|f_i(x_k)| - f(x_k)) = 0 \quad \forall i \in L.
\end{gathered}$$

Since each sequence $\{\lambda_i^{(k)}\}$ is bounded by 1 for each i, we can assume that $\lambda_i^{(k)} \to \lambda_i$ for each $i \in N \cup L$ as $k \to +\infty$. We denote the index of binding constraints at x_0 by $I(x_0) = \{i \in N : f_i(x_0) = 0\}$. Taking the limit as $k \to \infty$ gives

$$\begin{gathered}
\lambda_i \geq 0 \quad \forall\ i \in I(x_0), \quad \lambda_i = 0 \quad \forall i \in N \backslash I(x_0), \\
\sum_{i\in N} \lambda_i + \sum_{i\in L} |\lambda_i| = 1, \text{ and} \\
0 \in \sum_{i\in I(x_0)} \lambda_i \partial f_i(x_0) \ + \sum_{i\in L} \lambda_i \nabla f_i(x_0) + N_C(x_0),
\end{gathered}$$

where the inclusion follows from the fact that $\partial f_i(x_k)$ is the subdifferential of f_i at x_k and $N_C(x_k)$ is the normal cone to C at x_k in the sense of convex analysis. Since by assumption $\{-\nabla f_i(x_0) : i \in L\}$ is C-linearly independent, this inclusion implies that there is at least one $i_0 \in I(x_0)$ such that $\lambda_{i_0} > 0$, from which we would obtain a contradiction.

In fact, if we use the above λ_i to define the function

$$g(y) = \sum_{i\in I(x_0)} \lambda_i f_i(y) + \sum_{i\in L} \lambda_i f_i(y) + \psi_C(y),$$

then g is convex, and by the sum rule of subdifferentials (in the sense of convex analysis) we have

$$0 \in \sum_{i\in I(x_0)} \lambda_i \partial f_i(x_0) + \sum_{i\in L} \lambda_i \nabla f_i(x_0) + N_C(x_0) = \partial g(x_0),$$

which means that g attains its global minimum at x_0. Therefore this together with the continuity of g yields

$$0 = g(x_0) \le g(x^*) \le \lambda_{i_0} f_{i_0}(x^*) < 0.$$

This is a contradiction.

Now suppose that $X = R^n$. Let $\delta > 0$ be the positive number stated in (i). Then for any fixed bounded set Ω we can take $\epsilon > \delta$ such that $\Omega + \delta\overline{B} \subseteq \overline{B}(x_0, \epsilon/2)$. By Theorem 3.3, it suffices to show that there exists $\mu > 0$ such that $\|\xi\| \ge \mu^{-1} \quad \forall \xi \in \partial f(x) + N_C(x)$ for any $x \in C$ with $\delta \le \|x - x_0\| \le \epsilon$ and $0 < f(x)$.

Suppose that there exist sequences $\{x_k\} \subseteq C$ and $\xi_k \in \partial f(x_k) + N_C(x_k)$ such that $\delta \le \|x_k - x_0\| \le \epsilon$, $0 < f(x_k)$, and $\xi_k \to 0$ as $k \to +\infty$. Since $\{x_k\}$ lies in a compact set, we can assume that x_k converges to some point $\overline{x} \in C$ with $\delta \le \|\overline{x} - x_0\| \le \epsilon$. Taking the limit for $\xi_k \in \partial f(x_k) + N_C(x_k)$, we have $0 \in \partial f(\overline{x}) + N_C(\overline{x}) \subseteq \partial(f + \psi_C)(\overline{x})$ by the sum rule of subdifferentials in the sense of convex analysis. This means that f attains its global minimum over C at $\overline{x}$ since $f + \psi_C$ is convex. Note that f is continuous and $f(x_k)$ is positive. Thus

$$0 = f(x^*) \ge f(\overline{x}) = \lim_{k \to +\infty} f(x_k) \ge 0.$$

Thus $\overline{x} \in S$. But by statement (i) there exist positive numbers δ and μ such that

$$\|\xi\| \ge \mu^{-1} \quad \forall \xi \in \partial f(x) + N_C(x)$$

for any $x \in C$ with $\|x - \overline{x}\| < \delta$ and $0 < f(x)$. This contradicts the properties of subsequences $\{x_k\}$ and $\{\xi_k\}$. □

Remark 5.2. In Proposition 5.5, the Slater condition $f_i(x^*) < 0$ for $i \in N$ is important to guarantee that (i) and (ii) hold. Without this condition, (i) and (ii) may fail. One simple example is the function $f(x) = x^2$ with $S = \{x \in R : f(x) \le 0\} = \{0\}$. On the other hand, statement (i) is a local property, i.e., without additional conditions, property (i) cannot generally be extended to all points outside the neighborhood. For example, for any $n \ge 2$, the function

$$f(x) = \begin{cases} x - 1 & \text{if } x \ge 0; \\ -\sqrt{2 - (x+1)^2} & \text{if } -1 - \sqrt{\frac{2}{n^2+1}} < x < 0; \\ -\frac{x}{n} - \frac{1+\sqrt{2(n^2+1)}}{n} & \text{if } x \le -1 - \sqrt{\frac{2}{n^2+1}} \end{cases}$$

is differentiable and convex with $f(0) = -1$ and $f(1) = 0$. The inequality in statement (i) of Proposition 5.5 holds for $x_0 = 1$, $\delta = 1$, and $\mu = 1$. But for any $x < -1 - \sqrt{2(n^2+1)}$, $f(x) > 0$ and $|f'(x)| = 1/n < 1/\mu$.

REFERENCES

[1] A. Auslender and J. P. Crouzeix, *Global regularity theorems*, Math. Oper. Res., 13 (1988), pp. 243–253.

[2] J. V. Burke and P. Tseng, *A unified analysis of Hoffman's bound via Fenchel duality*, SIAM J. Optim., 6 (1996), pp. 265–282.

[3] F. H. Clarke, *Optimization and Nonsmooth Analysis*, Wiley-Interscience, New York, 1983.

[4] F. H. Clarke, Yu. S. Ledyaev, R. J. Stern, and P. R. Wolenski, *Nonsmooth Analysis and Control Theory*, Springer-Verlag, New York, 1998.

[5] S. Deng, *Computable error bounds for convex inequality systems in reflexive Banach spaces*, SIAM J. Optim., 7 (1997), pp. 274–279.

[6] S. Deng, *Global error bounds for convex inequality systems in Banach spaces*, SIAM J. Control Optim., 36 (1998), pp. 1240–1249.
[7] I. Ekeland, *On the variational principle*, J. Math. Anal. Appl., 47 (1974), pp. 324–353.
[8] M. Fabian, *Subdifferentiability and trustworthiness in the light of a new variational principle of Borwein and Preiss*, Acta Univ. Carolin. Math. Phys., 30 (1989), pp. 51–56.
[9] M. C. Ferris and J. S. Pang, *Nondegenerate solutions and related concepts in affine variational inequalities*, SIAM J. Control Optim., 34 (1996), pp. 244–263.
[10] A. J. Hoffman, *On approximate solutions of systems of linear inequalities*, J. Research Nat. Bur. Standards, 49 (1952), pp. 263–265.
[11] A. D. Ioffe, *Regular points of Lipschitz functions*, Trans. Amer. Math. Soc., 251 (1979), pp. 61–69.
[12] A. D. Ioffe, *Proximal analysis and approximate subdifferentials,* J. London Math. Soc., 41 (1990), pp. 175–192.
[13] A. Jourani, *Hoffman's error bound, local controllability, and sensitivity analysis*, SIAM J. Control Optim., 38 (2000), pp. 947–970.
[14] A. S. Lewis and J. S. Pang, *Error bounds for convex inequality systems,* in Generalized Convexity, Generalized Monotonicity: Recent Results, J.-P. Crouzeix, J.-E. Martinez-Legaz, and M. Volle, eds., Kluwer, Dordrecht, 1998, pp. 75–110.
[15] W. Li, *Abadie's constraint qualification, metric regularity, and error bounds for differentiable convex inequalities*, SIAM J. Optim., 7 (1997), pp. 966–978.
[16] X.-D. Luo and Z.-Q. Luo, *Extension of Hoffman's error bound to polynomial systems*, SIAM J. Optim., 4 (1994), pp. 383–392.
[17] O. L. Mangasarian, *A condition number for differentiable convex inequalities*, Math. Oper. Res., 10 (1985), pp. 175–179.
[18] S. M. Robinson, *An application of error bounds for convex programming in a linear space*, SIAM J. Control, 13 (1975), pp. 271–273.
[19] R. T. Rockafellar, *Level sets and continuity of conjugate convex functions*, Trans. Amer. Math. Soc., 123 (1966), pp. 46–63.
[20] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, NJ, 1970.
[21] Z. Wu, *Subdifferentials and Their Applications,* Masters thesis, The University of Victoria, Victoria, BC, Canada, 1997.
[22] J. J. Ye, *New uniform parametric error bounds*, J. Optim. Theory Appl., 98 (1998), pp. 197–219.
[23] J. J. Ye, *Constraint qualifications and necessary optimality conditions for optimization problems with variational inequality constraints*, SIAM J. Optim., 10 (2000), pp. 943–962.

SIAM J. OPTIM.
Vol. 12, No. 2, pp. 436–460

SMOOTHING FUNCTIONS FOR SECOND-ORDER-CONE COMPLEMENTARITY PROBLEMS*

MASAO FUKUSHIMA†, ZHI-QUAN LUO‡, AND PAUL TSENG§

Abstract. Smoothing functions have been much studied in the solution of optimization and complementarity problems with nonnegativity constraints. In this paper, we extend smoothing functions to problems in which the nonnegative orthant is replaced by the direct product of second-order cones. These smoothing functions include the Chen–Mangasarian class and the smoothed Fischer–Burmeister function. We study the Lipschitzian and differential properties of these functions and, in particular, we derive computable formulas for these functions and their Jacobians. These properties and formulas can then be used to develop and analyze noninterior continuation methods for solving the corresponding optimization and complementarity problems. In particular, we establish the existence and uniqueness of the Newton direction when the underlying mapping is monotone.

Key words. second-order cone, complementarity problem, smoothing function, Jordan algebra

AMS subject classifications. 90C33, 65K05

PII. S1052623400380365

1. Introduction.

The *second-order cone* (SOC) in $\Re^n$ ($n \geq 1$), also called the Lorentz cone, is defined to be

$$\mathcal{K}^n = \{(x_1, x_2) \in \Re \times \Re^{n-1} \mid \|x_2\| \leq x_1\},$$

where $\|\cdot\|$ denotes the Euclidean norm. If $n = 1$, then $\mathcal{K}^n$ is the set of nonnegative reals $\Re_+$. We are interested in optimization and complementarity problems whose constraints involve the direct product of SOCs. In particular, we wish to find vectors $x, y \in \Re^n$ and $\zeta \in \Re^\ell$ satisfying

$$\langle x, y\rangle = 0, \quad x \in \mathcal{K}, \quad y \in \mathcal{K}, \quad F(x, y, \zeta) = 0, \tag{1.1}$$

where $\langle\cdot,\cdot\rangle$ denotes the Euclidean inner product, $F : \Re^n \times \Re^n \times \Re^\ell \to \Re^n \times \Re^\ell$ is a continuously differentiable mapping, and

$$\mathcal{K} = \mathcal{K}^{n_1} \times \cdots \times \mathcal{K}^{n_m}, \tag{1.2}$$

with $\ell \geq 0$, $m, n_1, \ldots, n_m \geq 1$, and $n_1 + \cdots + n_m = n$. We will refer to (1.1), (1.2) as the *second-order-cone complementarity problem* (SOCCP). This problem has wide applications and, in particular, includes a large class of quadratically constrained problems as special cases [14]. It also includes as a special case the well-known nonlinear complementarity problem (NCP), corresponding to $n_i = 1$ for all i; i.e., $\mathcal{K}$ is

*Received by the editors November 2, 2000; accepted for publication (in revised form) June 17, 2001; published electronically December 14, 2001. This research was supported by a Grant-in-Aid for Scientific Research (B) from the Ministry of Education, Science, Sports and Culture of Japan. The third author was also supported by National Science Foundation grant CCR-9731273.
http://www.siam.org/journals/siopt/12-2/38036.html

†Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan (fuku@i.kyoto-u.ac.jp).

‡Department of Electrical and Computer Engineering, McMaster University, Hamilton, Ontario, L8S 4L7, Canada (luozq@mcmail.cis.mcmaster.ca).

§Department of Mathematics, University of Washington, Seattle, WA 98195 (tseng@math.washington.edu).

the nonnegative orthant $\Re^n_+$. In particular, when $\ell = 0$ and the mapping F has the form

$$F(x, y, \zeta) = F_0(x) - y \tag{1.3}$$

for some $F_0 : \Re^n \to \Re^n$, the SOCCP (1.1) becomes

$$\langle x, F_0(x)\rangle = 0, \quad x \in \mathcal{K}, \quad F_0(x) \in \mathcal{K},$$

which is a natural generalization of the ordinary NCP corresponding to $\mathcal{K} = \Re^n_+$.

Optimization problems with SOC constraints have been the focus of several recent studies. It is known that $\mathcal{K}^n$, like $\Re^n_+$ and the cone $\mathcal{S}^n$ of $n \times n$ real symmetric positive semidefinite matrices, belongs to the class of symmetric cones, to which a Jordan algebra may be associated [9]. Using this connection, interior-point methods have been developed for solving linear programs with SOC constraints [14, 15, 23] and, more generally, linear programs with symmetric cone constraints [1, 18]. An alternative approach based on reformulating SOC constraints as smooth convex constraints was studied in [24]. It is also known that $\mathcal{K}^n$ may be viewed as a linear section of $\mathcal{S}^n$. In particular, it is easily verified that

$$(x_1, x_2) \in \mathcal{K}^n \quad \Longleftrightarrow \quad \begin{bmatrix} x_1 & x_2^T \\ x_2 & x_1 I \end{bmatrix} \in \mathcal{S}^n, \tag{1.4}$$

where I denotes the identity matrix and superscript T denotes transpose. Thus, $\mathcal{K}^n = \mathcal{S}^n \cap \mathcal{A}$ for some subspace $\mathcal{A}$ of $\Re^{n\times n}$. Using this fact, we can reformulate any problem with SOC constraints as a problem with semidefinite cone constraints plus an additional subspace constraint. However, this increases the problem dimension from n to $n(n+1)/2$ and it is not known whether this increase can be mitigated by, say, exploiting the structure of $\mathcal{A}$ when carrying out the linear algebra in computation.

In this paper, we consider a (noninterior) smoothing approach to solving problems with SOC constraints. In this approach, we choose a continuously differentiable function $\phi_\mu : \Re^n \times \Re^n \to \Re^n$, parameterized by $\mu > 0$, such that the pointwise limit $\phi_0(x, y) = \lim_{\mu\to 0^+} \phi_\mu(x, y)$ satisfies

$$\langle x, y\rangle = 0, \quad x \in \mathcal{K}, \quad y \in \mathcal{K} \quad \Longleftrightarrow \quad \phi_0(x, y) = 0. \tag{1.5}$$

We note that ϕ_0 is typically nonsmooth. The smoothing function ϕ_μ leads to the following noninterior continuation method for solving (1.1): Starting with an initial $\mu > 0$, we solve approximately the smooth equations

$$\phi_\mu(x, y) = 0, \qquad F(x, y, \zeta) = 0 \tag{1.6}$$

by, say, applying a few Newton steps; then we decrease μ and repeat the iteration. This noninterior approach offers an attractive alternative to the interior approach and it has been much studied in the case of NCP. In this case, one popular choice of ϕ_μ is a smooth approximation of the "min" function suggested by Chen and Harker [3], Kanzow [13], and Smale [19], and further generalized by Chen and Mangasarian [4, 5] (also see [2, 7, 8, 10, 11, 16, 22, 25] and references therein). Another popular choice is a smooth approximation of the Fischer–Burmeister function suggested by Kanzow [13]. Recently, these smoothing functions and their nonsmooth analogues were extended to the setting of semidefinite complementarity problems (SDCP) [6, 20, 21]. Analogous to the semidefinite setting, our definition of smoothing functions is based on the spectral

factorization of vectors in $\Re^n$, as specified by the Jordan algebra associated with SOC. This appears to be the most natural and convenient way to extend a function defined for NCP to one defined for SOCCP, while preserving essential differential and Lipschitzian properties. The latter properties are needed in analyzing the convergence properties of the corresponding continuation method; see, e.g., [2, 7, 8, 22] for such analysis in the case of NCP, and [6] for an analysis in the setting of SDCP. Unlike the semidefinite setting, the simpler structure of an SOC allows for a more direct analysis of its differential properties. In particular, we do not need to invoke a local upper Lipschitzian property of the spectral factorization, as is done in [6].

In what follows, $\Re^n$ $(n \geq 1)$ denotes the space of n-dimensional real column vectors, $\Re^{n_1} \times \cdots \times \Re^{n_m}$ is identified with $\Re^{n_1+\cdots+n_m}$. Thus, $(x_1, \ldots, x_m) \in \Re^{n_1} \times \cdots \times \Re^{n_m}$ is viewed as a column vector in $\Re^{n_1+\cdots+n_m}$. For any $x, y \in \Re^n$ we write $x \succeq_{\mathcal{K}} y$ or $y \preceq_{\mathcal{K}} x$ (respectively, $x \succ_{\mathcal{K}} y$ or $y \prec_{\mathcal{K}} x$) if $x - y$ is in $\mathcal{K}$ (respectively, the interior of $\mathcal{K}$, denoted by int $\mathcal{K}$). For any square matrices $A, B \in \Re^{n\times n}$, we write $A \succ B$ (respectively, $A \succeq B$) if the symmetric part of $A - B$, namely $(A - B + A^T - B^T)/2$, is positive definite (respectively, positive semidefinite). Also, $\Re_+$ and $\Re_{++}$ denote the nonnegative and positive reals. For any $x, y \in \Re^n$, the Euclidean inner product and norm are denoted $\langle x, y\rangle = x^T y$ and $\|x\| = \sqrt{x^T x}$. For any Fréchet-differentiable mapping $G : \Re^n \to \Re^m$, we denote its (transposed) Jacobian at $x \in \Re^n$ by $\nabla G(x) \in \Re^{n\times m}$, i.e., $(G(x + u) - G(x) - \nabla G(x)^T u)/\|u\| \to 0$ as $u \to 0$.

2. Jordan algebra associated with the SOC. For any $x = (x_1, x_2) \in \Re \times \Re^{n-1}$ and $y = (y_1, y_2) \in \Re \times \Re^{n-1}$, we define their *Jordan product* as

$$x \cdot y = \big(x^T y,\ y_1 x_2 + x_1 y_2\big). \tag{2.1}$$

We will write x^2 to mean $x \cdot x$ and write $x + y$ to mean the usual componentwise addition of vectors. Then, $\cdot$, $+$, together with

$$e = (1, 0, \ldots, 0) \in \Re^n,$$

have the following basic properties (see [9, Chapter II]).

PROPERTY 2.1.

1. $e \cdot x = x \quad \forall x \in \Re^n$. (*Identity*)
2. $x \cdot y = y \cdot x \quad \forall x, y \in \Re^n$. (*Commutativity* 1)
3. $x \cdot (x^2 \cdot y) = x^2 \cdot (x \cdot y) \quad \forall x, y \in \Re^n$. (*Commutativity* 2)
4. $(x + y) \cdot z = x \cdot z + y \cdot z \quad \forall x, y, z \in \Re^n$. (*Distributivity*)

Notice that the Jordan product is *not associative* in general. For example, for $n = 3$ and $x = (1, -1, 1)$, $y = z = (1, 0, 1)$, we have

$$(x \cdot y) \cdot z = (4, -1, 4) \quad \neq \quad x \cdot (y \cdot z) = (4, -2, 4).$$

In fact, this example shows that $(x \cdot y) \cdot y \neq x \cdot y^2$ in general. On the other hand, it can be verified that associativity holds under the inner product in the sense that

$$\langle x, y \cdot z\rangle = \langle y, z \cdot x\rangle = \langle z, x \cdot y\rangle \quad \forall x, y, z \in \Re^n.$$

The Jordan product (2.1) is associated with the SOC $\mathcal{K}^n$ via the following useful facts.

1. For each $x = (x_1, x_2) \in \Re \times \Re^{n-1}$, the *determinant* and the *trace* of x are defined by

$$\det(x) = x_1^2 - \|x_2\|^2 \quad \text{and} \quad \operatorname{tr}(x) = 2x_1,$$

respectively. Unlike matrices, we have in general $\det(x \cdot y) \neq \det(x)\det(y)$ unless $\alpha x_2 + \beta y_2 = 0$ for some $(\alpha, \beta) \neq (0,0) \in \Re^2$.

2. A vector $x = (x_1, x_2) \in \Re \times \Re^{n-1}$ is said to be *invertible* if $\det(x) \neq 0$. If x is invertible, then there exists a unique $y = (y_1, y_2) \in \Re \times \Re^{n-1}$ satisfying $x \cdot y = e$. We shall call this y the *inverse* of x and denote it by x^{-1}. Direct calculation yields

$$x^{-1} = \frac{1}{x_1^2 - \|x_2\|^2}(x_1, -x_2) = \frac{\mathrm{tr}(x)\, e - x}{\det(x)}. \tag{2.2}$$

From (2.2), it is clear that $x \in \mathrm{int}\, \mathcal{K}^n$ if and only if $x^{-1} \in \mathrm{int}\, \mathcal{K}^n$.

3. If $x \in \mathcal{K}^n$, then there exists a unique vector in $\mathcal{K}^n$, which we denote by $x^{1/2}$, such that $(x^{1/2})^2 = x^{1/2} \cdot x^{1/2} = x$. Direct calculation yields

$$x^{1/2} = \left(s, \frac{x_2}{2s}\right), \quad \text{where} \quad s = \sqrt{\left(x_1 + \sqrt{x_1^2 - \|x_2\|^2}\right)/2}. \tag{2.3}$$

In the above formula, the term x_2/s is defined to be the zero vector if $x_2 = 0$ and $s = 0$, i.e., $x = 0$.

4. For any $x \in \Re^n$, we have $x^2 \in \mathcal{K}^n$. Hence there exists a unique vector $(x^2)^{1/2} \in \mathcal{K}^n$, which we denote by $|x|$. Clearly we have $x^2 = |x|^2$.

Notice that the SOC $\mathcal{K}^n$ is *not closed* under the Jordan product. For example,

$$x = (\sqrt{2}, 1, 1) \in \mathcal{K}^3, \quad y = (\sqrt{2}, 1, -1) \in \mathcal{K}^3, \qquad \text{but} \qquad x \cdot y = (2, 2\sqrt{2}, 0) \notin \mathcal{K}^3.$$

For any $x = (x_1, x_2) \in \Re \times \Re^{n-1}$, we define the symmetric matrix

$$L_x = \begin{bmatrix} x_1 & x_2^T \\ x_2 & x_1 I \end{bmatrix}, \tag{2.4}$$

viewed as a linear mapping from $\Re^n$ to $\Re^n$. It is easily verified that

$$L_x y = x \cdot y \quad \forall y \in \Re^n.$$

Moreover, L_x is positive definite (and hence invertible) if and only if $x \in \mathrm{int}\, \mathcal{K}^n$ (see (1.4)). Notice, however, that in general $L_x^{-1} y \neq x^{-1} \cdot y$ for $x \in \mathrm{int}\, \mathcal{K}^n$ and $y \in \Re^n$.

We next introduce the spectral factorization of vectors in $\Re^n$ associated with $\mathcal{K}^n$. Let $x = (x_1, x_2) \in \Re \times \Re^{n-1}$. Then x can be decomposed as

$$x = \lambda_1 u^{(1)} + \lambda_2 u^{(2)}, \tag{2.5}$$

where λ_1, λ_2 and $u^{(1)}$, $u^{(2)}$ are the spectral values and the associated spectral vectors of x given by

$$\lambda_i = x_1 + (-1)^i \|x_2\|, \tag{2.6}$$

$$u^{(i)} = \begin{cases} \frac{1}{2}\left(1, (-1)^i \dfrac{x_2}{\|x_2\|}\right) & \text{if } x_2 \neq 0, \\ \frac{1}{2}\left(1, (-1)^i w\right) & \text{if } x_2 = 0, \end{cases} \tag{2.7}$$

for $i = 1, 2$, with w being any vector in $\Re^{n-1}$ satisfying $\|w\| = 1$. If $x_2 \neq 0$, decomposition (2.5) is unique.

Some interesting properties of λ_1, λ_2 and $u^{(1)}$, $u^{(2)}$ are summarized below. Notice that the identity element e is uniquely identified by its two spectral values which are exactly equal to 1.

PROPERTY 2.2. *For any $x = (x_1, x_2) \in \Re \times \Re^{n-1}$, the spectral values λ_1, λ_2 and spectral vectors $u^{(1)}$, $u^{(2)}$, as given by* (2.6) *and* (2.7)*, have the following properties:*

1. $u^{(1)}$ *and* $u^{(2)}$ *are orthogonal under the Jordan product and have length* $1/\sqrt{2}$*, i.e.,*

$$u^{(1)} \cdot u^{(2)} = 0, \qquad \|u^{(1)}\| = \|u^{(2)}\| = 1/\sqrt{2}.$$

2. $u^{(1)}$ *and* $u^{(2)}$ *are idempotent under the Jordan product, i.e.,*

$$u^{(i)} \cdot u^{(i)} = u^{(i)}, \quad i = 1, 2.$$

3. λ_1 *and* λ_2 *are nonnegative* (*respectively, positive*) *if and only if* $x \in \mathcal{K}^n$ (*respectively,* $x \in \operatorname{int} \mathcal{K}^n$).
4. *The determinant, the trace, and the Euclidean norm of* x *can all be represented in terms of* λ_1 *and* λ_2*:*

$$\det(x) = \lambda_1\lambda_2, \quad \operatorname{tr}(x) = \lambda_1 + \lambda_2, \quad 2\|x\|^2 = \lambda_1^2 + \lambda_2^2.$$

It can be verified that $\lambda_1,\ \lambda_2$ are in fact the eigenvalues of the $n \times n$ matrix L_x (see (2.4)), with $u^{(1)}$, $u^{(2)}$ being the corresponding eigenvectors. The remaining $n-2$ eigenvalues of this matrix are identically x_1, with corresponding eigenvectors of the form $(0, v)$, where v lies in the subspace of $\Re^{n-1}$ orthogonal to x_2. A corollary of this is the equivalence (1.4). It can also be verified that λ_1, λ_2 are Lipschitz continuous in x, with Lipschitz constant $\sqrt{2}$, and that $u^{(1)}, u^{(2)}$ have a local upper Lipschitzian property. However, we will not need these properties in the subsequent analysis.

The next proposition shows that, in the case of $\mathcal{K} = \mathcal{K}^n$, the SOCCP (1.1) can be equivalently stated as

$$x \cdot y = 0, \quad x \in \mathcal{K}, \quad y \in \mathcal{K}, \quad F(x, y, \zeta) = 0.$$

This equivalence also extends to the general case in which $\mathcal{K}$ has the direct product structure (1.2), provided that the Jordan product is defined according to this structure (see Section 6).

PROPOSITION 2.1. *For any* x *and* y *in* $\Re^n$*, we have*

$$\langle x, y\rangle = 0,\ x \in \mathcal{K}^n,\ y \in \mathcal{K}^n \quad \Longleftrightarrow \quad x \cdot y = 0,\ x \in \mathcal{K}^n,\ y \in \mathcal{K}^n.$$

Proof. The "$\Leftarrow$" direction is obvious from the definition of a Jordan product. To prove the "$\Rightarrow$" direction, suppose $\langle x, y\rangle = 0,\ x \in \mathcal{K}^n,\ y \in \mathcal{K}^n$. Then,

$$\langle x, y\rangle = x_1y_1 + x_2^T y_2 = 0, \quad x_1 \geq \|x_2\|, \quad y_1 \geq \|y_2\|,$$

implying $-x_2^T y_2 = x_1y_1 \geq \|x_2\|\|y_2\|$. Since $-x_2^T y_2 \leq \|x_2\|\|y_2\|$, this shows that x_2, y_2 make an angle of 180° and, moreover, $x_1 = \|x_2\|$, $y_1 = \|y_2\|$. Thus,

$$x \cdot y = (\langle x, y\rangle, x_1y_2 + y_1x_2) = (0, \|x_2\|y_2 + \|y_2\|x_2) = 0.$$

This completes the proof. □

3. Functions associated with the SOC. For any function $\hat{g} : \Re \to \Re$, we define a function on $\Re^n$ associated with $\mathcal{K}^n$ $(n \geq 1)$ by

$$g(x) = \hat{g}(\lambda_1)u^{(1)} + \hat{g}(\lambda_2)u^{(2)} \quad \forall x = (x_1, x_2) \in \Re \times \Re^{n-1}, \tag{3.1}$$

where $\lambda_1, \lambda_2, u^{(1)}, u^{(2)}$ are the spectral values and vectors of x (see (2.5), (2.6), (2.7)). If $\hat{g}$ is defined on only a subset of $\Re$, then g is defined on the corresponding subset

of $\Re^n$. The definition (3.1) is clearly unambiguous when $x_2 \neq 0$, since $u^{(1)}, u^{(2)}$ are unique. When $x_2 = 0$, we see from (2.6), (2.7) that $g(x) = \hat{g}(x_1)e$, and thus the definition is again unambiguous. The cases of $g(x) = x^{1/2},\ x^2,\ \exp(x)$ are discussed in the book of Faraut and Korányi [9]. The above definition (3.1) is analogous to one associated with the semidefinite cone $\mathcal{S}^n$; see [6, 20]. In fact, if a function $\tilde{g}$ associated with $\mathcal{S}^n$ has the property that it maps the subspace $\mathcal{A}$ of matrices of the form (2.4) into $\mathcal{A}$, then $g = L^{-1} \circ \tilde{g} \circ L$ is a function associated with $\mathcal{K}^n$, where $L : \Re^n \to \mathcal{A}$ is the linear mapping such that $L(x)$ is the matrix L_x given by (2.4) for all $(x_1, x_2) \in \Re \times \Re^{n-1}$. However, the functions considered in [6, 20] do not have this property.

For any $x \in \Re^n$ and integer $k \geq 1$, we can take the Jordan product of x with itself k times to obtain the kth power of x, denoted by x^k. We define $x^0 = e$. If $x \in \operatorname{int} \mathcal{K}^n$, then $x^{-k} = (x^k)^{-1}$ is also well defined. The spectral factorization (2.5)–(2.7) and its properties (see Property 2.2) provide a very useful tool for evaluating functions defined using powers. For example, for any $x \in \Re^n$, the spectral factorization of x yields

$$\begin{aligned} x^2 &= (\lambda_1 u^{(1)} + \lambda_2 u^{(2)}) \cdot (\lambda_1 u^{(1)} + \lambda_2 u^{(2)}) \\ &= \lambda_1^2 (u^{(1)})^2 + \lambda_2^2 (u^{(2)})^2 + 2\lambda_1 \lambda_2 u^{(1)} \cdot u^{(2)} \\ &= \lambda_1^2 u^{(1)} + \lambda_2^2 u^{(2)}, \end{aligned} \tag{3.2}$$

where we used the orthogonal and idempotent properties of $u^{(1)}$ and $u^{(2)}$ (see Property 2.2). Notice that this is consistent with (3.1). The above formula shows that squaring a vector is the same as squaring the spectral values in its spectral factorization. A corollary of this formula is that, for any $x \in \Re^n$, the spectral values of x^2 are nonnegative and hence $x^2 \in \mathcal{K}^n$. Conversely, when $x \in \mathcal{K}^n$, we have from Property 2.2 that $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$, and thus $w = \sqrt{\lambda_1} u^{(1)} + \sqrt{\lambda_2} u^{(2)}$ is defined and in $\mathcal{K}^n$. Moreover, analogous to (3.2), $w^2 = x$. Thus, $x^{1/2} = w$, i.e.,

$$x^{1/2} = \sqrt{\lambda_1} u^{(1)} + \sqrt{\lambda_2} u^{(2)}. \tag{3.3}$$

The above derivation can be extended beyond powers to any function admitting a power series expansion. This is shown in the following proposition.

PROPOSITION 3.1. *Suppose $\hat{g} : \Re \to \Re$ admits a power series expansion $\hat{g}(\alpha) = \sum_{k=0}^{\infty} a_k \alpha^k$ for some real coefficients $a_0, a_1, \ldots$. Then the function $g : \Re^n \to \Re^n$ given by* (3.1) *satisfies*

$$g(x) \ = \ \sum_{k=0}^{\infty} a_k (\lambda_1^k u^{(1)} + \lambda_2^k u^{(2)}) \ = \ \sum_{k=0}^{\infty} a_k x^k \qquad \forall\, x \in \Re^n,$$

where λ_1, λ_2, $u^{(1)}$, $u^{(2)}$ are the spectral values and vectors of x given by (2.6), (2.7).

Proof. Using Property 2.2, we have for all $k \geq 1$

$$(u^{(i)})^k = u^{(i)}, \qquad u^{(i)} \cdot u^{(j)} = 0,$$

where $i, j = 1, 2$, and $i \neq j$. Using the binomial expansion, this yields

$$x^k = (\lambda_1 u^{(1)} + \lambda_2 u^{(2)})^k = \lambda_1^k u^{(1)} + \lambda_2^k u^{(2)}.$$

Substituting this into the definition of $g(x)$ yields

$$\begin{aligned} g(x) &= \hat{g}(\lambda_1)u^{(1)} + \hat{g}(\lambda_2)u^{(2)} \\ &= \left(\sum_{k=0}^{\infty} a_k \lambda_1^k\right) u^{(1)} + \left(\sum_{k=0}^{\infty} a_k \lambda_2^k\right) u^{(2)} \\ &= \sum_{k=0}^{\infty} a_k (\lambda_1^k u^{(1)} + \lambda_2^k u^{(2)}) \\ &= \sum_{k=0}^{\infty} a_k x^k. \qquad \square \end{aligned}$$

Define

$$\exp(x) = \sum_{k=0}^{\infty} \frac{x^k}{k!} \qquad \forall x \in \Re^n.$$

Also, define $\ln(x)$ to be the unique $w \in \Re^n$ satisfying $\exp(w) = x$ for each $x \in \operatorname{int} \mathcal{K}^n$. In the proposition below, we use Proposition 3.1 to derive a simple formula for $\exp(x)$. This in turn is used to show that $\ln(x)$ is well defined and has a simple formula.

PROPOSITION 3.2.

(a) *For any* $x = (x_1, x_2) \in \Re \times \Re^{n-1}$,

$$\exp(x) = \begin{cases} \frac{1}{2}\exp(x_1)\left(\cosh(\|x_2\|), \sinh(\|x_2\|)\dfrac{x_2}{\|x_2\|}\right) & \text{if } x_2 \neq 0, \\ \exp(x_1)\,(1,0) & \text{if } x_2 = 0, \end{cases}$$

where $\cosh(\alpha) = (\exp(\alpha) + \exp(-\alpha))/2$ *and* $\sinh(\alpha) = (\exp(\alpha) - \exp(-\alpha))/2$ *for* $\alpha \in \Re$.

(b) *For any* $x = (x_1, x_2) \in \operatorname{int} \mathcal{K}^n$, $\ln(x)$ *is well defined and*

$$\ln(x) = \begin{cases} \frac{1}{2}\left(\ln\left(x_1^2 - \|x_2\|^2\right), \ln\left(\dfrac{x_1 + \|x_2\|}{x_1 - \|x_2\|}\right)\dfrac{x_2}{\|x_2\|}\right) & \text{if } x_2 \neq 0, \\ \ln(x_1)\,(1,0) & \text{if } x_2 = 0. \end{cases}$$

Proof. (a) By Proposition 3.1, $\exp(x) = g(x)$ with $\hat{g}(\alpha) = \exp(\alpha)$. Using (3.1) with λ_1, λ_2 and $u^{(1)}, u^{(2)}$ given by (2.6) and (2.7), we have

$$\begin{aligned} \exp(x) &= \exp(\lambda_1)u^{(1)} + \exp(\lambda_2)u^{(2)} \\ &= \exp(x_1 - \|x_2\|)u^{(1)} + \exp(x_1 + \|x_2\|)u^{(2)} \\ &= \exp(x_1)(\exp(-\|x_2\|)u^{(1)} + \exp(\|x_2\|)u^{(2)}). \end{aligned}$$

We consider only the case of $x_2 \neq 0$. The case of $x_2 = 0$ can be argued analogously. Using (2.7), we have

$$u^{(i)} = \frac{1}{2}\left(1,\ \frac{(-1)^i x_2}{\|x_2\|}\right), \qquad i = 1, 2.$$

The previous expression simplifies to

$$\exp(x) = \frac{1}{2}\exp(x_1)\left(\exp(\|x_2\|) + \exp(-\|x_2\|),\ (\exp(\|x_2\|) - \exp(-\|x_2\|)) \cdot \frac{x_2}{\|x_2\|}\right).$$

(b) Fix any $x = (x_1, x_2) \in \operatorname{int} \mathcal{K}^n$, i.e., $\|x_2\| < x_1$. We will prove by construction that there is a unique $w \in \Re^n$ satisfying $\exp(w) = x$, which will show that $\ln(x)$ is well defined. We have from part (a) that $w = (w_1, w_2) \in \Re \times \Re^{n-1}$ satisfies $\exp(w) = x$ if and only if

$$\begin{aligned} x_1 &= \frac{1}{2}\exp(w_1)\left(\exp(\|w_2\|) + \exp(-\|w_2\|)\right), \\ x_2 &= \frac{1}{2}\exp(w_1)\left(\exp(\|w_2\|) - \exp(-\|w_2\|)\right)\frac{w_2}{\|w_2\|}. \end{aligned} \tag{3.4}$$

Letting $a = \exp(w_1)$ and $b = \exp(\|w_2\|)$, we obtain from (3.4) that

$$x_1 = \frac{a}{2}(b + b^{-1}), \qquad \|x_2\| = \frac{a}{2}(b - b^{-1}).$$

We can solve these equations uniquely for a and b to yield

$$b = \sqrt{\frac{x_1 + \|x_2\|}{x_1 - \|x_2\|}}, \qquad a = \frac{2x_1}{b + b^{-1}} = \sqrt{x_1^2 - \|x_2\|^2}.$$

Suppose $x_2 \neq 0$, so that $b \neq 1$. We obtain $w_1 = \ln(a)$, $\|w_2\| = \ln(b)$, and we have from (3.4) that $w_2 = 2a^{-1}\|w_2\|x_2/(b - b^{-1})$. Thus,

$$w = (w_1, w_2) = \left(\ln(a), 2a^{-1}\ln(b)x_2/(b - b^{-1})\right).$$

Using the above formulas for a and b and then simplifying expressions, we obtain the desired formula for w. Suppose instead $x_2 = 0$, so that $a = x_1$ and $b = 1$. We then obtain $w_1 = \ln(x_1)$, $\|w_2\| = \ln(1) = 0$, and thus $w_2 = 0$. □

It can be verified that $\ln(x)$ is alternatively given by (3.1) with $\hat{g}(\alpha) = \ln(\alpha)$. For any $x \in \Re^n$, we define $[x]_+$ to be the nearest-point (in the Euclidean norm) projection of x onto $\mathcal{K}^n$. For $\alpha \in \Re$, let $[\alpha]_+ = \max\{0, \alpha\}$. The following proposition shows that $|x|$ and $[x]_+$ have the form (3.1) and are related to each other as in the cases of nonnegative orthant $\Re^n_+$ and positive semidefinite cone $\mathcal{S}^n$.

PROPOSITION 3.3. *For any $x = (x_1, x_2) \in \Re \times \Re^{n-1}$, let λ_1, λ_2, $u^{(1)}, u^{(2)}$ denote the spectral values and vectors of x, i.e.,* (2.5)–(2.7). *Then the following results hold.*

(a) $|x| = (x^2)^{1/2} = |\lambda_1|u^{(1)} + |\lambda_2|u^{(2)}$.

(b) *The projection onto the SOC $\mathcal{K}^n$ can be written as*

$$[x]_+ = [\lambda_1]_+ u^{(1)} + [\lambda_2]_+ u^{(2)} = (x + |x|)/2. \tag{3.5}$$

Proof. (a) By using (3.2) and (3.3), we have

$$\begin{aligned} (x^2)^{1/2} &= (\lambda_1^2 u^{(1)} + \lambda_2^2 u^{(2)})^{1/2} \\ &= (\lambda_1^2)^{1/2} u^{(1)} + (\lambda_2^2)^{1/2} u^{(2)} \\ &= |\lambda_1|u^{(1)} + |\lambda_2|u^{(2)}. \end{aligned}$$

(b) Since $[\alpha]_+ = (\alpha + |\alpha|)/2$ for all $\alpha \in \Re$, the second equality in (3.5) follows from part (a) and (2.5). We now prove the first equality in (3.5). First, consider the case of $x \in \mathcal{K}^n$. Then $[x]_+ = x$. Also, $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$, and thus $[\lambda_1]_+ = \lambda_1$ and $[\lambda_2]_+ = \lambda_2$. Thus the first equality in (3.5) holds. Next, consider the case of $x \notin \mathcal{K}^n$. Then, $x_1 < \|x_2\|$ so that $[\lambda_1]_+ = [x_1 - \|x_2\|]_+ = 0$. Consider the projection problem:

$$\min \frac{1}{2}\|y - x\|^2 \quad \text{subject to} \quad \|y_2\| - y_1 \leq 0.$$

This problem is convex, its inequality constraint can be satisfied strictly, and its unique optimal solution $y = [x]_+$ satisfies the inequality constraint with equality. Hence this optimal solution satisfies the following necessary conditions for optimality (see [17, Theorem 28.2]):

$$y - x + \mu(-1, \nu) = 0, \quad \|y_2\| = y_1, \tag{3.6}$$

for some nonnegative $\mu \in \Re$ (μ is a Lagrange multiplier) and some subgradient $\nu \in \Re^{n-1}$ of $\|\cdot\|$ at y_2, i.e.,

$$\nu = y_2/\|y_2\| \quad \text{if } y_2 \neq 0, \qquad \|\nu\| \leq 1 \quad \text{if } y_2 = 0. \tag{3.7}$$

Suppose $y_2 \neq 0$. Then, solving (3.6) and (3.7) for y and μ and using $x_1 < \|x_2\|$ yield

$$y_1 = \frac{1}{2}[x_1 + \|x_2\|]_+ \qquad \text{and} \qquad y_2 = \begin{cases} \frac{1}{2}[x_1 + \|x_2\|]_+ x_2/\|x_2\| & \text{if } x_2 \neq 0, \\ 0 & \text{if } x_2 = 0. \end{cases}$$

If $x_2 = 0$, then $x_1 < 0$, and thus $[x_1 + \|x_2\|]_+ = 0$ and $y_2 = 0$, a contradiction of $y_2 \neq 0$. Thus $x_2 \neq 0$, and we obtain

$$(y_1, y_2) = \frac{1}{2}[x_1 + \|x_2\|]_+ \left(1, \frac{x_2}{\|x_2\|}\right) \;=\; [\lambda_2]_+ u^{(2)}.$$

Since $[\lambda_1]_+ = 0$, this proves the first equality in (3.5). Suppose instead $y_2 = 0$. Then, (3.6) implies $y_1 = 0$ and $x_1 = -\mu$, $x_2 = \mu\nu$. Since $\mu \geq 0$ and $\|\nu\| \leq 1$, we have $\|x_2\| = \mu\|\nu\| \leq \mu = -x_1$. Thus $x_1 + \|x_2\| \leq 0$, implying $[\lambda_2]_+ = 0$. Since $[\lambda_1]_+ = 0$, this proves the first equality in (3.5). □

A corollary of Proposition 3.3 is the third part of Property 2.2. Another corollary is

$$|x| \succeq_{\mathcal{K}^n} x \qquad \forall x \in \Re^n.$$

Using this corollary, we now prove the following proposition, which establishes some order-preserving properties of x^2 and L_x^2 relative to $\mathcal{K}^n$ and $\mathcal{S}^n$, respectively. These properties, though difficult to prove, are crucial to some of our subsequent analyses. We speculate that these properties may be useful in other contexts.

PROPOSITION 3.4. *For any x, y in $\Re^n$ and any $w \succ_{\mathcal{K}^n} 0$, we have*

$$w^2 \succ_{\mathcal{K}^n} x^2 + y^2 \quad \Longrightarrow \quad L_w^2 \succ L_x^2 + L_y^2, \tag{3.8}$$

$$w^2 \succ_{\mathcal{K}^n} x^2 \quad \Longrightarrow \quad w \succ_{\mathcal{K}^n} x. \tag{3.9}$$

Moreover, the implications (3.8) *and* (3.9) *remain true when "$\succ$" is replaced by "$\succeq$" everywhere.*

Proof. First, we prove (3.8) for the case in which w has the form

$$w = (x^2 + y^2 + \delta e)^{1/2} \tag{3.10}$$

for some $\delta > 0$. Fix any $x = (x_1, x_2)$, $y = (y_1, y_2)$ in $\Re \times \Re^{n-1}$, and any $\delta > 0$. Let $w = (x^2 + y^2 + \delta e)^{1/2}$. Since $w^2 = x^2 + y^2 + \delta e \succ_{\mathcal{K}^n} 0$, we have $w \succ_{\mathcal{K}^n} 0$. Moreover, (2.3) yields

$$w = (s, w_2) \quad \text{with} \quad w_2 = (x_1 x_2 + y_1 y_2)/s, \tag{3.11}$$

where

$$s = \sqrt{\|x\|^2 + \|y\|^2 + \delta + \sqrt{(\|x\|^2 + \|y\|^2 + \delta)^2 - 4\|x_1 x_2 + y_1 y_2\|^2}}/\sqrt{2}.$$

Since $\delta > 0$, we have $s > 0$. Since $w \succ_{\mathcal{K}^n} 0$, we have from (3.11) that $\|w_2\| < s$. Since $w^2 - x^2 - y^2 = \delta e \succ_{\mathcal{K}^n} 0$, we have

$$L_{w^2} - L_{x^2} - L_{y^2} \succ 0.$$

Direct calculation using (3.11) shows that $L_{w^2} - L_{x^2} - L_{y^2} = (\|w\|^2 - \|x\|^2 - \|y\|^2)I$, so it must be that

$$\|x\|^2 + \|y\|^2 < \|w\|^2. \tag{3.12}$$

Using (3.11), the inequality (3.12) expands out to

$$x_1^2 + \|x_2\|^2 + y_1^2 + \|y_2\|^2 < s^2 + x_1^2\|x_2\|^2/s^2 + y_1^2\|y_2\|^2/s^2 + 2x_1 y_1 x_2^T y_2/s^2,$$

which, upon multiplying both sides by s^2 and rearranging terms, can be rewritten equivalently as

$$\|y_1 x_2 - x_1 y_2\|^2 < (s^2 - x_1^2 - y_1^2)(s^2 - \|x_2\|^2 - \|y_2\|^2).$$

Thus, either both $s^2 - x_1^2 - y_1^2$ and $s^2 - \|x_2\|^2 - \|y_2\|^2$ are positive or both are negative. If both are negative, then we would have $x_1^2 + y_1^2 > s^2$ and $\|x_2\|^2 + \|y_2\|^2 > s^2$, contradicting the fact that $\|x\|^2 + \|y\|^2 < \|w\|^2 = s^2 + \|w_2\|^2 < 2s^2$ (recall (3.12) and $\|w_2\| < s$). Thus, we must instead have

$$x_1^2 + y_1^2 < s^2 \quad \text{and} \quad \|x_2\|^2 + \|y_2\|^2 < s^2. \tag{3.13}$$

By direct calculation using (3.11), we have

$$L_w^2 - L_x^2 - L_y^2 = \operatorname{diag}\left\{\|w\|^2 - \|x\|^2 - \|y\|^2, (s^2 - x_1^2 - y_1^2)I + w_2 w_2^T - x_2 x_2^T - y_2 y_2^T\right\}.$$

Our goal is to show that this matrix is positive definite. In view of (3.12), it suffices to prove

$$(s^2 - x_1^2 - y_1^2)I + w_2 w_2^T - x_2 x_2^T - y_2 y_2^T \succ 0. \tag{3.14}$$

For any $d \in \Re^{n-1}$, we have from (3.11) that

$$\begin{aligned}
& d^T(w_2 w_2^T - x_2 x_2^T - y_2 y_2^T)d \\
&= (w_2^T d)^2 - (x_2^T d)^2 - (y_2^T d)^2 \\
&= (x_1 x_2^T d + y_1 y_2^T d)^2/s^2 - (x_2^T d)^2 - (y_2^T d)^2 \\
&= \left(x_1^2(x_2^T d)^2 + y_1^2(y_2^T d)^2 + 2(y_1 x_2^T d)(x_1 y_2^T d)\right)/s^2 - (x_2^T d)^2 - (y_2^T d)^2 \\
&\le \left(x_1^2(x_2^T d)^2 + y_1^2(y_2^T d)^2 + (y_1 x_2^T d)^2 + (x_1 y_2^T d)^2\right)/s^2 - (x_2^T d)^2 - (y_2^T d)^2 \\
&= \left(x_1^2 + y_1^2 - s^2\right)\left((x_2^T d)^2 + (y_2^T d)^2\right)/s^2,
\end{aligned}$$

where the inequality uses the fact that $2\alpha\beta \le \alpha^2 + \beta^2$ for $\alpha, \beta \in \Re$. By (3.13), the right-hand side is always nonpositive, which implies that $0 \succeq w_2 w_2^T - x_2 x_2^T - y_2 y_2^T$ and hence

$$w_2 w_2^T - x_2 x_2^T - y_2 y_2^T \succeq \operatorname{tr}(w_2 w_2^T - x_2 x_2^T - y_2 y_2^T)I = (\|w_2\|^2 - \|x_2\|^2 - \|y_2\|^2)I.$$

This in turn implies

$$(s^2 - x_1^2 - y_1^2)I + w_2 w_2^T - x_2 x_2^T - y_2 y_2^T \succeq (s^2 - x_1^2 - y_1^2 + \|w_2\|^2 - \|x_2\|^2 - \|y_2\|^2)I$$
$$= (\|w\|^2 - \|x\|^2 - \|y\|^2)I.$$

Using (3.12), this proves (3.14), as desired.

We have proven that the implication (3.8) holds true for any x, y in $\Re^n$ and any w of the form (3.10) for some $\delta > 0$. We will now use this result to prove (3.8) for any x, y in $\Re^n$ and any $w \succ_{\mathcal{K}^n} 0$. Suppose $w^2 \succ_{\mathcal{K}^n} x^2 + y^2$. Then, there exists some $\delta > 0$ such that

$$w^2 \succ_{\mathcal{K}^n} x^2 + y^2 + 2\delta e.$$

Let $\bar{w} = (x^2 + y^2 + \delta e)^{1/2}$. Then $w^2 \succ_{\mathcal{K}^n} \bar{w}^2 + \delta e$, and so $z = (w^2 - \bar{w}^2 - \delta e)^{1/2}$ is defined and $w = (\bar{w}^2 + z^2 + \delta e)^{1/2}$. Thus,

$$\bar{w}^2 \succ_{\mathcal{K}^n} x^2 + y^2, \qquad w^2 \succ_{\mathcal{K}^n} \bar{w}^2 + z^2.$$

Since $\bar{w}$ has the form (3.10) with "w" replaced by "$\bar{w}$", (3.8) holds true with this replacement. Similarly, since w has the form (3.10) with "x", "y" replaced by "$\bar{w}$", "z", respectively, (3.8) holds true with this replacement. Thus, the above relations imply

$$L_{\bar{w}}^2 \succ L_x^2 + L_y^2, \qquad L_w^2 \succ L_{\bar{w}}^2 + L_z^2.$$

These two relations combine to yield $L_w^2 \succ L_x^2 + L_y^2$. Thus (3.8) holds true.

We now use (3.8) to prove (3.9) for any x, y in $\Re^n$ and any $w \succ_{\mathcal{K}^n} 0$. Suppose $w^2 \succ_{\mathcal{K}^n} x^2$. Let $y = 0$. Then $w^2 \succ_{\mathcal{K}^n} x^2 + y^2 = |x|^2 + y^2$ and, by (3.8), we have $L_w^2 \succ L_{|x|}^2 + L_y^2$ and hence $L_w^2 - L_{|x|}^2 \succ 0$. Since $w \succ_{\mathcal{K}^n} 0$ and $|x| \succeq_{\mathcal{K}^n} 0$ so that $L_w \succ 0$ and $L_{|x|} \succeq 0$, we also have $L_w + L_{|x|} \succ 0$ and, in particular, $(L_w + L_{|x|})^{1/2}$ is defined and invertible. Finally, we have

$$(L_w - L_{|x|})(L_w + L_{|x|}) = (L_w^2 - L_{|x|}^2) + (L_w L_{|x|} - L_{|x|} L_w) \succ 0, \tag{3.15}$$

where the last step follows from $L_w^2 - L_{|x|}^2 \succ 0$ and the fact that $L_w L_{|x|} - L_{|x|} L_w$ is antisymmetric. Thus, the matrix $(L_w - L_{|x|})(L_w + L_{|x|})$ is positive definite, though not necessarily symmetric. Since $(L_w + L_{|x|})^{1/2}$ is invertible, the symmetric matrix $(L_w + L_{|x|})^{1/2}(L_w - L_{|x|})(L_w + L_{|x|})^{1/2}$ has the same eigenvalues as the matrix $(L_w - L_{|x|})(L_w + L_{|x|})$. It follows that the eigenvalues of the latter matrix must be real, and in view of (3.15), positive. As a result, we have $(L_w + L_{|x|})^{1/2}(L_w - L_{|x|})(L_w + L_{|x|})^{1/2} \succ 0$ and hence $L_w - L_{|x|} \succ 0$ or, equivalently, $w \succ_{\mathcal{K}^n} |x|$. Since $|x| \succeq_{\mathcal{K}^n} x$, this yields $w \succ_{\mathcal{K}^n} x$. Thus (3.9) holds true.

Finally, it can be shown by a standard continuity argument that the implications (3.8) and (3.9) remain true when "$\succ$" is replaced by "$\succeq$" everywhere. □

The converses of both (3.8) and (3.9) are false. In particular, take $n = 3$ and let

$$x = (\sqrt{2.5}, 1, 1), \quad y = (0, 0, 0), \quad w = (2\sqrt{2.5}, 2, 0).$$

Then, it can be checked that $w \succ_{\mathcal{K}^n} x \succ_{\mathcal{K}^n} 0$ and $L_w^2 \succ L_x^2 = L_x^2 + L_y^2$, while $w^2 \not\succ_{\mathcal{K}^n} x^2 = x^2 + y^2$. The property that $w \succ_{\mathcal{K}^n} x$ whenever $w \succ_{\mathcal{K}^n} 0$ and $w^2 \succ_{\mathcal{K}^n} x^2$ has a matrix counterpart. In particular, it can be shown that $W \succ X$ whenever

$W \succ 0$ and $W^2 \succ X^2$, where W, X are $n \times n$ real symmetric matrices [21, Lemma 6.1(c)]. The converse of this implication is false, much like its vector counterpart.

By using Proposition 3.4, we have the following lemma on the positive definite property of certain matrices. This lemma, analogous to a result in the semidefinite setting [6, proof of Lemma 6], will be needed to establish the existence of Newton direction when a smoothing approach based on (4.6) is applied to solve the SOCCP (see Proposition 6.2).

LEMMA 3.5. *For any x, y in $\Re^n$ and any $w \succ_{\mathcal{K}^n} 0$, we have*

$$\begin{array}{ll} & w^2 \succ_{\mathcal{K}^n} x^2 + y^2 \\ \Longrightarrow & (L_w - L_x)(L_w - L_y) \succ 0, \quad L_w - L_x \succ 0, \quad L_w - L_y \succ 0. \end{array} \tag{3.16}$$

Moreover, (3.16) *remains true when "$\succ$" is replaced by "$\succeq$" everywhere.*

Proof. Fix any x, y in $\Re^n$ and any $w \succ_{\mathcal{K}^n} 0$. Suppose $w^2 \succ_{\mathcal{K}^n} x^2 + y^2$. Then, $w^2 \succ_{\mathcal{K}^n} x^2$ and $w^2 \succ_{\mathcal{K}^n} y^2$, and thus it follows from Proposition 3.4 that

$$L_w^2 \succ L_x^2 + L_y^2, \quad L_w \succ L_x, \quad L_w \succ L_y.$$

Thus, it suffices to prove that $(L_w - L_x)(L_w - L_y) \succ 0$. Since the matrix $(L_w - L_x)(L_w - L_y)$ is not necessarily symmetric, we need to prove that its symmetric part is positive definite. Let S denote the symmetric part of $(L_w - L_x)(L_w - L_y)$. Then

$$\begin{aligned} S &= L_w^2 - \frac{L_x + L_y}{2} L_w - L_w \frac{L_x + L_y}{2} + \frac{L_x L_y + L_y L_x}{2} \\ &= \left(\frac{1}{2} L_w^2 + \frac{1}{2} L_x^2 + \frac{1}{2} L_y^2 - \frac{L_x + L_y}{2} L_w - L_w \frac{L_x + L_y}{2} + \frac{L_x L_y + L_y L_x}{2} \right) \\ &\quad + \frac{1}{2} \left(L_w^2 - L_x^2 - L_y^2 \right) \\ &= \frac{1}{2} \left(L_w - L_x - L_y \right)^2 + \frac{1}{2} \left(L_w^2 - L_x^2 - L_y^2 \right) \succ 0, \end{aligned}$$

where the third equality follows from completing the square, and the last step is due to the fact that $L_w^2 - L_x^2 - L_y^2 \succ 0$.

An analogous argument shows that the implication (3.16) remains true when "$\succ$" is replaced by "$\succeq$" everywhere. □

We note that $(L_w - L_x)(L_w - L_y) \succ 0$ is false if w is replaced more generally by any vector w such that $L_w - L_x \succ 0$ and $L_w - L_y \succ 0$. In other words, the product of two positive definite matrices of the form (2.4) may not be positive definite. Thus, the specific form of w (as it relates to x and y) appears to be crucial.

4. Smoothing functions for the SOCCP. In this and the next section, we use the results of the previous section to define and analyze smoothing functions for SOCCP (1.1). The first class of smoothing functions that we consider is a natural generalization of a proposal of Chen and Mangasarian [4, 5] in the case of NCP:

$$\phi_\mu(x, y) = x - \mu g((x - y)/\mu), \tag{4.1}$$

where $\mu > 0$ and $g \in \mathcal{CM}$. Here $\mathcal{CM}$ denotes the class of functions $g : \Re^n \to \Re^n$ defined by (3.1) with $\hat{g} : \Re \to \Re_+$ a continuously differentiable convex function satisfying $\lim_{\alpha \to -\infty} \hat{g}(\alpha) = 0$, $\lim_{\alpha \to \infty} (\hat{g}(\alpha) - \alpha) = 0$ and $0 < \hat{g}'(\alpha) < 1$ for all $\alpha \in \Re$

(see [22]). A special case of this smoothing function is a proposal of Chen and Harker [3], Kanzow [13], and Smale [19], corresponding to

$$\hat{g}(\alpha) = \left(\sqrt{\alpha^2 + 4} + \alpha\right)/2.$$

Another choice of $\hat{g}$ is obtained by integrating the sigmoid function $\alpha \mapsto 1/(1 + \exp(-\alpha))$ used in neural networks [4]:

$$\hat{g}(\alpha) = \ln(\exp(\alpha) + 1).$$

Using Proposition 3.3, we derive below some simple formulas for ϕ_μ and ϕ_0, and we show that ϕ_0 satisfies (1.5).

PROPOSITION 4.1. *For any $g \in \mathcal{CM}$ and $\mu > 0$, let $\phi_\mu : \Re^n \times \Re^n \to \Re^n$ be the smoothing function given by* (4.1). *Then the following results hold.*

(a) *For any $x = (x_1, x_2), y = (y_1, y_2) \in \Re \times \Re^{n-1}$, $\phi_\mu(x, y)$ has the formula*

$$\phi_\mu(x, y) = x - \mu(\hat{g}(\lambda_1/\mu)u^{(1)} + \hat{g}(\lambda_2/\mu)u^{(2)}), \tag{4.2}$$

where, for $i = 1, 2$,

$$\lambda_i = x_1 - y_1 + (-1)^i \|x_2 - y_2\|, \tag{4.3}$$

$$u^{(i)} = \begin{cases} \frac{1}{2}\left(1, (-1)^i \dfrac{x_2 - y_2}{\|x_2 - y_2\|}\right) & \text{if } x_2 \neq y_2, \\ \frac{1}{2}\left(1, (-1)^i w\right) & \text{if } x_2 = y_2, \end{cases} \tag{4.4}$$

with $w \in \Re^{n-1}$ being an arbitrary vector satisfying $\|w\| = 1$.

(b) *The pointwise limit $\phi_0 = \lim_{\mu \to 0^+} \phi_\mu$ has the formula*

$$\phi_0(x, y) \;=\; x - ([\lambda_1]_+ u^{(1)} + [\lambda_2]_+ u^{(2)}) \;=\; x - [x - y]_+, \tag{4.5}$$

where, for $i = 1, 2$, λ_i and $u^{(i)}$ are given by (4.3) and (4.4), *respectively, with $w \in \Re^{n-1}$ being an arbitrary vector satisfying $\|w\| = 1$. Moreover, ϕ_0 satisfies* (1.5) *with $\mathcal{K} = \mathcal{K}^n$.*

Proof. Part (a) follows from (4.1), (3.1), and (2.5)–(2.7). We prove part (b) below.

Since $\lim_{\alpha \to -\infty} \hat{g}(\alpha) = 0$ and $\lim_{\alpha \to \infty}(\hat{g}(\alpha) - \alpha) = 0$, we have $\lim_{\mu \to 0^+} \mu \hat{g}(\lambda_i/\mu) = [\lambda_i]_+$ for $i = 1, 2$. It then follows from (4.2) that

$$\begin{aligned} \lim_{\mu \to 0^+} \phi_\mu(x, y) &= x - \lim_{\mu \to 0^+} \mu(\hat{g}(\lambda_1/\mu)u^{(1)} + \hat{g}(\lambda_2/\mu)u^{(2)}) \\ &= x - ([\lambda_1]_+ u^{(1)} + [\lambda_2]_+ u^{(2)}) \\ &= x - [x - y]_+, \end{aligned}$$

where λ_i, $i = 1, 2$, are given by (4.3), and the last equality follows from (3.5). This proves (4.5).

Finally, $\mathcal{K}^n$ is a closed convex cone that is self-dual, i.e., $\mathcal{K}^n = \{x \in \Re^n | \langle x, y\rangle \geq 0 \ \forall y \in \mathcal{K}^n\}$. Then, it is known [26, Lemma 2.2] that two vectors x, y in $\Re^n$ satisfy $x = [x - y]_+$ if and only if $x \in \mathcal{K}^n$, $y \in \mathcal{K}^n$, $\langle x, y\rangle = 0$. This together with (4.5) proves (1.5). □

A second type of smoothing function that we consider is a generalization of a proposal of Kanzow [4] in the case of the NCP, based on smoothing the Fischer–Burmeister function:

$$\phi_\mu(x,y) \;=\; x+y-(x^2+y^2+2\mu^2 e)^{1/2}. \tag{4.6}$$

Notice that $\phi_\mu(x,y)=0$ if and only if $x\in \operatorname{int}\mathcal{K}^n$, $y\in \operatorname{int}\mathcal{K}^n$, and $x\cdot y=\mu^2 e$. Analogous to Proposition 4.1, we derive below simple formulas for ϕ_μ and the corresponding ϕ_0, and we show that ϕ_0 satisfies (1.5).

PROPOSITION 4.2. *For any $\mu>0$, let $\phi_\mu:\Re^n\times\Re^n\to\Re^n$ be given by* (4.6). *Then the following results hold.*

(a) *For any $x=(x_1,x_2), y=(y_1,y_2)\in\Re\times\Re^{n-1}$, $\phi_\mu(x,y)$ has the formula*

$$\phi_\mu(x,y)=x+y-\left(\sqrt{\lambda_1}u^{(1)}+\sqrt{\lambda_2}u^{(2)}\right), \tag{4.7}$$

where, for $i=1,2$,

$$\lambda_i=\|x\|^2+\|y\|^2+2\mu^2+2(-1)^i\|x_1x_2+y_1y_2\|, \tag{4.8}$$

$$u^{(i)}=\begin{cases}\frac{1}{2}\left(1,(-1)^i\dfrac{x_1x_2+y_1y_2}{\|x_1x_2+y_1y_2\|}\right) & \text{if } x_1x_2+y_1y_2\neq 0,\\ \frac{1}{2}\left(1,(-1)^i w\right) & \text{if } x_1x_2+y_1y_2=0,\end{cases} \tag{4.9}$$

with $w\in\Re^{n-1}$ being an arbitrary vector satisfying $\|w\|=1$.

(b) *The pointwise limit $\phi_0=\lim_{\mu\to 0^+}\phi_\mu$ has the formula*

$$\phi_0(x,y) \;=\; x+y-\left(\sqrt{\lambda_1}u^{(1)}+\sqrt{\lambda_2}u^{(2)}\right) \;=\; x+y-(x^2+y^2)^{1/2}, \tag{4.10}$$

where, for $i=1,2$,

$$\lambda_i \;=\; \|x\|^2+\|y\|^2+2(-1)^i\|x_1x_2+y_1y_2\| \tag{4.11}$$

and $u^{(i)}$ is given by (4.9), *with $w\in\Re^{n-1}$ being an arbitrary vector satisfying $\|w\|=1$. Moreover, ϕ_0 satisfies* (1.5) *with $\mathcal{K}=\mathcal{K}^n$.*

Proof. Part (a) follows from (4.6), (3.3), and the observation that λ_1, λ_2 given by (4.8) are the spectral values of $x^2+y^2+2\mu^2 e$. We prove part (b) below.

Fix any $x=(x_1,x_2), y=(y_1,y_2)\in\Re\times\Re^{n-1}$. For $i=1,2$, we have that λ_i given by (4.8) converges to λ_i given by (4.11) as $\mu\to 0^+$. Since $u^{(i)}$ given by (4.9) is independent of μ, then $\phi_\mu(x,y)$ given by (4.7) converges to the second term in (4.10), thus proving the first equality in (4.10). Using (3.3) and the observation that λ_1, λ_2 given by (4.11) are the spectral values of x^2+y^2, we obtain the second equality in (4.10).

To show that (1.5) holds, let us first suppose $\langle x,y\rangle=0$, $x\in\mathcal{K}^n$, $y\in\mathcal{K}^n$. Then, by Proposition 2.1, we have $x\cdot y=0$, implying $(x+y)^2=x^2+y^2$ and hence $x+y=(x^2+y^2)^{1/2}$, i.e., $\phi_0(x,y)=0$. Conversely, suppose x and y satisfy $\phi_0(x,y)=0$. Then, $x+y=(x^2+y^2)^{1/2}$, so that, upon squaring both sides, we obtain $x\cdot y=0$. Let $w=(x^2+y^2)^{1/2}$. Then, $w\succeq_{\mathcal{K}^n}0$ and $w^2=x^2+y^2$, implying $w^2\succeq_{\mathcal{K}^n}x^2$ and $w^2\succeq_{\mathcal{K}^n}y^2$. Thus, it follows from Proposition 3.4 that $w\succeq_{\mathcal{K}^n}x$ and $w\succeq_{\mathcal{K}^n}y$. As a result, $x=w-y\succeq_{\mathcal{K}^n}0$ and $y=w-x\succeq_{\mathcal{K}^n}0$, or equivalently, $x,y\in\mathcal{K}^n$. Since $x\cdot y=0$ as shown above, this together with Proposition 2.1 completes the proof. ∎

5. Differential and Lipschitzian properties of smoothing functions. In this section we study differential and Lipschitzian properties of the smoothing functions (4.1) and (4.6) introduced in the previous section.

PROPOSITION 5.1. *For any $\mu > 0$, let $\phi_\mu : \Re^n \times \Re^n \to \Re^n$ be given by either* (4.1) *with $g \in \mathcal{CM}$ or* (4.6). *Then, for any x, $y \in \Re^n$ and any $\mu > \nu > 0$, we have*

$$\rho(\mu - \nu)e \succeq_{\mathcal{K}^n} \phi_\nu(x,y) - \phi_\mu(x,y) \succ_{\mathcal{K}^n} 0, \tag{5.1}$$

$$\rho\mu e \succeq_{\mathcal{K}^n} \phi_0(x,y) - \phi_\mu(x,y) \succ_{\mathcal{K}^n} 0, \tag{5.2}$$

where $\rho = \hat{g}(0)$ if ϕ_μ is given by (4.1), *and $\rho = \sqrt{2}$ if ϕ_μ is given by* (4.6).

Proof. Fix any $x = (x_1, x_2)$, $y = (y_1, y_2)$ in $\Re \times \Re^{n-1}$.

Suppose that ϕ_μ is given by (4.1) with $g \in \mathcal{CM}$. Then, for any $\mu > \nu > 0$, Proposition 4.1(a) yields

$$\phi_\nu(x,y) - \phi_\mu(x,y) = (\mu\hat{g}(\lambda_1/\mu) - \nu\hat{g}(\lambda_1/\nu))u^{(1)} + (\mu\hat{g}(\lambda_2/\mu) - \nu\hat{g}(\lambda_2/\nu))u^{(2)}, \tag{5.3}$$

where λ_i and $u^{(i)}$ are given by (4.3) and (4.4) for $i = 1, 2$. By [22, Lemma 3.1], we have $0 < \mu\hat{g}(\lambda_i/\mu) - \nu\hat{g}(\lambda_i/\nu) \le \hat{g}(0)(\mu - \nu)$ for $i = 1, 2$. This, together with (5.3) and Property 2.2 and $u^{(i)} \succeq_{\mathcal{K}^n} 0$ for $i = 1, 2$, yields

$$\begin{aligned} 0 &\preceq_{\mathcal{K}^n} \phi_\nu(x,y) - \phi_\mu(x,y) \\ &\preceq_{\mathcal{K}^n} \hat{g}(0)(\mu - \nu)u^{(1)} + \hat{g}(0)(\mu - \nu)u^{(2)} \\ &= \hat{g}(0)(\mu - \nu)e. \end{aligned}$$

This proves (5.1) with $\rho = \hat{g}(0)$.

Suppose that ϕ_μ is given by (4.6). Then, for any $\mu > \nu > 0$, Proposition 4.2(a) yields that

$$\begin{aligned} \phi_\nu(x,y) - \phi_\mu(x,y) &= \left(\sqrt{\lambda_1(\mu)} - \sqrt{\lambda_1(\nu)}\right)u^{(1)} + \left(\sqrt{\lambda_2(\mu)} - \sqrt{\lambda_2(\nu)}\right)u^{(2)} \\ &= \frac{2(\mu^2 - \nu^2)}{\sqrt{\lambda_1(\mu)} + \sqrt{\lambda_1(\nu)}}u^{(1)} + \frac{2(\mu^2 - \nu^2)}{\sqrt{\lambda_2(\mu)} + \sqrt{\lambda_2(\nu)}}u^{(2)}, \end{aligned} \tag{5.4}$$

where we define

$$\lambda_i(\mu) \;=\; \|x\|^2 + \|y\|^2 + 2\mu^2 + 2(-1)^i\|x_1x_2 + y_1y_2\|,$$

and $u^{(i)}$ is given by (4.9) for $i = 1, 2$. Since $\mu > \nu > 0$, and so $\mu^2 - \nu^2 > 0$, we obtain from (5.4) and $u^{(i)} \succeq_{\mathcal{K}^n} 0$ for $i = 1, 2$, that

$$\phi_\nu(x,y) - \phi_\mu(x,y) \succ_{\mathcal{K}^n} 0.$$

Moreover, since

$$\lambda_i(\mu) \ge 2\mu^2, \quad \lambda_i(\nu) \ge 2\nu^2, \quad i = 1, 2,$$

we have

$$\frac{2(\mu^2 - \nu^2)}{\sqrt{\lambda_i(\mu)} + \sqrt{\lambda_i(\nu)}} \le \sqrt{2}(\mu - \nu), \quad i = 1, 2.$$

Then, we obtain from (5.4) and Property 2.2 that

$$\begin{aligned} \phi_\nu(x,y) - \phi_\mu(x,y) &\preceq_{\mathcal{K}^n} \sqrt{2}(\mu - \nu)u^{(1)} + \sqrt{2}(\mu - \nu)u^{(2)} \\ &= \sqrt{2}(\mu - \nu)e. \end{aligned}$$

This proves (5.1) with $\rho = \sqrt{2}$.

We have shown for either choice of ϕ_μ that (5.1) holds for any $\mu > \nu > 0$. This implies that $-\phi_\nu$ is monotone in $\nu > 0$ with respect to the partial ordering $\succ_{\mathcal{K}^n}$. Then, taking $\nu \to 0^+$ in (5.1) yields

$$\rho\mu e \succeq_{\mathcal{K}^n} \lim_{\nu \to 0^+} \phi_\nu(x,y) - \phi_\mu(x,y) = \phi_0(x,y) - \phi_\mu(x,y) \succ_{\mathcal{K}^n} 0.$$

This proves (5.2). □

Next, we analyze the differential properties of the smoothing functions given by (4.1) and (4.6). We begin with the following key proposition showing that the smoothness property of $\hat{g}$ is inherited by the corresponding function g associated with an SOC.

PROPOSITION 5.2. *For any $\hat{g} : \Re \to \Re$ that is Fréchet-differentiable (respectively, continuously differentiable), the function $g : \Re^n \to \Re^n$ defined by (3.1) is Fréchet-differentiable (respectively, continuously differentiable), and its Jacobian at $z = (z_1, z_2) \in \Re \times \Re^{n-1}$ is given by the formula $\nabla g(z) = \hat{g}'(z_1)I$ if $z_2 = 0$, and otherwise is given by*

$$\nabla g(z) = \begin{bmatrix} b & c\, z_2^T/\|z_2\| \\ c\, z_2/\|z_2\| & aI + (b-a)z_2 z_2^T/\|z_2\|^2 \end{bmatrix}, \tag{5.5}$$

where

$$a = \frac{\hat{g}(\lambda_2) - \hat{g}(\lambda_1)}{\lambda_2 - \lambda_1}, \quad b = \frac{1}{2}\left(\hat{g}'(\lambda_2) + \hat{g}'(\lambda_1)\right), \quad c = \frac{1}{2}\left(\hat{g}'(\lambda_2) - \hat{g}'(\lambda_1)\right), \tag{5.6}$$

with $\lambda_i = z_1 + (-1)^i\|z_2\|$, $i = 1, 2$. If $\hat{g}'(\alpha) > 0$ for all $\alpha \in \Re$, then $\nabla g(z)$ is positive definite for all $z \in \Re^n$.

Proof. Assume that $\hat{g}$ is Fréchet-differentiable. Fix any $z = (z_1, z_2) \in \Re \times \Re^{n-1}$. First, we consider the case of $z_2 \neq 0$. By (3.1), we have

$$g(z) = \hat{g}(\lambda_1)u^{(1)} + \hat{g}(\lambda_2)u^{(2)}, \tag{5.7}$$

where

$$\lambda_i = z_1 + (-1)^i\|z_2\|, \qquad u^{(i)} = \frac{1}{2}\left(1, (-1)^i \frac{z_2}{\|z_2\|}\right), \qquad i = 1, 2.$$

Using the fact that

$$\nabla_w \left(\frac{w}{\|w\|}\right) = \frac{1}{\|w\|}\left(I - \frac{ww^T}{\|w\|^2}\right) \qquad \forall w \neq 0,$$

we obtain that $u^{(i)}$ is Fréchet-differentiable with respect to z, with

$$\nabla_z u^{(i)} = \frac{(-1)^i}{2\|z_2\|}\begin{bmatrix} 0 & 0 \\ 0 & I - z_2 z_2^T/\|z_2\|^2 \end{bmatrix}, \qquad i = 1, 2. \tag{5.8}$$

Similarly, λ_i is Fréchet-differentiable with respect to z, with

$$\nabla_z \lambda_i = 2u^{(i)}, \qquad i = 1, 2. \tag{5.9}$$

Since $\hat{g}$ is Fréchet-differentiable, using the chain rule and product rule for differentiation, we obtain from (5.7) that

$$\begin{aligned}\nabla g(z) &= \hat{g}(\lambda_1)\nabla_z u^{(1)} + u^{(1)}(\nabla_z \hat{g}(\lambda_1))^T + \hat{g}(\lambda_2)\nabla_z u^{(2)} + u^{(2)}(\nabla_z \hat{g}(\lambda_2))^T \\ &= \hat{g}(\lambda_1)\nabla_z u^{(1)} + \hat{g}(\lambda_2)\nabla_z u^{(2)} + u^{(1)}\hat{g}'(\lambda_1)(\nabla_z \lambda_1)^T + u^{(2)}\hat{g}'(\lambda_2)(\nabla_z \lambda_2)^T \\ &= \frac{\hat{g}(\lambda_2) - \hat{g}(\lambda_1)}{2\|z_2\|} \begin{bmatrix} 0 & 0 \\ 0 & I - z_2 z_2^T/\|z_2\|^2 \end{bmatrix} \\ &\quad + 2\hat{g}'(\lambda_1)u^{(1)}(u^{(1)})^T \;+\; 2\hat{g}'(\lambda_2)u^{(2)}(u^{(2)})^T \\ &= a \begin{bmatrix} 0 & 0 \\ 0 & I - z_2 z_2^T/\|z_2\|^2 \end{bmatrix} + 2\hat{g}'(\lambda_1)u^{(1)}(u^{(1)})^T + 2\hat{g}'(\lambda_2)u^{(2)}(u^{(2)})^T,\end{aligned}$$

where the third equality uses (5.8) and (5.9), and the last equality uses the definition of a and the fact that $\lambda_2 - \lambda_1 = 2\|z_2\|$. Notice that

$$u^{(i)}(u^{(i)})^T = \frac{1}{4}\begin{bmatrix} 1 & (-1)^i z_2^T/\|z_2\| \\ (-1)^i z_2/\|z_2\| & z_2 z_2^T/\|z_2\|^2 \end{bmatrix}, \quad i = 1, 2.$$

Substituting this into the previous expression and simplifying yields

$$\begin{aligned}\nabla g(z) &= a \begin{bmatrix} 0 & 0 \\ 0 & I - z_2 z_2^T/\|z_2\|^2 \end{bmatrix} + \begin{bmatrix} b & c\, z_2^T/\|z_2\| \\ c\, z_2/\|z_2\| & b\, z_2 {z_2}^T/\|z_2\|^2 \end{bmatrix} \\ &= \begin{bmatrix} b & c\, z_2^T/\|z_2\| \\ c\, z_2/\|z_2\| & aI + (b-a) z_2 z_2^T/\|z_2\|^2 \end{bmatrix},\end{aligned}$$

where we have used the definitions of b and c. This proves the proposition for the case of $z_2 \neq 0$.

Next we consider the case of $z_2 = 0$. We calculate the Jacobian matrix ∇g at z by perturbing z by $\Delta z = (\Delta z_1, \Delta z_2) \in \Re \times \Re^{n-1}$ and considering the resulting variation in g. First consider the case of $\Delta z_2 \neq 0$. We have from (3.1) that

$$g(z + \Delta z) = \hat{g}(\lambda + \Delta\lambda_1)u^{(1)} + \hat{g}(\lambda + \Delta\lambda_2)u^{(2)}, \qquad g(z) = \hat{g}(\lambda)u^{(1)} + \hat{g}(\lambda)u^{(2)},$$

where we define

$$\lambda = z_1, \quad \Delta\lambda_i = \Delta z_1 + (-1)^i \|\Delta z_2\|, \quad u^{(i)} = \frac{1}{2}\left(1, (-1)^i \frac{\Delta z_2}{\|\Delta z_2\|}\right), \quad i = 1, 2.$$

We also have from the Taylor expansion of $\hat{g}$ at λ that

$$\hat{g}(\lambda + \Delta\lambda_i) - \hat{g}(\lambda) = \hat{g}'(\lambda)\Delta\lambda_i + o(\Delta\lambda_i) = \hat{g}'(\lambda)\Delta\lambda_i + o(\|\Delta z\|),$$

where $o(\cdot)$ is the usual "little o" notation, i.e., $\beta = o(\alpha)$ means $\beta/\alpha \to 0$ as $\alpha \to 0$. Combining the preceding two relations, we obtain

$$\begin{aligned} g(z + \Delta z) - g(z) &= (\hat{g}(\lambda + \Delta\lambda_1)u^{(1)} + \hat{g}(\lambda + \Delta\lambda_2)u^{(2)}) - (\hat{g}(\lambda)u^{(1)} + \hat{g}(\lambda)u^{(2)}) \\ &= (\hat{g}(\lambda + \Delta\lambda_1) - \hat{g}(\lambda))u^{(1)} + (\hat{g}(\lambda + \Delta\lambda_2) - \hat{g}(\lambda))u^{(2)} \\ &= \hat{g}'(\lambda)\Delta\lambda_1 u^{(1)} + \hat{g}'(\lambda)\Delta\lambda_2 u^{(2)} + o(\|\Delta z\|) \\ &= \frac{\hat{g}'(\lambda)}{2}\left(\Delta\lambda_2 + \Delta\lambda_1, (\Delta\lambda_2 - \Delta\lambda_1)\frac{\Delta z_2}{\|\Delta z_2\|}\right) + o(\|\Delta z\|) \\ &= \hat{g}'(\lambda)\,(\Delta z_1, \Delta z_2) + o(\|\Delta z\|) \\ &= \hat{g}'(\lambda)\Delta z + o(\|\Delta z\|), \end{aligned}$$

where the fourth equality follows from the definition of $u^{(i)}$, and the fifth equality uses the fact that

$$\Delta\lambda_2 + \Delta\lambda_1 = 2\Delta z_1, \qquad \Delta\lambda_2 - \Delta\lambda_1 = 2\|\Delta z_2\|.$$

In the case of $\Delta z_2 = 0$, the same argument applies except that $\Delta z_2/\|\Delta z_2\|$ is replaced by any $w \in \Re^{n-1}$ satisfying $\|w\| = 1$. This shows that g is Fréchet-differentiable at z and

$$\nabla g(z) = \hat{g}'(\lambda) I.$$

Assume that $\hat{g}$ is continuously differentiable. It is readily seen from (5.5) and (5.6) that ∇g is continuous (entrywise) at every z with $z_2 \neq 0$. To argue the continuity of ∇g at every z with $z_2 = 0$, fix any $x = (x_1, 0) \in \Re^n$ and consider the limit of $\nabla g(z)$ as $z = (z_1, z_2)$ approaches x. Let $\lambda_i = z_1 + (-1)^i\|z_2\|$, $i = 1, 2$, and let a, b, c be given by (5.6). Then, since $\lim_{z\to x} \lambda_1 = \lim_{z\to x} \lambda_2 = x_1$, we have

$$\lim_{z\to x} a = \lim_{z\to x} \frac{\hat{g}(\lambda_2) - \hat{g}(\lambda_1)}{\lambda_2 - \lambda_1} = \hat{g}'(x_1), \quad \lim_{z\to x} b \;=\; \hat{g}'(x_1), \quad \lim_{z\to x} c \;=\; 0.$$

Thus, taking the limit in (5.5) as $z \to x$ yields

$$\lim_{z\to x} \nabla g(z) = \hat{g}'(x_1) I = \nabla g(x).$$

Last, assume that $\hat{g}'(\alpha) > 0$ for all $\alpha \in \Re$. Fix any $z = (z_1, z_2)$. We will show that $\nabla g(z)$ is positive definite. If $z_2 \neq 0$, then $\nabla g(z)$ is given by (5.5) with a, b, c given by (5.6) and $\lambda_i = z_1 + (-1)^i\|z_2\|$, $i = 1, 2$. Since $b > 0$, it suffices to show that the Schur complement of b in $\nabla g(z)$ is positive definite. This Schur complement has the form

$$aI + (b-a)\frac{z_2 z_2^T}{\|z_2\|^2} - \frac{c^2 z_2 z_2^T}{b\|z_2\|^2} = a\left(I - \frac{z_2 z_2^T}{\|z_2\|^2}\right) + b\left(1 - \frac{c^2}{b^2}\right)\frac{z_2 z_2^T}{\|z_2\|^2}.$$

Since $a > 0$, $b > 0$, and $b > c \geq 0$, the right-hand side is a linear positive combination of the matrices $I - z_2 z_2^T/\|z_2\|^2$ and $z_2 z_2^T/\|z_2\|^2$, and thus it is positive definite. If $z_2 = 0$, then $\nabla g(z) = \hat{g}'(z_1) I$, which is positive definite due to $\hat{g}'(z_1) > 0$. □

Notice that the Jacobian $\nabla g(z)$ is symmetric. Moreover, it can be seen that Proposition 5.2 still holds if "$\hat{g} : \Re \to \Re$" is replaced by "$\hat{g} : \Re_{++} \to \Re$" and "$z \in \Re^n$" is replaced by "$z \in \operatorname{int} \mathcal{K}^n$." By using Proposition 5.2 and this observation, we obtain the following differentiability results for the smoothing functions given by (4.1) and (4.6).

COROLLARY 5.3. *For any $\mu > 0$ and any $g \in \mathcal{CM}$, the Chen–Mangasarian smoothing function $\phi_\mu : \Re^n \times \Re^n \to \Re^n$ given by* (4.1) *is continuously differentiable, and its Jacobian is given by*

$$\nabla\phi_\mu(x, y) = \begin{bmatrix} I - \nabla g(z) \\ \nabla g(z) \end{bmatrix},$$

where $z = (x - y)/\mu$ and $\nabla g(z)$ has the formula in Proposition 5.2.

COROLLARY 5.4. *For any $\mu > 0$, the smoothed Fischer–Burmeister function $\phi_\mu : \Re^n \times \Re^n \to \Re^n$ given by* (4.6) *is continuously differentiable, and its Jacobian can be written as*

$$\nabla\phi_\mu(x, y) = \begin{bmatrix} I - 2L_x \nabla g(z) \\ I - 2L_y \nabla g(z) \end{bmatrix} = \begin{bmatrix} I - L_x L_w^{-1} \\ I - L_y L_w^{-1} \end{bmatrix},$$

where $z = x^2 + y^2 + 2\mu^2 e$, $w = z^{1/2}$, *and* $\nabla g(z)$ *has the formula in Proposition* 5.2 *with* $\hat{g}(\alpha) = \alpha^{1/2}$ *for all* $\alpha \in \Re_{++}$.

Proof. Fix any $\mu > 0$. Define the mapping $Z : \Re^n \times \Re^n \to \Re^n$ by $Z(x,y) = x^2 + y^2 + 2\mu^2 e$. Direct calculation yields

$$Z(x,y) = (\|x\|^2 + \|y\|^2 + 2\mu^2, 2(x_1 x_2 + y_1 y_2)),$$

from which we see that Z is continuously differentiable and that

$$\nabla_x Z(x,y) = 2\begin{bmatrix} x_1 & x_2^T \\ x_2 & x_1 I \end{bmatrix} = 2L_x, \qquad \nabla_y Z(x,y) = 2\begin{bmatrix} y_1 & y_2^T \\ y_2 & y_1 I \end{bmatrix} = 2L_y. \tag{5.10}$$

Also, since $x^2 \succeq_{\mathcal{K}^n} 0$, $y^2 \succeq_{\mathcal{K}^n} 0$, $e \succ_{\mathcal{K}^n} 0$, we have that $Z(x,y) \succ_{\mathcal{K}^n} 0$ for all $(x,y) \in \Re^n \times \Re^n$, i.e., Z maps $\Re^n \times \Re^n$ into int $\mathcal{K}^n$. Since $\hat{g}(\cdot) = (\cdot)^{1/2}$ is continuously differentiable on $\Re_{++}$, it follows from Proposition 5.2 and the subsequent remark that g is continuously differentiable on int $\mathcal{K}^n$. Thus, by the chain rule for differentiation, the composite mapping $\psi_\mu = g \circ Z$ is continuously differentiable and

$$\nabla_x \psi_\mu(x,y) = \nabla_x Z(x,y) \nabla g(Z(x,y)), \qquad \nabla_y \psi_\mu(x,y) = \nabla_y Z(x,y) \nabla g(Z(x,y)).$$

This together with (5.10) and the fact that

$$\nabla g(z) = \frac{1}{2} L_w^{-1} \qquad \text{with} \qquad w = z^{1/2} \tag{5.11}$$

yields the desired Jacobian formula.

It remains to verify (5.11). Although this can be verified by using (5.5), (5.6), and $\hat{g}(\alpha) = \alpha^{1/2}$ to calculate $\nabla g(z)^{-1}$ explicitly and by comparing it with L_w, the algebra becomes quite involved. Instead we will use the following simpler argument: Since $z \succ_{\mathcal{K}^n} 0$, there exists scalar $\delta > 0$ such that $z + d \succ_{\mathcal{K}^n} 0$ for all $d \in \Re^n$ with $\|d\| < \delta$. For any such d we have, upon letting $u = (z+d)^{1/2} - w$ and using $w = z^{1/2}$, that

$$d = (w+u)^2 - w^2 = 2w \cdot u + u^2 = 2L_w u + u^2.$$

Since $w \succ_{\mathcal{K}^n} 0$, L_w is invertible, applying L_w^{-1} to both sides yields

$$L_w^{-1} d = 2u + L_w^{-1} u^2.$$

This shows that $\|u\|$ is in the order of $\|d\|$ whenever $\|d\|$ is sufficiently small. Moreover,

$$g(z+d) - g(z) = u = \frac{1}{2} L_w^{-1} d - \frac{1}{2} L_w^{-1} u^2,$$

and it follows that $\nabla g(z) = \frac{1}{2} L_w^{-1}$. □

For simplicity, we have considered in this section only the case of a single SOC. Extension of the results in this section to the case of a direct product of SOCs, i.e., (1.2), is straightforward and is omitted.

6. A smoothing approach to solving the SOCCP. In this section, we study the use of smoothing functions in a smoothing approach to solving the SOCCP (1.1), where $F : \Re^n \times \Re^n \times \Re^\ell \to \Re^n \times \Re^\ell$ is continuously differentiable and $\mathcal{K}$ is given by (1.2) with $m, n_1, \ldots, n_m \geq 1$ and $n_1 + \cdots + n_m = n$. Our study bears some similarity

with that given in [6] for the SDCP, though our analysis is quite different due to the different structures of the SOC.

In what follows, we write

$$x = (x_1, \ldots, x_m) \in \Re^n$$

to implicitly mean $x_i \in \Re^{n_i}$, $i = 1, \ldots, m$. Then, using the direct product structure of (1.2), the SOCCP (1.1) may be written equivalently as

$$\langle x_i, y_i \rangle = 0, \quad x_i \in \mathcal{K}^{n_i}, \quad y_i \in \mathcal{K}^{n_i}, \quad i = 1, \ldots, m, \qquad F(x, y, \zeta) = 0.$$

We choose, for each $i \in \{1, \ldots, m\}$, a continuously differentiable function $\phi_\mu^i : \Re^{n_i} \times \Re^{n_i} \to \Re^{n_i}$, parameterized by $\mu > 0$, such that the pointwise limit $\phi_0^i(x_i, y_i) = \lim_{\mu \to 0^+} \phi_\mu^i(x_i, y_i)$ satisfies

$$\langle x_i, y_i \rangle = 0, \quad x_i \in \mathcal{K}^{n_i}, \quad y_i \in \mathcal{K}^{n_i} \quad \Longleftrightarrow \quad \phi_0^i(x_i, y_i) = 0.$$

Then the function $\phi_\mu : \Re^n \times \Re^n \to \Re^n$ defined by

$$\phi_\mu(x, y) = (\phi_\mu^1(x_1, y_1), \ldots, \phi_\mu^m(x_m, y_m)) \quad \forall x, y \in \Re^n \tag{6.1}$$

is continuously differentiable and satisfies (1.5). The system of smooth equations (1.6) decomposes accordingly.

In the smoothing approach to solving (1.1), we fix $\mu > 0$ and solve (1.6) approximately by applying a few Newton steps; then we decrease μ and repeat this iteration. For the Newton direction to be well defined and unique, it is essential that the Jacobian matrix of the left-hand side of (1.6), viewed as a mapping from $\Re^{2n+\ell}$ to $\Re^{2n+\ell}$, be invertible. To this end, consider the following rank and monotonicity assumptions on $\nabla F(x, y, \zeta)$:

$$\operatorname{rank} \nabla_\zeta F(x, y, \zeta) = \ell, \tag{6.2}$$

$$(u, v, \Delta) \in \Re^n \times \Re^n \times \Re^\ell, \quad \nabla F(x, y, \zeta)^T (u, v, \Delta) = 0 \quad \Longrightarrow \quad u^T v \geq 0. \tag{6.3}$$

These assumptions are reasonable and, in particular, hold for convex programs with linear and SOC constraints (see the next section). When $F(x, y, \zeta)$ has the form (1.3), the above assumptions reduce to the assumption that $\nabla F_0(x)$ is positive semidefinite. The following proposition shows that if (6.2) and (6.3) hold and each ϕ_μ^i is given by (4.1), then the aforementioned Jacobian is invertible for any $\mu > 0$, and thus the corresponding Newton direction is well defined.

PROPOSITION 6.1. *Assume that $F : \Re^n \times \Re^n \times \Re^\ell \to \Re^n \times \Re^\ell$ is continuously differentiable. Let $\phi_\mu : \Re^n \times \Re^n \to \Re^n$ be defined by* (6.1) *with*

$$\phi_\mu^i(x_i, y_i) = x_i - \mu g((x_i - y_i)/\mu) \quad \forall x_i, y_i \in \Re^{n_i}$$

and $g \in \mathcal{CM}$. Then, for each $\mu > 0$ and $(x, y, \zeta) \in \Re^n \times \Re^n \times \Re^\ell$ satisfying (6.2) *and* (6.3)*, the matrix $J_\mu(x, y, \zeta)$ is invertible, where*

$$J_\mu(x, y, \zeta) = \begin{bmatrix} I - \operatorname{diag}\{\nabla g(z_i)\}_{i=1}^m & \operatorname{diag}\{\nabla g(z_i)\}_{i=1}^m & 0 \\ \nabla_x F(x, y, \zeta)^T & \nabla_y F(x, y, \zeta)^T & \nabla_\zeta F(x, y, \zeta)^T \end{bmatrix}, \tag{6.4}$$

$z_i = (x_i - y_i)/\mu$, and $\nabla g(z_i) \in \Re^{n_i \times n_i}$ is specified as in Proposition 5.2.

Proof. Fix any $\mu > 0$ and any $(x, y, \zeta) \in \Re^n \times \Re^n \times \Re^\ell$ satisfying (6.2) and (6.3). By using (6.1) and Corollary 5.3, it is readily verified that $J_\mu(x, y, \zeta)$ given by (6.4) is indeed the Jacobian matrix for the left-hand side of (1.6) with ϕ_μ as defined. We show below that $J_\mu(x, y, \zeta)$ is invertible.

Let $(u, v, \Delta) \in \Re^n \times \Re^n \times \Re^\ell$ be a vector in the null space of $J_\mu(x, y, \zeta)$. We will show that $u = v = 0$, $\Delta = 0$. By (6.4),

$$(I - B)\, u + Bv = 0, \qquad \nabla F(x, y, \zeta)^T (u, v, \Delta) = 0, \tag{6.5}$$

where for simplicity we denote $B = \mathrm{diag}\{B_1, B_2, \ldots, B_m\}$ and $B_i = \nabla g(z_i)$, with $z_i = (x_i - y_i)/\mu$. In what follows, we write $z_i = (z_{i1}, z_{i2}) \in \Re \times \Re^{n_i - 1}$ for $i = 1, \ldots, m$.

We claim that $I \succ B \succ 0$. To see this, fix any $i \in \{1, \ldots, m\}$. Suppose $z_{i2} = 0$. Then Proposition 5.2 yields $B_i = \hat{g}'(z_{i1})I$. Since $g \in \mathcal{CM}$ so that $1 > \hat{g}'(z_{i1}) > 0$, this implies $I \succ B_i \succ 0$. Suppose $z_{i2} \neq 0$. Then Proposition 5.2 yields that B_i is symmetric and

$$I - B_i = \begin{bmatrix} 1 - b_i & -\dfrac{c_i {z_{i2}}^T}{\|z_{i2}\|} \\ -\dfrac{c_i z_{i2}}{\|z_{i2}\|} & (1 - a_i)I + (a_i - b_i)\dfrac{z_{i2}{z_{i2}}^T}{\|z_{i2}\|^2} \end{bmatrix},$$

where

$$a_i = \frac{\hat{g}(\lambda_{i2}) - \hat{g}(\lambda_{i1})}{\lambda_{i2} - \lambda_{i1}}, \quad b_i = \frac{1}{2}(\hat{g}'(\lambda_{i2}) + \hat{g}'(\lambda_{i1})), \quad c_i = \frac{1}{2}(\hat{g}'(\lambda_{i2}) - \hat{g}'(\lambda_{i1})),$$

with $\lambda_{ij} = z_{i1} + (-1)^j \|z_{i2}\|$ for $j = 1, 2$. Since $g \in \mathcal{CM}$, we have $1 > \hat{g}'(\alpha) > 0$ for all $\alpha \in \Re$. Also, $\lambda_{i2} > \lambda_{i1}$ and the convexity of $\hat{g}$ imply $\hat{g}'(\lambda_{i2}) \geq \hat{g}'(\lambda_{i1})$. Thus,

$$1 > a_i > 0, \quad b_i > c_i \geq 0, \quad 1 - b_i > c_i \geq 0. \tag{6.6}$$

Since $0 < b_i < 1$, it suffices to show that the Schur complement of b_i in $I - B_i$ is positive definite (see [12]). This Schur complement has the form

$$\begin{aligned} &(1 - a_i)I + (a_i - b_i)\frac{z_{i2}{z_{i2}}^T}{\|z_{i2}\|^2} - \frac{c_i^2 z_{i2}{z_{i2}}^T}{(1 - b_i)\|z_{i2}\|^2} \\ &= (1 - a_i)\left(I - \frac{z_{i2}{z_{i2}}^T}{\|z_{i2}\|^2}\right) + (1 - b_i)\left(1 - \frac{c_i^2}{(1 - b_i)^2}\right)\frac{z_{i2}{z_{i2}}^T}{\|z_{i2}\|^2}, \end{aligned}$$

which, by (6.6), is a linear positive combination of $I - z_{i2}{z_{i2}}^T/\|z_{i2}\|^2$ and $z_{i2}{z_{i2}}^T/\|z_{i2}\|^2$ and hence is positive definite. Thus, $I \succ B_i$. Also, Proposition 5.2 shows that $B_i \succ 0$. Thus, we have shown that $I \succ B_i \succ 0$ for all i or, equivalently, $I \succ B \succ 0$.

Since B is invertible, multiplying both sides of the first equation in (6.5) by $u^T B^{-1}$ yields

$$u^T(B^{-1} - I)u + u^T v = 0. \tag{6.7}$$

Since B is symmetric and $I \succ B \succ 0$, it follows that B^{-1} is symmetric and $B^{-1} \succ I$, and so $u^T(B^{-1} - I)u \geq 0$. Also, by the assumption (6.3), we have from the second equation in (6.5) that $u^T v \geq 0$. Thus, (6.7) implies $u^T(B^{-1} - I)u = 0$ and hence $u = 0$. Since B is invertible, the first equation in (6.5) then implies $v = 0$. Since $\nabla_\zeta F(x, y, \zeta)$ has rank ℓ, the second equation in (6.5) then implies $\Delta = 0$. Thus,

the null space of $J_\mu(x,y,\zeta)$ comprises only the origin, and thus $J_\mu(x,y,\zeta)$ is invertible. □

The next proposition shows that, when the smoothed Fischer–Burmeister function is used, the Jacobian matrix for the left-hand side of (1.6) is invertible. The proof uses Corollary 5.4 and Lemma 3.5.

PROPOSITION 6.2. *Assume that* $F : \Re^n \times \Re^n \times \Re^\ell \to \Re^n \times \Re^\ell$ *is continuously differentiable. Let* $\phi_\mu : \Re^n \times \Re^n \to \Re^n$ *be defined by* (6.1) *with*

$$\phi_\mu^i(x_i, y_i) = x_i + y_i - (x_i^2 + y_i^2 + 2\mu^2 e)^{1/2} \quad \forall x_i, y_i \in \Re^{n_i}.$$

Then, for each $\mu > 0$ *and* $(x,y,\zeta) \in \Re^n \times \Re^n \times \Re^\ell$ *satisfying* (6.2) *and* (6.3), *the matrix* $J_\mu(x,y,\zeta)$ *is invertible, where*

$$J_\mu(x,y,\zeta) = \begin{bmatrix} \mathrm{diag}\{\nabla_{x_i}\phi_\mu^i(x_i,y_i)^T\}_{i=1}^m & \mathrm{diag}\{\nabla_{y_i}\phi_\mu^i(x_i,y_i)^T\}_{i=1}^m & 0 \\ \nabla_x F(x,y,\zeta)^T & \nabla_y F(x,y,\zeta)^T & \nabla_\zeta F(x,y,\zeta)^T \end{bmatrix}; \tag{6.8}$$

$\nabla_{x_i}\phi_\mu^i(x_i,y_i)$ *and* $\nabla_{y_i}\phi_\mu^i(x_i,y_i)$ *are given as in Corollary* 5.4, *with* x, y, w, z *replaced by* x_i, y_i, w_i, z_i, *respectively; and* $\nabla g(z_i) \in \Re^{n_i \times n_i}$ *is specified as in Proposition* 5.2 *with* $\hat{g}(\alpha) = \alpha^{1/2}$ *for all* $\alpha \in \Re_{++}$.

Proof. Fix any $\mu > 0$ and $(x,y,\zeta) \in \Re^n \times \Re^n \times \Re^\ell$ satisfying (6.2) and (6.3). From (6.1), it is easily seen that $J_\mu(x,y,\zeta)$ given by (6.8) is the Jacobian matrix of the left-hand side of (1.6) with ϕ_μ as defined. We show below that $J_\mu(x,y,\zeta)$ is invertible.

Let $(u,v,\Delta) \in \Re^n \times \Re^n \times \Re^\ell$ be a vector in the null space of $J_\mu(x,y,\zeta)$. We will show that $u = v = 0$, $\Delta = 0$. By Corollary 5.4, we have $\nabla_{x_i}\phi_\mu^i(x_i,y_i)^T = I - L_{w_i}^{-1}L_{x_i}$ and $\nabla_{y_i}\phi_\mu^i(x_i,y_i)^T = I - L_{w_i}^{-1}L_{y_i}$, where $w_i = z_i^{1/2}$ and $z_i = x_i^2 + y_i^2 + 2\mu^2 e$. Then, by (6.8),

$$(I - L_w^{-1}L_x)u + (I - L_w^{-1}L_y)v = 0, \qquad \nabla F(x,y,\zeta)^T(u,v,\Delta) = 0, \tag{6.9}$$

where for simplicity we define

$$L_w = \mathrm{diag}\,\{L_{w_1}, \ldots, L_{w_m}\},\ L_x = \mathrm{diag}\,\{L_{x_1}, \ldots, L_{x_m}\},\ L_y = \mathrm{diag}\,\{L_{y_1}, \ldots, L_{y_m}\}.$$

Applying L_w to both sides of the first equation in (6.9) yields

$$(L_w - L_x)u + (L_w - L_y)v = 0.$$

Since $w_i^2 = x_i^2 + y_i^2 + 2\mu^2 e \succ_{\mathcal{K}^{n_i}} x_i^2 + y_i^2$, Lemma 3.5 shows that $L_{w_i} - L_{y_i}$ is invertible for all i, and hence $L_w - L_y$ is invertible. Then, multiplying both sides of the above equation by $u^T(L_w - L_y)^{-1}$ from the left yields

$$u^T(L_w - L_y)^{-1}(L_w - L_x)u + u^T v = 0. \tag{6.10}$$

Lemma 3.5 shows that $(L_{w_i} - L_{x_i})(L_{w_i} - L_{y_i}) \succ 0$ for all i, and hence $(L_w - L_x)(L_w - L_y) \succ 0$. Then $u^T(L_w - L_y)^{-1}(L_w - L_x)u = d^T(L_w - L_x)(L_w - L_y)d \geq 0$, where we let $d = (L_w - L_y)^{-1}u$. Also, by assumption (6.3), we have from the second equation in (6.9) that $u^T v \geq 0$. Thus, (6.10) implies $d^T(L_w - L_x)(L_w - L_y)d = 0$ and hence $d = 0$. Then $u = 0$ and since $I - L_w^{-1}L_y$ is invertible, the first equation in (6.9) implies $v = 0$. Since $\nabla_\zeta F(x,y,\zeta)$ has rank ℓ, the second equation in (6.9) then implies $\Delta = 0$.

Thus, the null space of $J_\mu(x,y,\zeta)$ comprises only the origin, and thus $J_\mu(x,y,\zeta)$ is invertible. □

Thus far we have not specified an algorithm for solving the SOCCP. This is intentional, since algorithm design and analysis is not a focus of this paper. In some cases and respects, a smoothing algorithm designed for solving an NCP can be extended to solve an SOCCP in a relatively straightforward manner. For example, the algorithm in [22], which was extended in [6] to solve an SDCP, can also be extended to solve an SOCCP. Its global convergence analysis [22, Proposition 3.1] then extends accordingly using Propositions 6.1 and 6.2.

7. Applications to optimization with SOC constraints. Consider the following optimization problem with SOC constraints:

$$\begin{array}{ll} \text{minimize} & f(\zeta) \\ \text{subject to} & h(\zeta) - x = 0, \quad x \in \mathcal{K}, \end{array} \tag{7.1}$$

where $f : \Re^\ell \to \Re$ and $h : \Re^\ell \to \Re^n$ are twice continuously differentiable functions and $\mathcal{K}$ has the form (1.2). In the special case in which both f and g are affine, this problem reduces to that studied in [14]. We discuss below how results from the previous section can be applied to solving this problem.

By attaching the Lagrange multiplier vector $y \in \Re^n$ to the equality constraints in (7.1) and introducing the Lagrangian

$$l(x,y,\zeta) = f(\zeta) - h(\zeta)^T y + x^T y,$$

we can write the first-order necessary optimality conditions for (7.1) in the form of SOCCP (1.1) with

$$F(x,y,\zeta) = \begin{pmatrix} \nabla_y l(x,y,\zeta) \\ \nabla_\zeta l(x,y,\zeta) \end{pmatrix} = \begin{pmatrix} x - h(\zeta) \\ \nabla f(\zeta) - \nabla h(\zeta) y \end{pmatrix}. \tag{7.2}$$

Notice that F is continuously differentiable. Moreover, it is straightforward to verify that $\nabla F(x,y,\zeta)$ satisfies the rank and monotonicity assumptions (6.2) and (6.3) if

$$\operatorname{rank} \nabla h(\zeta) = \ell \qquad \text{and} \qquad \nabla^2_{\zeta\zeta} l(x,y,\zeta) \succeq 0.$$

In particular, this is satisfied if f is convex and h is affine with ∇h having linearly independent columns.

Using (7.2), it can be seen that the Newton equation associated with (1.6) has the form

$$\begin{aligned} \nabla_x \phi_\mu(x,y)^T u + \nabla_y \phi_\mu(x,y)^T v &= -\phi_\mu(x,y), \\ u - A^T \Delta &= h(\zeta) - x, \\ Q\Delta - Av &= -\nabla_\zeta l(x,y,\zeta), \end{aligned}$$

where for simplicity we denote $A = \nabla h(\zeta) \in \Re^{\ell \times n}$ and $Q = \nabla^2_{\zeta\zeta} l(x,y,\zeta) \in \Re^{\ell \times \ell}$. Upon eliminating u and v from these equations, we obtain the reduced system

$$\big(Q + A(\nabla_y \phi_\mu(x,y)^T)^{-1} \nabla_x \phi_\mu(x,y)^T A^T\big)\Delta = r,$$

where

$$r = -\nabla_\zeta l(x,y,\zeta) - A(\nabla_y \phi_\mu(x,y)^T)^{-1} \big(\phi_\mu(x,y) + \nabla_x \phi_\mu(x,y)^T (h(\zeta) - x)\big).$$

Upon writing $A = [A_1 \ \cdots \ A_m]$, with $A_i \in \Re^{\ell \times n_i}$ for $i = 1, \ldots, m$, and using the product structure (6.1), we can rewrite the above reduced system as

$$\left(Q + \sum_{i=1}^{m} A_i D_i A_i^T \right) \Delta = r, \tag{7.3}$$

where for simplicity we define

$$D_i = (\nabla_{y_i} \phi_\mu^i(x_i, y_i)^T)^{-1} \nabla_{x_i} \phi_\mu^i(x_i, y_i)^T.$$

Here and throughout, our notations are as in the previous section. We show below that the left-hand matrix of (7.3) can be efficiently computed for the choices of ϕ_μ^i specified in Proposition 6.1 or 6.2. In particular, we show that D_i can be efficiently computed and has a sparsity structure.

For ϕ_μ^i as specified in Proposition 6.1, we have

$$D_i = \nabla g(z_i)^{-1} - I$$

with $z_i = (x_i - y_i)/\mu$. If $z_{i2} = 0$, then Proposition 5.2 yields $\nabla g(z_i)^{-1} = \hat{g}'(z_{i1})^{-1} I$. If $z_{i2} \neq 0$, then Proposition 5.2 together with $a_i b_i = b_i^2 - c_i^2$ yields

$$\nabla g(z_i)^{-1} = \frac{1}{a_i} \begin{bmatrix} 1 & -c_i z_{i2}^T / (b_i \|z_{i2}\|) \\ -c_i z_{i2} / (b_i \|z_{i2}\|) & I \end{bmatrix},$$

where a_i, b_i, c_i are as defined in the proof of Proposition 6.1. Thus D_i is almost diagonal, except for a dense first row and first column.

For ϕ_μ^i as specified in Proposition 6.2, we have by Corollary 5.4

$$\begin{aligned} D_i &= (I - L_{w_i}^{-1} L_{y_i})^{-1} (I - L_{w_i}^{-1} L_{x_i}) \\ &= (L_{w_i} - L_{y_i})^{-1} (L_{w_i} - L_{x_i}) \\ &= (L_{w_i - y_i})^{-1} L_{w_i - x_i}, \end{aligned}$$

where $w_i = (x_i^2 + y_i^2 + 2\mu^2 e)^{1/2}$. When calculating $L_{w_i - x_i}$, we only need to compute and store the vector $w_i - x_i$, since it can be used to easily determine $L_{w_i - x_i}$. The main computational effort therefore lies in computing and storing the matrix $L_{w_i - y_i}^{-1}$. Since $L_{w_i - y_i}$ is the sum of the matrix $(w_{i1} - y_{i1})I$, whose inverse is easily obtained, and a symmetric rank two matrix, we can store only the vector $w_i - y_i$ and use the well-known Sherman–Morrison–Woodbury updating formula to evaluate $L_{w_i - y_i}^{-1}$. The total number of such updates required to compute all of the D_i's is then proportional to m, and the total storage requirement is proportional to n.

REFERENCES

[1] F. Alizadeh and S. Schmieta, *Symmetric cones, potential reduction methods*, in Handbook of Semidefinite Programming, H. Wolkowicz, R. Saigal, and L. Vandenberghe, eds., Kluwer, Boston, 2000, pp. 195–233.

[2] J. Burke and S. Xu, *The global linear convergence of a non-interior path-following algorithm for linear complementarity problems*, Math. Oper. Res., 23 (1998), pp. 719–734.

[3] B. Chen and P. T. Harker, *A non-interior-point continuation method for linear complementarity problems*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 1168–1190.

[4] C. Chen and O. L. Mangasarian, *Smoothing methods for convex inequalities and linear complementarity problems*, Math. Programming, 71 (1995), pp. 51–69.

[5] C. Chen and O. L. Mangasarian, *A class of smoothing functions for nonlinear and mixed complementarity problems*, Comput. Optim. Appl., 5 (1996), pp. 97–138.
[6] X. Chen and P. Tseng, *Non-interior continuation methods for solving semidefinite complementarity problems,* Math. Program., to appear.
[7] B. Chen and N. Xiu, *A global linear and local quadratic noninterior continuation method for nonlinear complementarity problems based on Chen–Mangasarian smoothing functions*, SIAM J. Optim., 9 (1999), pp. 605–623.
[8] X. Chen, L. Qi, and D. Sun, *Global and superlinear convergence of the smoothing Newton method and its application to general box constrained variational inequalities*, Math. Comp., 67 (1998), pp. 519–540.
[9] U. Faraut and A. Korányi, *Analysis on Symmetric Cones*, Oxford Mathematical Monographs, Oxford University Press, New York, 1994.
[10] M. C. Ferris and J.-S. Pang, eds., *Complementarity and Variational Problems: State of the Art*, SIAM, Philadelphia, 1997.
[11] M. Fukushima and L. Qi, eds., *Reformulation—Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods*, Kluwer, Boston, 1999.
[12] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.
[13] C. Kanzow, *Some noninterior continuation methods for linear complementarity problems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 851–868.
[14] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, *Applications of second-order cone programming*, Linear Algebra Appl., 284 (1998), pp. 193–228.
[15] R. D. C. Monteiro and T. Tsuchiya, *Polynomial convergence of primal-dual algorithms for the second-order cone programs based on the MZ-family of directions*, Math. Program., 88 (2000), pp. 61–83.
[16] L. Qi and H. Jiang, *Semismooth Karush-Kuhn-Tucker equations and convergence analysis of Newton and quasi-Newton methods for solving these equations*, Math. Oper. Res., 22 (1997), pp. 301–325.
[17] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.
[18] S. Schmieta and F. Alizadeh, *Associative Algebras, Symmetric Cones and Polynomial Time Interior Point Algorithms*, Report RRR 17-98, RUTCOR, Rutgers University, Piscataway, NJ, 1998.
[19] S. Smale, *Algorithms for solving equations*, in Proceedings of the International Congress of Mathematicians, American Mathematical Society, Providence, RI, 1987, pp. 172–195.
[20] D. Sun and J. Sun, *Semismooth matrix valued functions*, manuscript, School of Mathematics, The University of New South Wales, Sydney, Australia, 2000.
[21] P. Tseng, *Merit functions for semi-definite complementarity problems*, Math. Programming, 83 (1998), pp. 159–185.
[22] P. Tseng, *Analysis of a non-interior continuation method based on Chen-Mangasarian smoothing functions for complementarity problems,* in Reformulation—Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods, M. Fukushima and L. Qi, eds., Kluwer, Boston, 1999, pp. 381–404.
[23] T. Tsuchiya, *A convergence analysis of the scaling-invariant primal-dual path-following algorithms for second-order cone programming*, Optim. Methods Softw., 11 (1999), 141–182.
[24] R. J. Vanderbei and H. Y. Benson, *On Formulating Semidefinite Programming Problems as Smooth Convex Nonlinear Optimization Problems*, ORFE 99-01, Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ, 1999.
[25] N. Yamashita and M. Fukushima, *Modified Newton methods for solving semismooth reformulations of monotone complementarity problems*, Math. Programming, 76 (1997), pp. 469–491.
[26] E. H. Zarantonello, *Projections on convex sets in Hilbert space and spectral theory*, in Contributions to Nonlinear Functional Analysis, E. H. Zarantonello, ed., Academic Press, New York, 1971, pp. 237–424.

SIAM J. OPTIM.
Vol. 12, No. 2, pp. 461–478

A VARIATIONAL PRINCIPLE FOR PROBLEMS WITH FUNCTIONAL CONSTRAINTS*

A. D. IOFFE†, R. E. LUCCHETTI‡, AND J. P. REVALSKI§

Abstract. In this paper we show that in several important classes of optimization problems, like mathematical programming with k-smooth data, quadratic programming in a Hilbert space, convex programming in a Banach space, semi-infinite programming, and optimal control of linear systems with quadratic cost, most of the problems (in the Baire category sense) are well-posed. This is derived from a general variational principle for problems with functional constraints.

Key words. variational principle, well-posed optimization problem, mathematical programming, control problems, genericity, Baire category

AMS subject classifications. 49K40, 90C31, 49J99

PII. S1052623400378274

1. Introduction and preliminaries. Existence results for optimization problems are usually based on the classical Weierstrass theorem claiming that a lower semicontinuous function attains its minimal value on a compact set. But it has been noticed that in certain classes of optimization problems, solutions may typically exist (and even have some additional good properties like uniqueness, stability, etc.) even if the compactness condition of the Weierstrass theorem is not satisfied for most of the problems. The terms "typically exists" and "most of the problems" are understood here in the sense of the Baire category: Given a class of problems $\mathcal{P}$, endowed with a complete metric, there exists a dense G_δ-subset $\mathcal{P}'$ of $\mathcal{P}$ such that every problem $P \in \mathcal{P}'$ has a solution (or is well-posed; see the definition below). In other words, the existence of the solution is a generic property in the class $\mathcal{P}$. We can refer, e.g., to [BL, CK, CKR1, CKR2, LP, R1, R2], where it was shown that for different classes of optimization problems a typical problem is well-posed in one or another sense. Surveys of the results obtained at this stage can be found in [DoZo] and [KR].

At the abstract level, the fact that a typical minimization problem in some class is well-posed under certain conditions is obtained either by involving set-valued mappings and their continuity-like properties (see, e.g., [CKR1, CKR2]) or, more directly, by using some variational principle, for instance that of Deville–Godefroy–Zizler [DGZ] and its recent modifications by Deville–Revalski [DR] and Ioffe–Zaslavski [IZa] (see also [IT]). The first of these modifications concerns what is meant by "typical" ("dense G_δ" is replaced by the stronger property of being the complement of a σ-porous set), while the principle from [IZa] has been recently used in several new situations (e.g., calculus of variations [IZa] or convex and quasi-convex programming [IL]). In this paper we use the Ioffe–Zaslavski principle to get another principle suited

*Received by the editors September 18, 2000; accepted for publication (in revised form) July 5, 2001; published electronically December 14, 2001.
http://www.siam.org/journals/siopt/12-2/37827.html

†Department of Mathematics, Technion, Haifa 32000, Israel (ioffe@math.technion.il).

‡Department of Mathematics, Politecnico di Milano, Piazzale Garbetto 6, 22100 Como, Italy (rel@komodo.ing.unico.it). This author's research was partially supported by Ministero dell'Università e della Ricerca Scientifica e Tecnologica (40%, 1999) and by Gruppo Nazionale per l'Analisi Funzionale e le sue Applicazioni.

§Institute of Mathematics, Bulgarian Academy of Sciences, Acad. G. Bonchev Street, Block 8, 1113 Sofia, Bulgaria (revalski@math.bas.bg). This author's research was partially supported by NFSR at the Bulgarian Ministry of Science and Education grant MM-701/97.

to a general infinite dimensional problem of the form

$$\text{(P)} \qquad \text{minimize} \quad f(x) \quad \text{subject to} \quad F(x) \in C, \quad G(x) = 0,$$

where f, F, G are continuous maps.

The list of known results concerning the generic well-posedness of (P) with respect to variations of "natural data" such as (f, F, G) is rather short: we can mention just [R1, IL], where convex and quasi-convex minimization under only inequality constraints was considered, and an earlier paper by Spingarn and Rockafellar [SRo], which was concerned with the question of the generic regularity of necessary optimality conditions.

The main result of this paper is a variational principle for problems with constraints as in (P). It is stated in the next section and proved in the concluding section 4. In section 3 we consider several important classes of problems (mathematical programming with k-smooth data, quadratic programming in a Hilbert space, convex programming in a Banach space, semi-infinite programming, and optimal control of linear systems with quadratic cost) and show that generically, in the set of "meaningful data," they all are well-posed with respect to some natural metrics in the data space.

We begin by explaining what kind of well-posedness we are going to consider and by giving a suitable formulation of the variational principle of Ioffe–Zaslavski on which our main result is based.

Let $(X, \|\cdot\|)$ be a real Banach space and $(\mathcal{A}, d)$ be a metric space which is a Baire space. (The latter means that a countable intersection of open dense subsets is dense in the space.) We shall call X the *domain space*. The space $\mathcal{A}$ will serve as a *data space*. Assume that a lower semicontinuous extended real-valued function $f_a : X \to \mathbf{R} \cup \{+\infty\}$ is associated with each $a \in \mathcal{A}$, and consider the problem of minimizing f_a on X. Denote by $\inf f_a$ the infimum of f_a on the space X.

We say that this problem (for a given a) is *well-posed*, provided that the following are true:

1. $\inf f_a$ is finite and attained at a unique point $x_0 \in X$;
2. for any sequence $\{a_n\}$ converging to a, $\inf f_{a_n}$ is finite for large n and any sequence $\{z_n\} \subset X$ such that $f_{a_n}(z_n) - \inf f_{a_n} \to 0$ strongly converges to x_0;
3. if a_n converges to a, then $\inf f_{a_n} \to \inf f_a = f_a(x_0)$.

The first two conditions are called "well-posedness by perturbations" in [Zo1, Zo2], while the third one is known in the literature as *value Hadamard well-posedness* (see, e.g., [DoZo]).

Now, consider the following condition.

($\mathcal{H}$) *There is a dense subset $\mathcal{B} \subset \mathcal{A}$ such that for any $a \in \mathcal{B}$, any $\varepsilon > 0, \gamma > 0$, there exist a nonempty open set $\mathcal{V} \subset \mathcal{A}$, $\bar{x} \in X$, $\alpha \in \mathbf{R}$, and $\lambda > 0$ such that for every $b \in \mathcal{V}$ we have the following:*

(i) *$d(a, b) < \varepsilon$ and $\inf f_b > -\infty$;*

(ii) *if $z \in X$ is such that $f_b(z) < \inf f_b + \lambda$, then $\|z - \bar{x}\| \le \gamma$ and $|f_b(z) - \alpha| \le \gamma$.*

The variational principle from [IZa] can now be stated as follows.

THEOREM 1.1 (see [IZa]). *Let X be a real Banach space and $(\mathcal{A}, d)$ be a Baire space. Suppose that ($\mathcal{H}$) holds. Then there exists a dense G_δ-subset $\mathcal{A}_1$ of $(\mathcal{A}, d)$ such that for every $a \in \mathcal{A}_1$ the corresponding minimization problem is well-posed.*

Finally, we mention that some of the results contained in this paper were announced in [ILR].

2. Statement of the main results.

2.1. The abstract framework. We begin by introducing the main classes of objects which are involved in the formulation of optimization problems of the (P) type. Throughout the paper, X, Y, Z are real Banach spaces and $C \subset Y$ is a closed convex cone with a nonempty interior. We shall use the same symbol $\|\cdot\|$ for the norms in all three spaces without indicating which specific norm is considered; this will always be clear from the context. We shall also fix an element $e \in \operatorname{Int} C$ (the interior of C).

Next we introduce the space $\mathcal{C}$ of all triples (f, F, G), where f, F, and G are continuous mappings from X into $\mathbf{R}$, Y, and Z, respectively. We assume that this space is a complete metric space with respect to a metric whose properties will be specified later (see hypothesis (A_1) below). We shall call $\mathcal{C}$ the *cost-constraint space*. Occasionally, we shall denote by $\mathcal{C}_0$, $\mathcal{C}_{\leq}$, and $\mathcal{C}_{=}$ the component spaces of elements of $\mathcal{C}$ corresponding to the mappings f, F, and C, respectively, with the projection topologies. To be more precise, we shall consider as the cost component $\mathcal{C}_0$ of $\mathcal{C}$ the factor space of the space of continuous functions by the subspace of constants (as the problem obviously does not change if we add a constant to the cost function). But we shall always deal with representatives of the classes and therefore shall treat elements of $\mathcal{C}_0$ as functions.

Finally, we shall consider one more metric space $(\mathcal{D}, d)$, which will be called the *data space*, along with a continuous mapping $\pi : \mathcal{D} \to \mathcal{C}$. We shall not assume the data space to be complete and will require only that it is a Baire space. Now given $a \in \mathcal{D}$, we consider the problem

$$(\mathrm{P})_a \qquad \text{minimize} \quad f(x) \quad \text{subject to} \quad F(x) \in C, \quad G(x) = 0,$$

where $(f, F, G) = \pi(a)$. The idea of having the mapping π is (as will be seen in the examples) to put different types of problems into the above scheme of triples. But we stress the fact that all genericity results will be proved in the original class $(\mathcal{D}, d)$.

Every problem $(\mathrm{P})_a$ can be equivalently represented as an unconstrained minimization of the extended real-valued function

$$f_a(x) = \begin{cases} f(x) & \text{if } F(x) \in C, \quad G(x) = 0, \\ \infty, & \text{otherwise.} \end{cases}$$

Certainly, the feasible set of $(\mathrm{P})_a$ coincides with the domain of f_a:

$$\operatorname{dom} f_a = \{x \in X : F(x) \in C,\ G(x) = 0\}.$$

Two subsets of the data space will be of special importance: the set of *feasible data*

$$\mathcal{F} = \{a \in \mathcal{D} : \operatorname{dom} f_a \neq \emptyset\}$$

and the set of *meaningful data*

$$\mathcal{M} = \{a \in \mathcal{D} : |\inf f_a| < \infty\}.$$

Clearly, $\mathcal{M} \subset \mathcal{F}$.

Example 2.1. The scheme we have just described typically occurs in the formulation of concrete problems. Indeed, consider for instance the following simple problem of quadratic programming with linear equality constraints:

$$\text{minimize} \quad (Qx|x) + (c|x) \quad \text{subject to} \quad Ax = b,$$

where $(\cdot|\cdot)$ is the scalar product in $\mathbf{R}^n$, Q is an $n \times n$ symmetric matrix, A is an $m \times n$ matrix, $c \in \mathbf{R}^n$, and $b \in \mathbf{R}^m$.

It is natural to consider the set of all quadruples (Q, A, c, b) as the data space, with the mapping π defined (for any $a = (Q, A, c, b)$) by $\pi(a) = (f, G)$, where $f(x) = (Qx|x)+(c|x)$ and $G(x) = Ax-b$. We shall consider the natural product metric on the data space $\mathcal{D}$ while the space $\mathcal{C}$ is endowed by the topology of uniform convergence on bounded sets of $\mathbf{R}^n$. Clearly, π is continuous. (To be very formal, we are also assuming that the space Y is $Y = \{0\}$.)

It is possible to consider other data spaces for the same problem, for instance the collection of all quadruples (Q, B, c, b) with Q positive semidefinite, etc. We shall consider this example in more detail in the next section (Example 3.6).

2.2. Assumptions. In what follows we adopt the following convention: For $R_0 \in \{0, \infty\}$ (i.e., either $R_0 = 0$ or $R_0 = \infty$), the expression $R \geq R_0$ means that either $R = \infty$ if $R_0 = \infty$, or R is a nonnegative real number if $R_0 = 0$. The main assumptions concerning the cost constraint and the data spaces follow.

There is $R_0 \in \{0, \infty\}$ such that the following four conditions hold:

(A_1) (a) The metric topology in $\mathcal{C}$ is the topology of uniform convergence on balls, centered at zero, of radius $R \geq R_0$.
(b) Any function f such that $\pi(a) = (f, F, G)$ for some $a \in \mathcal{D}$ is bounded from below on every ball, centered at zero, of radius $R \geq R_0$.

(A_2) There is a continuous function $q(\cdot)$ on X with the following properties:
(a) $0 = q(0) = \min q$; $q(x) \geq 1$ if $\|x\| \geq 1$.
(b) For any $a \in \mathcal{D}$ with $\pi(a) = (f, F, G)$, any sufficiently small $t, \tau \geq 0$, every $z \in Z$ with sufficiently small norm, and every $\gamma > 0$ and $w \in X$, there is an element $\tilde{a} = \tilde{a}(t, \tau, \gamma, w, z) \in \mathcal{D}$ such that $\pi(\tilde{a}) = (f_{t\gamma w}, F_\tau, G_z)$, where $f_{t\gamma w}(x) = f(x)+tq(\gamma^{-1}(x-w))$, $F_\tau(x) = F(x)+\tau e$, and $G_z(x) = G(x) + z$, $x \in X$. Moreover, for each fixed γ, $d(\tilde{a}, a) \to 0$ uniformly for w on every ball, centered at zero, with radius $R \geq R_0$, as t, τ, and z tend to zero.

(A_3) For any $a \in \mathcal{D}$ with $\pi(a) = (f, F, G)$, any $x \in X$ with $G(x) = 0$, and every $\varepsilon > 0$, there is an open set $U \subset \mathcal{C}_=$ such that for each $\hat{G} \in U$ the following are true:
(a) There is $u \in X$ with $\|u - x\| < \varepsilon$ such that $\hat{G}(u) = 0$.
(b) There exists $\hat{a} \in \mathcal{D}$ with $\pi(\hat{a}) = (\hat{f}, \hat{F}, \hat{G})$ for some $\hat{f}, \hat{F}$, so that $d(a, \hat{a}) < \varepsilon$.

(A_4) There is a dense subset $\hat{\mathcal{M}}$ of $\mathcal{M}$ such that for every $a \in \hat{\mathcal{M}}$ there exist an open (in $\mathcal{D}$) neighborhood $\mathcal{N}_a$ of a, $\xi > 0$, and $R \geq R_0$ so that for every $b \in \mathcal{N}_a$,
(a) $\inf f_b > -\infty$;
(b) if $f_b(x) < \inf f_b + \xi$, then $\|x\| < R$.

We shall briefly comment on these hypotheses. (A_1) says that there are two types of topologies considered in $\mathcal{C}$: the topology of uniform convergence on bounded subsets of X (if $R_0 = 0$) and the topology of uniform convergence on the whole X (if $R_0 = \infty$). Although these two topologies are very different, it is possible to give a unified proof of the genericity of the well-posedness for both topologies, which is mainly a proof for the first (more difficult) case $R_0 = 0$ with a few stipulations concerning the second. We shall call R_0 *the index of the topology* in the cost-constraint space. The second part of (A_1) states that we (have to) consider lower bounded functions in the case of uniform convergence on the whole space, and functions which are lower bounded on

the bounded sets when $R_0 = 0$, which is quite natural.

Hypotheses (A_2) and (A_3) describe the types of perturbations in the components of the data space which we have to allow to reach the desired conclusion. They impose certain limitations on the range of the problems covered by our results. In particular, linear programming and quasi-convex programming are not covered, as the perturbations required in (A_2) are not possible in these settings. But we shall see in the next section that the assumptions allow quite a bit of flexibility and are not difficult to verify.

Finally, the last hypothesis (A_4): It is clear, in view of (A_1), that (A_4) is automatically satisfied if $R_0 = \infty$. If $R_0 = 0$, then (A_4) is a sort of coercivity assumption needed to get an amount of boundedness of minimizing sequences. Observe that it does not follow from (A_4) that all elements of $\mathcal{N}_a$ belong to $\mathcal{F}$. If $\operatorname{dom} f_b = \emptyset$, then the implication $f_b(x) < \inf f_b + \xi \Rightarrow \|x\| < R$ holds trivially, as the set of such x is empty.

2.3. Statement of the main theorem. Here we state our main result followed by a short comment. The proof itself is found in section 4.

THEOREM 2.2. *Under* (A_1)–(A_4), *either* $\mathcal{M} = \emptyset$ *or* $\mathcal{M}$ *is a Baire space which contains a dense* G_δ*-subset* $\mathcal{M}'$ *such that* $(MP)_a$ *is well-posed for every* $a \in \mathcal{M}'$.

We would like to emphasize once again that we prove our genericity result in the original data space $\mathcal{D}$, not in the cost-constraint space. This is the natural point of view, as will be seen also by the examples, since one usually deals with the data of a problem as they are given; thus one wants to be sure that for most of these data the corresponding optimization problem is well-posed.

3. Applications. In this section we shall give several examples of how our principle can be applied.

Example 3.1 (Nonlinear programming: Uniform convergence on the domain space). Let $X = \mathbf{R^n}$, $Y = \mathbf{R^m}$, $Z = \mathbf{R^l}$ be Euclidean spaces of dimensions n, m, and l, respectively. We assume $n \geq l$. Denote by $C^k(X)$, $k \geq 0$, the linear space of all real-valued functions in X which are continuous along with all of their derivatives up to the order k. We endow it with the (obviously complete) metric

$$d_k(f,g) = \sum_{i=0}^{k} \sup \left\{ \frac{|f^{(i)}(x) - g^{(i)}(x)|}{1 + |f^{(i)}(x) - g^{(i)}(x)|} : x \in X \right\}. \tag{3.1}$$

The corresponding topology is the topology of uniform convergence on the whole space X of functions and all their derivatives up to the order k. Let $C_b^k(X)$ be the subspace of $C^k(X)$ containing all functions bounded from below. Clearly, this is a closed subspace, and hence a complete metric space in the same metric d_k. Consider the linear spaces $C^k(X,Y)$ and $C^k(X,Z)$ of k-times continuously differentiable mappings from X into Y and Z, respectively, which we consider here with standard metrics (as above in (3.1)) defining the topology of uniform convergence on X of the mappings and their derivatives up to the order k.

As a first example we consider the standard mathematical programming problem:

(MP)
$$\text{minimize} \quad f(x) \quad \text{subject to} \quad f_1(x) \leq 0, \ldots, f_m(x) \leq 0,\ g_1(x) = 0, \ldots, g_l(x) = 0,$$

where $f \in C^k(X)$, $F(\cdot) = (f_1(\cdot), \ldots, f_m(\cdot)) \in C^k(X,Y)$, and $G(\cdot) = (g_1(\cdot), \ldots, g_l(\cdot)) \in C^k(X,Z)$.

We define the data space as

$$\mathcal{D} = C_b^k(X) \times C^k(X,Y) \times C^k(X,Z),$$

with some product metric generated by d_k-metrics in the coordinate spaces, which for simplicity we again denote by d_k. The cost-constraint space is

$$\mathcal{C} = C^0(X) \times C^0(X,Y) \times C^0(X,Z),$$

again with some product metric d_0 generated by the d_0-metrics on the coordinate spaces. The mapping π is the embedding of $\mathcal{D}$ into $\mathcal{C}$. Since on $\mathcal{D}$ the metric d_k is stronger than d_0, the mapping π is continuous. The cone C is the negative orthant in $\mathbf{R^m}$, and as e we take the vector $(-1,\dots,-1)$.

To apply our theorem, we have to verify (A_1)–(A_4). Here $R_0 = \infty$ and (A_1) is obvious. (A_2) is also valid: just take a bounded function $q \in C^k(X)$ satisfying condition (a) of (A_2) and, in addition, that $q(x) = 1$ for every $x \in X$ so that $\|x\| \geq 1$. (Such functions are called *bump functions* and always exist in finite dimensions.)

To verify (A_3), suppose we are given $\varepsilon > 0$ and a mapping

$$G(u) = (g_1(\cdot),\dots,g_l(\cdot)) \in C^k(X,Z)$$

such that $G(x) = 0$ at a given $x \in X$. We shall first approximate G by another mapping $\bar{G}$ as follows.

(i) If $k \geq 1$, then let

$$H(u) := G'(x)(u-x) - G(u), \quad u \in X.$$

Obviously $H \in C^k(X,Z)$, $H(x) = 0$, and $H'(x) = 0$. Take a function $\lambda \in C^k(X)$ with values in $[0,1]$ and such that

$$\lambda(u) = 1 \quad \text{if } \|u-x\| \leq \tfrac{1}{2}, \qquad \lambda(u) = 0 \quad \text{if } \|u-x\| > 1.$$

Let $\|\lambda\|_\infty$ be the usual sup-norm of λ in $C^k(X)$. Consider now a full rank $l \times n$ matrix A such that $\|A - G'(x)\| < \frac{\varepsilon}{2^{k+1}\|\lambda\|_\infty}$ (where the norm for the matrices is the usual one) and set

$$\bar{G}(u) = \lambda(u)A(u-x) + (1-\lambda(u))G'(x)(u-x) - H(u), \quad u \in X.$$

(ii) If $k = 0$, we take an arbitrary full rank $l \times n$ matrix A and fix $\delta > 0$ such that

$$\|G(u)\| < \frac{\varepsilon}{2} \quad \text{if } \|u-x\| < \delta, \qquad \|A\|\delta < \frac{\varepsilon}{2}$$

and take a continuous function λ with values in $[0,1]$ and such that

$$\lambda(u) = 1 \quad \text{if } \|u-x\| \leq \tfrac{\delta}{2}, \qquad \lambda(u) = 0 \quad \text{if } \|u-x\| > \delta.$$

Then set

$$\bar{G}(u) = (1-\lambda(u))G(u) + \lambda(u)A(u-x), \quad u \in X.$$

In both cases we have $\bar{G}(x) = 0$, the distance in $\mathcal{C}_=$ between $\bar{G}$ and G is less than or equal to $(\varepsilon/2)$, and $\bar{G}$ is continuously differentiable around x with $\bar{G}'(x) = A$ being a full rank matrix. Therefore, there is $c > 0$ such that for sufficiently small $t > 0$ the

image of the t-ball around x by $\bar{G}$ covers the ct-ball around $0 \in Z$. Then a standard degree argument shows that every continuous mapping $\hat{G}$ sufficiently close to $\bar{G}$ (in the uniform metric) on the t-ball around x will cover the $\frac{ct}{2}$-ball around zero $\in Z$. Consequently, there is u with $\|u - x\| \leq t$ such that $\hat{G}(u) = 0$. This shows (A_3). Finally, (A_4) does not need to be verified in this case ($R_0 = \infty$).

For the next examples we need the following auxiliary result. Its proof is given in section 4.

PROPOSITION 3.2. *Assume* (A_1)–(A_3). *Then* $\mathcal{F}$ *contains a dense set* $\hat{\mathcal{F}}$ *such that* $\hat{\mathcal{F}}$ *is open in* $\mathcal{D}$ *and for every* $a \in \hat{\mathcal{F}}$ *there are an open (in* $\mathcal{D}$*) neighborhood* $\mathcal{N}$ *of* a *and a real number* K *with* $\inf f_b < K$ *for all* $b \in \mathcal{N}$.

Observe that the above result is true only supposing the first three assumptions. It will be used below for verification of (A_4).

Example 3.3 (Nonlinear programming: Uniform convergence on bounded sets). As a second example we consider the same standard (MP) problem as in the previous one, but with a different data space and with the topology in cost-constraint space changed to the topology of uniform convergence on bounded sets. Suppose we are given a function $\psi : X \to \mathbf{R}$, continuous and coercive, i.e., $\psi(x) \to \infty$ as $\|x\| \to \infty$. Consider the set

$$C^k_\psi(X) = \{f \in C^k(X) : f \geq \psi\}$$

with the distance

$$d_{kb}(f, g) = \sum_{j=0}^{\infty} \frac{1}{2^j} \rho_{jk}(f, g), \tag{3.2}$$

where the pseudometric $\rho_{jk}(f, g)$ is defined as

$$\rho_{jk}(f, g) = \sum_{i=0}^{k} \sup \left\{ \frac{|f^{(i)}(x) - g^{(i)}(x)|}{1 + |f^{(i)}(x) - g^{(i)}(x)|} : \|x\| \leq j \right\}. \tag{3.3}$$

This distance defines the topology of the uniform convergence of functions and their derivatives up to the order k on the bounded subsets of X. As to the constraint mappings, we shall take them from the same spaces $C^k(X, Y)$, $C^k(X, Z)$, supplied this time with similar metrics defining the topology of uniform convergence of them and their derivatives on the bounded sets.

Thus the data space is now

$$\mathcal{D} = C^k_\psi(X) \times C^k(X, Y) \times C^k(X, Z),$$

with some product metric (which we denote by d_{kb}) obtained by the metrics d_{kb} in the component spaces. The cost-constraint space is the same as in the previous example, but this time endowed with a product metric d_{0b}. The map π is again the embedding and, by similar argument as above, it is continuous.

We claim that in this case the main theorem can also be applied, giving generic well-posedness in the set of meaningful data (which is obviously nonempty). The only condition we need to check is (A_4), as in this case $R_0 = 0$ and (A_1) is clearly satisfied, while (A_2) and (A_3) are verified exactly in the same way as above.

To check (A_4), first observe that because of the assumptions (coercivity of ψ and the fact that the underlying space is finite dimensional), $\mathcal{M} = \mathcal{F}$. Set $\hat{\mathcal{M}} = \hat{\mathcal{F}}$, where

the dense set $\hat{\mathcal{F}}$ is provided by Proposition 3.2. We have that for every $a \in \hat{\mathcal{F}}$ there exists a neighborhood $\mathcal{N}$ of a (we may think that $\mathcal{N} \subset \mathcal{M} = \mathcal{F}$) such that $\inf f_b$ is bounded above by a certain constant $K < \infty$ for all $b \in \mathcal{N}$.

With $\mathcal{N}$ so chosen, (A_4)(a) is clear. For (b), take any $x \in \mathrm{dom} f_b$ with $f_b(x) < \inf f_b + 1$. Then we have $\psi(x) \le K + 1$, and the coercivity of ψ implies that there must be $R < \infty$ such that $\|x\| < R$.

Remark. The first two examples can be extended to the case in which the domain space X is infinite dimensional, provided that there exists a C^k-bump function $q(\cdot)$ (which will satisfy property (a) of (A_2)). This is certainly the case in any Banach space if $k = 0$. If $k = 1$, a sufficient condition for the existence of such q is that X has an equivalent Fréchet differentiable norm. For more details concerning the existence of smooth bump functions (also for bigger k), we refer the reader to [DGZ].

In this infinite dimensional case, as far as the second example is concerned, we must take only those cost functions satisfying $f \ge \psi$ which are also bounded from below on the bounded sets. Then the verification of (A_4) is the same, since one easily sees that again $\mathcal{M} = \mathcal{F}$.

Example 3.4 (Convex programming). Now let X be a real Banach space, and let $\mathrm{Conv}(X)$ be the collection of all convex continuous functions on X equipped with the metric d_b of uniform convergence on bounded sets given by (3.2) and (3.3) with $k = 0$. As is well known, $(\mathrm{Conv}(X), d_b)$ is a complete metric space. Given $f, f_i \in \mathrm{Conv}(X)$, $i = 1, \dots, m$, we consider the problem of convex programming on X with only inequality constraints

$$\text{(CP)} \qquad \text{minimize} \quad f(x) \quad \text{subject to} \quad f_i(x) \le 0, \quad i = 1, \dots, m.$$

Here the data space is the product of $m + 1$ copies of $\mathrm{Conv}(X)$ with some product metric d_b. The cost-constraint space is $C^0(X) \times C^0(X, Y)$, with the same product metric d_b, where, as in the previous example, $Y = \mathbf{R}^m$ with the same negative cone C and $e = (-1, \dots, -1)$. The mapping π is again the embedding, which is obviously continuous. Generic well-posedness of (CP) in this case was derived in [IL] directly from Theorem 1.1. We get it here, in a simpler way, from Theorem 2.2.

Indeed, we have $R_0 = 0$, and (A_1) is satisfied. (Remember that any convex continuous function is bounded below on bounded sets.) To get (A_2), we set $q(x) = \|x\|$; the rest of (A_2) is easily checked. We need not verify (A_3), as the equality constraint is absent. Finally, to prove that (A_4) also holds, we consider the collection $\mathcal{D}'$ of data with coercive cost function: the family of coercive functions is dense in $(\mathrm{Conv}(X), d_b)$, as it is easy to verify (see also [IL]). Hence, $\mathcal{D}'$ is dense in $\mathcal{D}$. Set $\hat{\mathcal{M}} := \mathcal{D}' \cap \hat{\mathcal{F}}$, the latter set being the one provided by Proposition 3.2. Observe that $\hat{\mathcal{M}} \subset \mathcal{M}$ and that $\hat{\mathcal{M}}$ is dense in $\mathcal{M}$.

Now, it is also known that if the convex function $h \in \mathrm{Conv}(X)$ is coercive, then there are constants $\alpha > 0$, β, and a neighborhood V (in $(\mathrm{Conv}(X), d_b)$) of h such that, for all $h' \in V$, one has $h'(x) \ge \alpha\|x\| + \beta$ for each $x \in X$. Then the rest of (A_4) is checked as in Example 3.3 by using Proposition 3.2.

Before giving the next example we prove a simple regularity lemma on which verification of (A_3) in the following examples is based. Let, as above, X and Z be Banach spaces and $A : X \to Z$ a linear bounded operator which is *onto*. The latter entails that the Banach constant of A is positive:

$$C(A) = \sup\{r \ge 0 : rB_Z \subset A(B_X)\}$$

(where B_Z and B_X stand for the unit balls in the spaces Z and X, respectively). Then (see, e.g., [DuMO, Theorem 1.3]) for any other bounded linear operator A' from X

into Z, we have

$$C(A') \geq C(A) - \|A - A'\|.$$

LEMMA 3.5. *Let A be a bounded linear operator from X onto Z, and let $Ax = z$. Then for any $\varepsilon > 0$ there is $\delta > 0$ such that whenever an operator $A' : X \to Z$ and a vector $z' \in Z$ satisfy $\|A - A'\| < \delta$, $\|z - z'\| < \delta$, the equation $A'x' = z'$ has a solution x' satisfying $\|x - x'\| < \varepsilon$.*

Proof. Set $C(A) = 2c > 0$, and suppose that $\|A - A'\| < c$. Set further for a given $z' \in Z$, $v = z' - A'x$ (where $Ax = z$). Then $C(A') > c$ and we can find $u \in X$ such that $A'u = v$ and $\|u\| \leq c^{-1}\|v\|$. The equality means that for $x' = u + x$ we have $A'x' = z'$, whereas the inequality gives

$$\|x' - x\| \leq c^{-1}\|A'x - z'\| \leq c^{-1}(\|A - A'\|\|x\| + \|z - z'\|).$$

Thus, it is enough to take δ so small that $\delta < c$ and $\delta(\|x\| + 1) < c\varepsilon$. □

Example 3.6 (Quadratic programming in Hilbert space). In this example, X is a real Hilbert space with inner product $(\cdot|\cdot)$. The class of problems to be considered is described by the following scheme:

$$\text{minimize} \quad (Qx|x) + (c|x) \quad \text{subject to} \quad (Q_i x|x) + (c_i|x) \leq \alpha_i, \quad i = 1, \ldots, k, \quad Ax = u.$$

Here Q, Q_i are symmetric bounded linear operators in X, A is a bounded linear operator in X, $c, c_1, \ldots, c_k$ and u are elements of X, and α_i are real numbers. Thus the data space $\mathcal{D}$ is a collection of $(3k + 4)$-tuples

$$a = (Q, Q_1, \ldots, Q_k, A, c, c_1, \ldots, c_k, u, \alpha_1, \ldots, \alpha_k),$$

which we shall consider with the natural product topology corresponding to the norm convergences of operators and vectors of X and the usual convergence of numbers. The main (and only) assumption on the components of the data we adopt is that A must be an operator with full range: $A(X) = X$. Such operators form an open set in the space of bounded operators with the usual operator norm, which means that $\mathcal{D}$ is an open set in the space of all $(3k + 4)$-tuples having such structure. Hence, $\mathcal{D}$ is a Baire space in the product topology.

The definition of the cost-constraint space and the mapping π is equally straightforward. The cost-constraint space $\mathcal{C}$ is the collection of triples (f, F, G), where f is a continuous function on X, F is a continuous mapping from X into $\mathbf{R^k}$, and G is a continuous mapping from X into itself with the topology of uniform convergence on bounded sets, $\pi(a) = (f, F, G)$, where $f(x) = (Qx|x) + (c|x)$, $F = (f_1, \ldots, f_k)$ with $f_i(x) = (Q_i x|x) + (c_i|x) - \alpha_i$ and $G(x) = Ax - u$, the cone C is the negative orthant in $\mathbf{R^k}$, and $e = (-1, \ldots, -1)$. The verification that π is continuous in the just-defined topologies in $\mathcal{D}$ and $\mathcal{C}$ is straightforward.

We have in this case $R_0 = 0$. (A_1) is clear, (A_2) holds with $q(x) = \|x\|^2$, (A_3) holds by Lemma 3.5, and so it remains to verify (A_4). It is clear that $\mathcal{M} \neq \emptyset$. We define the set $\hat{\mathcal{M}}$ by replacing every $a \in \mathcal{M}$ by a sequence a_n of data, each having the same components as a with the exception of the first component, which for a_n is $Q + (1/n)I$, where I is the identity map and Q is the first component of a. Clearly, the collection of data so obtained is dense in $\mathcal{M}$. The feasible set $\mathrm{dom} f_{a_n}$ is the same as $\mathrm{dom} f_a$, and $f_{a_n}(x) = f_a(x) + (1/n)\|x\|^2$, $x \in X$. This function is coercive, and an argument similar to that used in the proof of (A_4) in Example 3.4 gives the desired conclusion.

Remark. Note that our argument does not apply for the "pure" quadratic problem of minimizing $(Qx|x)$ subject to $(Q_ix|x) \le \alpha_i$; condition (b) in (A_2) forces us to add the linear term in the cost function. But the argument applies without changes to problems

$$\text{minimize} \quad (Qx|x) + (c|x) \quad \text{subject to} \quad (Q_ix|x) \le \alpha_i.$$

Example 3.7 (Semi-infinite programming). Let $X = \mathbf{R^n}$, let T be a Hausdorff compact space, and consider the problem:

$$\text{minimize} \quad (Ax|x) + (c|x) \quad \text{subject to} \quad (B(t)|x) + b(t) \le 0,$$

where A is a real, symmetric $n \times n$ matrix, while $B : T \to \mathbf{R}^n$, $b : T \to \mathbf{R}$ are continuous functions on X. With the previous notations, the Banach space Y is the space $C(T)$ of the continuous functions from T to the reals (with the usual max norm), C is the cone of the functions which are nonpositive everywhere, and e is such that $e(t) = 1$ for all $t \in T$. The cost-constraint space is $\mathcal{C} := C^0(X) \times C^0(X, C(T))$ with the topology of uniform convergence on the bounded sets. The data space consists of 4-tuples $a = (A, B, c, b)$, considered with some product metric generated by the usual metrics on matrices A and vectors c and by the sup-norms for B in $C(T, \mathbf{R}^n)$ and for b in $C(T)$. The mapping π is defined by $\pi(a) = (f, F)$ with $f(x) = (Ax|x) + (c|x)$, and $F(x) = (B(\cdot)x|x) + b(\cdot)$, and is continuous. In this example $R_0 = 0$, and (A_1) is easily verified. As function q we take $q(x) = ||x||^2$, and (A_2) is verified as well. For (A_4), we can use exactly the same argument as in Examples 3.4 and 3.6.

Example 3.8 (Linear optimal control with quadratic cost). This is the class of problems covered by the following scheme:

$$\begin{aligned} &\text{minimize} \quad \int_0^1 [(Pu(t)|u(t)) + (Qx(t)|x(t)) + (c_0(t)|u(t)) + (b(t)|x(t))]dt \\ &\text{subject to} \quad \dot{x}(t) = Ax(t) + Bu(t), \quad x(0) = x_0,\ x(1) = x_1. \end{aligned}$$

Here $(\cdot|\cdot)$ stands for the inner product in the corresponding space; P and Q are symmetric matrices of orders m and n, respectively, with P positive semidefinite; A is a square matrix of order n; B is a matrix $n \times m$; x_0, $x_1 \in \mathbf{R}^n$; and $c_0(t)$ and $b(t)$ are square integrable mappings from $[0, 1]$ into $\mathbf{R}^m$ and $\mathbf{R}^n$, respectively. Thus the data space of the problem consists of 8-tuples $a = (P, Q, c_0(\cdot), b(\cdot), A, B, x_0, x_1)$; that is to say, it can be identified with the product $SL^+(m) \times SL(n) \times L_2^m(0, 1) \times L_2^n(0, 1) \times L(n) \times L(m, n) \times \mathbf{R}^n \times \mathbf{R}^n$ endowed with a natural product metric.

Next, we have to identify the domain and the cost-constraint spaces. To this end, we first note that, given a data tuple a, the trajectory $x(t)$ is completely defined by the control $u(t)$:

$$x(t) = e^{At}\left[x_0 + \int_0^t e^{-A\tau}Bu(\tau)d\tau\right]. \tag{3.4}$$

This means that the space $L_2^m(0, 1)$ of $\mathbf{R}^m$-valued square integrable functions with the standard norm can be considered as the domain space. Substituting (3.4) into the functional and the constraint, we reduce the problem to the standard form

$$\text{minimize} \quad f(u(\cdot)) \quad \text{subject to} \quad G(u(\cdot)) = 0,$$

with

$$f(u(\cdot)) = \int_0^1 [(Pu(t)|u(t)) + (c(t)|u(t))]dt + \mathcal{K}(u(\cdot)) \tag{3.5}$$

and

$$G(u(\cdot)) = x_0 - e^{-A}x_1 + \int_0^1 e^{-At}Bu(t)dt, \tag{3.6}$$

where

$$c(t) = c_0(t) + B^*e^{-A^*t}\int_t^1 \left[e^{A\tau}b(\tau) + 2Qe^{A\tau}x_0\right]d\tau, \tag{3.7}$$

$\mathcal{K}$ is a quadratic functional on $L_2^m(0,1)$ defined by

$$\mathcal{K}(u(\cdot)) = \int_0^1 (QK[u(\cdot)](t)|K[u(\cdot)](t))dt, \tag{3.8}$$

and K is the following integral operator in $L_2^m(0,1)$:

$$K[u(\cdot)](t) = e^{At}\int_0^t e^{-A\tau}Bu(\tau)d\tau. \tag{3.9}$$

(In other words, $K[u(\cdot)](t)$ is the solution of the equation $\dot{x} = Ax + Bu$ with zero initial condition.)

Accordingly, we define the cost-constraint space as the product of the space of continuous functions on $L_2^m(0,1)$ and the space of continuous mappings from $L_2^m(0,1)$ into $\mathbf{R}^n$ with the topology of uniform convergence on bounded sets. Equalities (3.4)–(3.9) above define the mapping π from the data space to the cost-constraint space, which is obviously continuous. Here again the index R_0 is zero, and it follows immediately that (A_1) is satisfied. The second assumption (A_2) is also easy to verify: It is enough to take $q(u(\cdot)) = \|u(\cdot)\|^2 = \int_0^1 |u(t)|^2dt$. Then the element $\tilde{a} = \tilde{a}(t,\gamma,w,z)$ corresponds, for example, to

$$\begin{array}{llll} \tilde{P} = P + (t/\gamma)I, & \tilde{Q} = Q, & \tilde{c}_0(t) = c_0(t) + (t/\gamma)Pw(t), & \tilde{b}(t) = b(t), \\ \tilde{A} = A, & \tilde{B} = B, & \tilde{x}_0 = x_0 + z, & \tilde{x}_1 = x_1. \end{array}$$

Verification of the other two assumptions requires a bit more work. Let us start with (A_3). Fix some data a and suppose that $G(u(\cdot)) = 0$. Consider the operator $\Lambda_a : L_2^m(0,1) \to \mathbf{R}^n$ defined by

$$\Lambda_a(u(\cdot)) = K[u(\cdot)](1) = e^A\int_0^1 e^{-At}Bu(t)dt.$$

Observe that

$$G(u) = e^{-A}(\Lambda_a(u) - x_1) + x_0.$$

Assume for the moment that $m = 1$, that is, that B is a one column matrix. As is well known, in this case Λ_a is onto if and only if the vectors $B, AB, A^2B, \ldots, A^{n-1}B$ are linearly independent. It is clear that the collection of all pairs $(A,B) \in L(n) \times \mathbf{R}^n$

with this property is open and dense in $L(n) \times \mathbf{R}^n$. Coming back to the general case, suppose that Λ_a is not onto. Then, appealing to the previous remark, we can construct a new element b, close to a, in the following way. We start by slightly changing A and B to make Λ_b onto, and then we correct x_0 in order to satisfy the equality $\hat{G}(u(\cdot)) = 0$, where by $\hat{G}$ we denote the equality constraint mapping relative to b. In such a way we are able to apply Lemma 3.5 to the new data and this proves (A_3).

Let us prove (A_4). It is clear that $\mathcal{M} \neq \emptyset$. For any a denote by q_a the quadratic component of the cost function

$$q_a(u(\cdot)) = \int_0^1 (Pu(t)|u(t))dt + \mathcal{K}(u(\cdot)),$$

and by S_a the intersection of $\mathrm{Ker}\Lambda_a$ with the unit sphere in $L_2^m(0,1)$. Finally, set

$$r_a = \inf_{S_a} q_a(u(\cdot)).$$

Then $r_a > 0$ means that q_a is positive definite on $\mathrm{Ker}\Lambda_a$.

We define $\hat{\mathcal{M}} \subset \mathcal{M}$ as the collection of all data $a \in \mathcal{M}$ with Λ_a onto and $r_a > 0$.

To prove that $\hat{\mathcal{M}}$ is dense in $\mathcal{M}$ we first note that $r_a \geq 0$ for any $a \in \mathcal{M}$ (otherwise $\inf f_a = -\infty$). It follows that the collection of $a \in \mathcal{M}$ with $r_a > 0$ is dense in $\mathcal{M}$. (Indeed, replace, if necessary, P by $P + \varepsilon I$.) Now to see that $\hat{\mathcal{M}}$ is dense in $\mathcal{M}$ we only need to verify that r_a as a function of a is lower semicontinuous. Indeed, in this case, given an $a \in \mathcal{M}$ with $r_a > 0$, we can slightly change a to make Λ_a onto (as was done above) and keep r_a positive.

The lower semicontinuity property of r_a follows in turn from the estimate

$$\mathrm{dist}(u(\cdot), \mathrm{Ker}\Lambda_a) \leq M\|\Lambda_a(u(\cdot)\| \quad \forall\, u(\cdot) \in L_2^m(0,1) \tag{3.10}$$

(with some positive M), following from the fact that the orthogonal complement of $\mathrm{Ker}\Lambda_a$ is isomorphic to the image of Λ_a (as the latter, being a subspace of a finite dimensional space, is closed).

Assume now that $a_n \to a$ and $u_n(\cdot)$ are such that $\|u_n(\cdot)\| = 1$, $\Lambda_{a_n}(u_n(\cdot)) = 0$, and $q_{a_n}(u_n(\cdot)) \leq r_{a_n} + \varepsilon_n$, where $\varepsilon_n \to 0$. By (3.10), $\|\Lambda_a(u_n(\cdot))\| \to 0$ and therefore there are $v_n(\cdot) \in \mathrm{Ker}\Lambda_a$ such that $\|v_n(\cdot) - u_n(\cdot)\| \to 0$. We can assume that the norms of all $v_n(\cdot)$ are also equal to one, and so $q_a(v_n(\cdot)) \geq r_a$. On the other hand, it is obvious that $q_{a_n}(u_n(\cdot)) - q_a(v_n(\cdot)) \to 0$ (because of the equicontinuity of the functions on bounded subsets), whence $\liminf r_{a_n} \geq r_a$.

Let $a \in \hat{\mathcal{M}}$ and set $\pi(a) = (f, G)$. We can choose $\delta > 0$ so small that $r_b \geq \varepsilon > 0$ for all b with $d(a,b) < \delta$. Then $f_b(u(\cdot)) \to \infty$ as $\|u(\cdot)\| \to \infty$ uniformly on the δ-ball around a. Therefore it is possible to find an R so big that

$$f_b(u(\cdot)) \geq \inf f_a + 1 \quad \text{if } \|u(\cdot)\| \geq R/2,\ d(a,b) < \delta. \tag{3.11}$$

Next choose an arbitrary $\xi \in (0, 1/4)$ and, if necessary, take a smaller δ to make sure that the following hold whenever $d(a,b) < \delta$:

(a) For any $u(\cdot)$ with $\|u(\cdot)\| < R/2$ there is a $v(\cdot) \in L_2^m(0,1)$ such that $\|v(\cdot) - u(\cdot)\| < \xi$ with $v \in \mathrm{dom}\, f_b$. This is possible by Lemma 3.5 as Λ_a is onto.

(b) $|f(u(\cdot)) - \hat{f}(u(\cdot))| < \xi$ if $\|u(\cdot)\| \leq R$, where $\hat{f}$ is the cost component of $\pi(b)$. This is clearly possible, as the cost function (as a function of the data) is continuous uniformly on the bounded subsets of $L_2^m(0,1)$.

Observe that from (a) and (b) it also follows that $\inf f_b < \inf f_a + 1/2$ for all b such that $d(a,b) < \delta$ (taking again a smaller δ, if necessary). Finally, let $\mathcal{N}$ be the δ-ball around a. It follows from (3.11)(a) and (b) that

$$\inf f_b < \inf\{f_a(u(\cdot)) : \|u(\cdot)\| \le R\} + 2\xi \le \inf f_a + 1/2$$

for any $b \in \mathcal{N}$. On the other hand, if $\|u(\cdot)\| \ge R$, then

$$f_b(u(\cdot)) \ge \inf f_a + 1 > \inf f_b + 1/2,$$

so that for $u(\cdot)$ with $f_b(u(\cdot)) \le \inf f_b + \xi$ we have $\|u(\cdot)\| < R$. This completes verification of (A_4).

4. Proofs. In this section we prove our main Theorem 2.2 as well as the auxiliary Proposition 3.2. Before we start, we need two conventions which will be used several times in what follows.

First, since π is continuous, the following property holds. Let R_0 be the index of the topology in the cost-constraint space. Then for every $R \ge R_0$, $a \in \mathcal{D}$, and $\sigma > 0$ there exists $r > 0$ so that

$$\begin{gathered} b \in \mathcal{D} \quad \text{and} \quad d(a,b) < r \quad \text{implies} \\ \sup\{|f(x) - \hat{f}(x)| + \|F(x) - \hat{F}(x)\| + \|G(x) - \hat{G}(x)\| : \|x\| \le R\} < \sigma, \end{gathered} \tag{4.1}$$

where $\pi(a) = (f, F, G)$ and $\pi(b) = (\hat{f}, \hat{F}, \hat{G})$.

Second, we may also assume, without loss of generality, that the unit ball in Y around e belongs to the cone C:

$$B(e,1) \subset C \qquad (\text{and hence } B(y + te, t) \subset C \quad \forall\, y \in C). \tag{4.2}$$

In any of the examples given in section 3 this last condition was fulfilled.

We continue now with the proof of Proposition 3.2. For this we need the following lemma.

LEMMA 4.1. *Assume* (A_1)–(A_3). *Let* $\bar{a} \in \mathcal{F}$, $\bar{x} \in \mathrm{dom} f_{\bar{a}}$, *and* $\varepsilon > 0$. *Then there is a nonempty open set* $\mathcal{N}$ *of* $\mathcal{D}$ *such that for every* $a \in \mathcal{N}$

(i) $d(\bar{a}, a) < \varepsilon$;

(ii) *there is* $u \in \mathrm{dom} f_a$ *such that* $||u - \bar{x}|| < \varepsilon$.

Proof. Let $\pi(\bar{a}) = (\bar{f}, \bar{F}, \bar{G})$. By (A_2) there is $t \in (0, \varepsilon/2)$ such that $\pi(a_t) = (\bar{f}, \bar{F}_t, \bar{G})$ for some $a_t \in \mathcal{D}$ with $d(a, a_t) < \varepsilon/2$.

Fix such t and a_t, and using the continuity of $\bar{F}$, let $\delta \in (0, t/2)$ be so that $\|\bar{F}(x) - \bar{F}(\bar{x})\| < t/2$ if $\|x - \bar{x}\| < \delta$. We may think that δ is also chosen in order that (4.1) holds for $R := \max\{R_0, \|\bar{x}\| + t\}$, $a = a_t$, $\sigma = t/2$, and $r = \delta$. Further, since $\bar{x} \in \mathrm{dom} f_{a_t}$ (remember that $\bar{F}_t(\bar{x}) = \bar{F}(\bar{x}) + te$), by (A_3) (applied to a_t and δ) there is a nonempty open set $U \subset \mathcal{C}_=$ so that for any $G \in U$ we can find $a \in \mathcal{D}$ and $u \in X$ with $\|u - \bar{x}\| < \delta$, $G(u) = 0$, $\pi(a) = (f, F, G)$, and $d(a, a_t) < \delta$. In particular, for any such a, $\|F(x) - \bar{F}_t(x)\| < t/2$ for every $x \in X$ with $\|x\| \le R$. But then

$$\|F(u) - \bar{F}_t(\bar{x})\| \le \|F(u) - \bar{F}_t(u)\| + \|\bar{F}_t(u) - \bar{F}_t(\bar{x})\| < t/2 + \|\bar{F}(u) - \bar{F}(\bar{x})\| < t,$$

which by (4.2) implies that $F(u) \in C$ and, consequently, $u \in \mathrm{dom} f_a$.

The above means that the set

$$\mathcal{N} = \{a \in \mathcal{D} : \pi(a) = (f, F, G),\ G \in U,\ d(a, a_t) < \delta\}$$

is nonempty and for every $a \in \mathcal{N}$ there is $u \in \mathrm{dom} f_a$ such that $\|u - \bar{x}\| < \varepsilon$. Since by construction any $a \in \mathcal{N}$ obviously satisfies part (i) of the lemma, it remains only to mention that $\mathcal{N}$ is open because of the continuity of π. The proof is complete. □

Proof of Proposition 3.2. For every $\bar{a} \in \mathcal{F}$, fix some $\bar{x} \in \mathrm{dom} f_{\bar{a}}$. Let $\bar{f}$ be the cost component of $\pi(\bar{a})$. Since $\bar{f}$ is continuous at $\bar{x}$, there is some $\bar{\varepsilon} \in (0,1)$ such that $\|x - \bar{x}\| < \bar{\varepsilon}$ implies $|\bar{f}(x) - \bar{f}(\bar{x})| \le 1$. Let further

$$\hat{\mathcal{F}} := \bigcup \{\mathcal{N}_{\bar{a},\bar{x},\varepsilon} : \bar{a} \in \mathcal{F}, 0 < \varepsilon \le \bar{\varepsilon}\},$$

where the set $\mathcal{N}_{\bar{a},\bar{x},\varepsilon}$ is given by Lemma 4.1. By condition (ii) of this lemma, $\hat{\mathcal{F}}$ lies entirely in $\mathcal{F}$. Obviously $\hat{\mathcal{F}}$ is open in $\mathcal{D}$ and dense in $\mathcal{F}$. Take some $a \in \hat{\mathcal{F}}$ and let $a \in \mathcal{N} := \mathcal{N}_{\bar{a},\bar{x},\varepsilon}$ for some $\bar{a}, \bar{x}$ and $0 < \varepsilon \le \bar{\varepsilon}$ as above. Set $K := \bar{f}(\bar{x}) + 2$, take an arbitrary $b \in \mathcal{N}$, and let $u \in \mathrm{dom} f_b$ be the point provided by Lemma 4.1. We have $\|u - \bar{x}\| < \varepsilon \le \bar{\varepsilon} < 1$. We may think that, given $\bar{a}$, $\bar{x}$, the number $\bar{\varepsilon}$ was chosen so that (4.1) holds for $R := \max\{R_0, \|\bar{x}\| + 1\}$, $a = \bar{a}$, $\sigma = 1$, and $r = \bar{\varepsilon}$. Hence we get

$$\inf f_b \le f_b(u) \le \bar{f}(u) + 1 \le \bar{f}(\bar{x}) + 2 = K,$$

which completes the proof. □

COROLLARY 4.2. *Suppose* (A_1)–(A_4) *and* $\mathcal{M} \neq \emptyset$. *Then* $\mathcal{M}$ *contains a dense set* $\mathcal{O}$ *which is open in* $\mathcal{D}$. *In particular,* $(\mathcal{M}, d)$ *is a Baire space.*

Proof. Let $\mathcal{O}' := \cup\{\mathcal{N}_a : a \in \hat{\mathcal{M}}\}$, where $\hat{\mathcal{M}}$ is the set provided by (A_4) and for each $a \in \hat{\mathcal{M}}$, $\mathcal{N}_a$ is the open set from the same assumption. Set $\mathcal{O} := \mathcal{O}' \cap \hat{\mathcal{F}}$, where the set $\hat{\mathcal{F}}$ is given by Proposition 3.2. By (a) of (A_4), we have $\mathcal{O} \subset \mathcal{M}$. Clearly, $\mathcal{O}$ is open in $\mathcal{D}$ and dense in $\mathcal{M}$. □

Further, in order to prove the main result, we need the following auxiliary lemma.

LEMMA 4.3. *Assume* (A_1) *and* (A_4). *Then for any* $a \in \hat{\mathcal{M}}$ (*the set from* (A_4)) *there is a ball* $\mathcal{U}$ *around* a *and a real number* N *such that* $\inf f_b \ge N$ *for all* $b \in \mathcal{U}$.

Proof. If the index $R_0 = +\infty$, the conclusion is immediate. So suppose $R_0 = 0$ and $a \in \hat{\mathcal{M}}$. Then by (A_4) there exist an open set $\mathcal{N} = \mathcal{N}_a$ containing a, $\xi > 0$, and $R > 0$ such that for any $b \in \mathcal{N}$, $\inf f_b > -\infty$ and $\|x\| < R$ if $\hat{f}(x) < \inf f_b + \xi$, where $\hat{f}$ is the cost component of $\pi(b)$.

Let f be the cost component of $\pi(a)$. By (A_1), f is bounded below on the ball of radius R. So let $N := \inf_{\|x\| \le R} f - \xi - 1$. Take a sufficiently small ball $\mathcal{U}$ around a contained in $\mathcal{N}$ to be sure (by (4.1) applied to R, a, and $\sigma = 1$) that $|f(x) - \hat{f}(x)| \le 1$ if $\|x\| \le R$. Let $b \in \mathcal{U}$. If $\mathrm{dom} f_b = \emptyset$, then we obviously have $\inf f_b > N$. So suppose $\mathrm{dom} f_b \neq \emptyset$. Then (since $\inf f_b > -\infty$) there is some $x \in \mathrm{dom} f_b$ with $f_b(x) = \hat{f}(x) < \inf f_b + \xi$. This entails $\|x\| \le R$, and therefore

$$\inf f_b > \hat{f}(x) - \xi \ge f(x) - \xi - 1 \ge \inf_{\|x\| \le R} f(x) - \xi - 1 = N,$$

as claimed. □

Now we move to the proof of Theorem 2.2. Suppose $\mathcal{M} \neq \emptyset$. Then, by Corollary 4.2, $(\mathcal{M}, d)$ is a Baire space. Therefore, to derive the conclusion of our main result we will use Theorem 1.1. We have to verify that the hypothesis $(\mathcal{H})$ holds for $(\mathcal{M}, d)$. To this end the proof will be divided into several steps which correspond to the construction of the ingredients of $(\mathcal{H})$ and the verification of its conditions (i) and (ii).

Step 1: *Construction of* $\mathcal{B}$. We set $\mathcal{B} = \hat{\mathcal{M}} \cap \mathcal{O}$, where $\hat{\mathcal{M}}$ is the dense set provided by (A_4) and $\mathcal{O}$ is the set given by Corollary 4.2, which is open in $\mathcal{D}$ and dense in $\mathcal{M}$. Clearly, $\mathcal{B}$ is dense in $\mathcal{M}$.

In what follows, we fix $a \in \mathcal{B}$, $\varepsilon > 0$, and $\gamma > 0$ and set $\pi(a) = (f, F, G)$.

Step 2: *Choice of* $\bar{x}$. For any collection $s = (t, \tau, \gamma, w, z)$, $t, \tau > 0$, $\gamma > 0$, $w \in X$, $z \in Z$, we put $b_s = \tilde{a}(t, \tau, \gamma, w, z)$—the element introduced in (A_2). Remember that $\pi(b_s) = (f_{t\gamma w}, F_t, G_z)$, where $f_{t\gamma w}(\cdot) = f(\cdot) + tq(\gamma^{-1}(\cdot - w))$, $F_t(\cdot) = F(\cdot) + te$, and $G_z(\cdot) = G(\cdot) + z$. Further, given $\eta > 0$, let

$$A_\eta := \{x \in X : F_t(x) \in C \text{ for some } t \in (0, \eta);\ \|G(x)\| < \eta\}.$$

We shall choose a sufficiently small $\eta > 0$ to make sure, in particular, that f is bounded below on A_η (this is automatic in the case $R_0 = \infty$) and then fix as $\bar{x}$ any point of A_η satisfying

$$f(\bar{x}) \le \inf_{x \in A_\eta} f(x) + \eta/5. \tag{4.3}$$

We show how to find a suitable η. First, since $a \in \hat{\mathcal{M}} \cap \mathcal{O}$, we have from (A_4) and Lemma 4.3 that there exist an open set $\mathcal{N} \subset \mathcal{D}$ containing a (we may think that $\mathcal{N} \subset \mathcal{M}$, because $a \in \mathcal{O}$), $\xi > 0$, $R \ge R_0$, and a real number N such that whenever $\hat{a} \in \mathcal{N}$ one has

$$\begin{array}{ll} \text{(i)} & d(a, \hat{a}) < \varepsilon; \\ \text{(ii)} & f_{\hat{a}}(x) < \inf f_{\hat{a}} + \xi \quad \text{implies} \quad \|x\| < R; \\ \text{(iii)} & \inf f_{\hat{a}} \ge N. \end{array} \tag{4.4}$$

Now choose $\eta > 0$ to satisfy the following requirements:

$$\begin{array}{ll} \text{(a)} & \eta < \min\left\{\dfrac{\varepsilon}{2}, \gamma, \xi\right\}; \\ \text{(b)} & t, \tau, \|z\| \le \eta \quad \text{and} \quad w \in X \quad \text{with} \quad \|w\| \le R \quad \text{implies} \quad b_s \in \mathcal{N}. \end{array} \tag{4.5}$$

The choice of η to satisfy (b) is ensured by (A_2)(b).

To see that f is bounded below on A_η, take any $x \in A_\eta$. Then $F_\tau(x) \in C$ for some $\tau < \eta$ and $\|G(x)\| < \eta$. Set $z = -G(x)$ and take b such that $\pi(b) = (f, F_\tau, G_z)$. Clearly, $x \in \operatorname{dom} f_b$, and, on the other hand, $b \in \mathcal{N}$ by (4.5)(b). Therefore, by (4.4)(iii), $f(x) \ge \inf f_b \ge N$.

This shows that the choice of $\bar{x}$ as in (4.3) is correct. We claim finally that

$$\|\bar{x}\| < R. \tag{4.6}$$

Indeed, set $\bar{z} = -G(\bar{x})$ and let $\bar{t} \in (0, \eta)$ be so that $F_{\bar{t}}(\bar{x}) \in C$. We see, as above, that $\bar{x} \in \operatorname{dom} f_{\bar{b}}$, where $\bar{b} \in \mathcal{N}$ is such that $\pi(\bar{b}) = (f, F_{\bar{t}}, G_{\bar{z}})$. Moreover, $\operatorname{dom} f_{\bar{b}} \subset A_\eta$, and hence $\inf f_{\bar{b}} \ge \inf f_{A_\eta}$. Therefore, by (4.3) and (4.5)(a) we have $f(\bar{x}) < \inf f_{\bar{b}} + \xi$, whence (4.6).

Step 3: *Construction of* $\mathcal{V}$. We first define an element $\bar{a} \in \mathcal{N}$, around which the set $\mathcal{V}$ will be built. Namely, as above let $\bar{z} = -G(\bar{x})$ and take $\bar{t}$ so that $(4/5)\eta < \bar{t} < \eta$ and $F_{\bar{t}}(\bar{x}) \in C$. Now set $\bar{a} = b_{\bar{s}}$ with $\bar{s} = (\bar{t}, \bar{t}, \gamma, \bar{x}, \bar{z})$. We have $\pi(\bar{a}) = (\bar{f}, \bar{F}, \bar{G})$, where

$$\bar{f}(x) = f_{\bar{t}\gamma\bar{x}}, \quad \bar{F}(x) = F_{\bar{t}}(x), \quad \bar{G}(x) = G(x) + \bar{z}.$$

Since $\|G(\bar{x})\| < \eta$, by (4.5)(b) and (4.6) we deduce that $\bar{a} \in \mathcal{N}$. In particular, $\bar{a} \in \mathcal{M}$. Observe also that $\bar{x} \in \operatorname{dom} f_{\bar{a}}$. Further, we have by (A_2)

$$\begin{array}{l} \bar{f}(x) \ge f(x) \quad \forall\, x, \\ \bar{f}(\bar{x}) = f(\bar{x}), \\ \bar{f}(x) \ge f(x) + \bar{t} \quad \text{if} \quad \|x - \bar{x}\| \ge \gamma. \end{array} \tag{4.7}$$

Next we choose a positive δ so small that any b with $d(\bar a, b) < \delta$ belongs to $\mathcal{N}$ and

$$\begin{aligned} &\delta < \min\Big\{\frac{\eta}{10}, \eta - \bar t, \eta - \|G(\bar x)\|, R - \|\bar x\|\Big\}, \\ &\|x - \bar x\| < \delta \Rightarrow |\bar f(x) - \bar f(\bar x)| < \frac{\eta}{10}, \quad \text{and} \quad y \in C, \quad \text{provided} \quad \|y - \bar F(x)\| < \delta. \end{aligned} \tag{4.8}$$

The choice of δ as in the second row is possible because of the continuity of $\bar f$ and $\bar F$ and (4.2). Let $r \in (0, \delta)$ be so small that (4.1) is satisfied for R, $\bar a$, and $\sigma = \delta$. By (A_3), applied for $\bar a$, $\bar x$, and r, there is a nonempty open set U of $\mathcal{C}_=$ such that for every $\hat G \in U$ there is $u \in X$ with $\|u - \bar x\| < r < \delta$ and $\hat G(u) = 0$. We now define $\mathcal{V}$ as follows:

$$\mathcal{V} = \{b \in \mathcal{D} : \pi(b) = (\hat f, \hat F, \hat G) : d(\bar a, b) < r,\ \hat G \in U\}.$$

Then $\mathcal{V} \neq \emptyset$ by (A_3), $\mathcal{V}$ is open (in $\mathcal{D}$) because π is continuous, and $\mathcal{V}$ lies in $\mathcal{N}$ and hence in $\mathcal{M}$.

With $\bar x$ and $\mathcal{V}$ so defined, we verify (i) and (ii) of ($\mathcal{H}$) for certain λ and α given below.

Step 4: *Verification of* (i). This is immediate from (4.4) and the fact that $\mathcal{V} \subset \mathcal{N}$.

Step 5: *Verification of* (ii). Take an arbitrary $b \in \mathcal{V}$ and let $\pi(b) = (\hat f, \hat F, \hat G)$. Then there is $u \in X$ with $\|u - \bar x\| < \delta$ such that $\hat G(u) = 0$. We have $\|u\| < R$ by the definition of δ, and hence by (4.1) and (4.8) we get $\hat F(u) \in C$. Thus $u \in \operatorname{dom} f_b$ and, consecutively using (4.1), (4.8), and (4.7), we get

$$\inf f_b \le \hat f(u) < \bar f(u) + \frac{\eta}{10} < \bar f(\bar x) + \frac{\eta}{5} = f(\bar x) + \frac{\eta}{5}. \tag{4.9}$$

Next, let $w \in X$ be such that

$$f_b(w) \le \inf f_b + \frac{\eta}{5}. \tag{4.10}$$

By (4.4)(ii), $\|w\| < R$, since $b \in \mathcal{N}$ and $\eta/5 < \xi$. We further observe that $w \in A_\eta$. Indeed, by the definition of $\bar G$, $G(x) = \bar G(x) + G(\bar x)$ for all $x \in X$. This, together with the inequality $\|\bar G(w) - \hat G(w)\| < \delta$ (which follows from (4.1)), (4.8), and the definition of $\mathcal{V}$, gives (as $\hat G(w) = 0$)

$$\|G(w)\| \le \|G(\bar x)\| + \|\bar G(w)\| = \|G(\bar x)\| + \|\bar G(w) - \hat G(w)\| < \|G(\bar x)\| + \delta < \eta.$$

Likewise, as $\hat F(w) \in C$, it follows from (4.1) and the definition of $\mathcal{V}$ that $F(w) + (\bar t + \delta)e \in C$. Since, by (4.8), $\bar t + \delta < \eta$, we conclude that $w \in A_\eta$.

Finally, we claim that actually $\|w - \bar x\| \le \gamma$. If this were not true, then we would have by (4.7), $\bar f(w) \ge f(w) + \bar t$, so that by (4.10), (4.1), the fact that $\delta < \eta/5$, the choice of $\bar t$, and (4.3), we get

$$\begin{aligned} \inf f_b &\ge \hat f(w) - \eta/5 \ge \bar f(w) - 2\eta/5 \\ &\ge f(w) - 2\eta/5 + \bar t > \inf_{A_\eta} f + 2\eta/5 \ge f(\bar x) + \eta/5, \end{aligned} \tag{4.11}$$

in contradiction with (4.9).

Furthermore, as follows from (4.9) and (4.10),

$$\hat f(w) < f(\bar x) + 2\eta/5,$$

whereas a calculation similar to that in (4.11) (but using the first inequality in (4.7)) gives

$$\hat{f}(w) \geq \bar{f}(w) - \eta/5 \geq f(w) - \eta/5 \geq \inf_{A_\eta} f - \eta/5 \geq f(\bar{x}) - 2\eta/5.$$

Thus

$$|\hat{f}(w) - f(\bar{x})| \leq 2\eta/5 < \gamma,$$

which shows that (ii) holds with $\alpha = f(\bar{x})$ and $\lambda = \eta/5$. This completes the proof of the theorem. □

Acknowledgments. We thank two anonymous referees for their comments, which helped us to improve the presentation of the results.

REFERENCES

[BL] G. Beer and R. Lucchetti, *The epi-distance topology: Continuity and stability results with application to convex optimization problems*, Math. Oper. Res., 17 (1992), pp. 715–726.

[CK] M. M. Čoban and P. S. Kenderov, *Generic Gâteaux differentiability of convex functionals in $C(T)$ and the topological properties of T*, in Mathematics and Education in Mathematics, Proceedings of the XVth Spring Conference of the Union of Bulgarian Mathematicians, Sunny Beach, Bulgaria, 1986, pp. 141–149.

[CKR1] M. M. Čoban, P. S. Kenderov, and J. P. Revalski, *Generic well-posedness of optimization problems in topological spaces*, Mathematika, 36 (1989), pp. 301–324.

[CKR2] M. M. Čoban, P. S. Kenderov, and J. P. Revalski, *Densely defined selections of multivalued mappings*, Trans. Amer. Math. Soc., 344 (1994), pp. 533–552.

[DGZ] R. Deville, G. Godefroy, and V. Zizler, *Smoothness and Renormings in Banach Spaces*, Pitman Monographs and Surveys in Pure and Applied Mathematics, Longman Scientific and Technical, Harlow, UK, 1993.

[DR] R. Deville and J. P. Revalski, *Porosity of ill-posed problems*, Proc. Amer. Math. Soc., 128 (2000), pp. 1117–1124.

[DuMO] A. V. Dmitruk, A. A. Miljutin, and N. P. Osmolovskii, *Ljusternik's theorem and the theory of extrema*, Russian Math. Surveys, 35 (1980), pp. 11–51.

[DoZo] A. Dontchev and T. Zolezzi, *Well-posed optimization problems*, Lecture Notes in Math. 1354, Springer-Verlag, Berlin, 1993.

[IL] A. Ioffe and R. Lucchetti, *Generic existence, uniqueness and stability in optimization*, in Nonlinear Optimization and Related Topics, G. Di Pillo and F. Giannessi, eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000, pp. 169–182.

[IT] A. D. Ioffe and V. M Tikhomirov, *Several remarks on variational principles*, Math. Zametki, 61 (1997), pp. 305–311 (in Russian); translated in Math. Notes, 61 (1997), pp. 248–253.

[IZa] A. D. Ioffe and A. J. Zaslavski, *Variational principles and well-posedness in optimization and calculus of variations*, SIAM J. Control Optim., 38 (2000), pp. 566–581.

[ILR] A. D. Ioffe, R. Lucchetti, and J. P. Revalski, *Generic well-posedness in mathematical programming*, C. R. Acad. Bulgare Sci., 54 (2000), pp. 17–20.

[KR] P. S. Kenderov and J. P. Revalski, *Generic well-posedness of optimization problems and the Banach-Mazur game*, in Recent Developments in Well-Posed Variational Problems, R. Lucchetti and J. P. Revalski, eds., Math. Appl. 331, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1995, pp. 117–136.

[LP] R. Lucchetti and F. Patrone, *Sulla densità e genericità di alcuni problemi di minimo ben posti*, Boll. Unione Mat. Ital., 15 (1978), pp. 225–240.

[R1] J. P. Revalski, *Generic properties concerning well-posed optimization problems*, C. R. Acad. Bulgare Sci., 38 (1985), pp. 1431–1434.

[R2] J. P. Revalski, *Generic properties in some classes of optimization problems*, Acta. Univ. Carolin. Math. Phys., 28 (1987), pp. 117–125.

[SRo] J. E. Spingarn and R. T. Rockafellar, *The generic nature of optimality conditions in nonlinear programming*, Math. Oper. Res., 4 (1979), pp. 425–430.

[Zo1] T. Zolezzi, *Well-posedness criteria in optimization with application to calculus of variations*, Nonlinear Anal., 25 (1995), pp. 437–453.

[Zo2] T. Zolezzi, *Extended well-posedness of optimization problems*, J. Optim. Theory Appl., 91 (1996), pp. 257–268.

SIAM J. OPTIM.
Vol. 12, No. 2, pp. 479–502

THE SAMPLE AVERAGE APPROXIMATION METHOD FOR STOCHASTIC DISCRETE OPTIMIZATION*

ANTON J. KLEYWEGT†, ALEXANDER SHAPIRO†, AND TITO HOMEM-DE-MELLO‡

Abstract. In this paper we study a Monte Carlo simulation–based approach to stochastic discrete optimization problems. The basic idea of such methods is that a random sample is generated and the expected value function is approximated by the corresponding sample average function. The obtained sample average optimization problem is solved, and the procedure is repeated several times until a stopping criterion is satisfied. We discuss convergence rates, stopping rules, and computational complexity of this procedure and present a numerical example for the stochastic knapsack problem.

Key words. stochastic programming, discrete optimization, Monte Carlo sampling, law of large numbers, large deviations theory, sample average approximation, stopping rules, stochastic knapsack problem

AMS subject classifications. 90C10, 90C15

PII. S1052623499363220

1. Introduction.

In this paper we consider optimization problems of the form

$$\min_{x\in\mathcal{S}} \{g(x) := \mathbb{E}_P G(x,W)\}\,. \tag{1.1}$$

Here W is a random vector having probability distribution P, $\mathcal{S}$ is a *finite* set (e.g., $\mathcal{S}$ can be a finite subset of $\mathbb{R}^n$ with integer coordinates), $G(x,w)$ is a real valued function of two (vector) variables x and w, and $\mathbb{E}_P G(x,W) = \int G(x,w)P(dw)$ is the corresponding expected value. We assume that the expected value function $g(x)$ is well defined, i.e., for every $x \in \mathcal{S}$ the function $G(x,\cdot)$ is measurable and $\mathbb{E}_P\{|G(x,W)|\} < \infty$.

We are particularly interested in problems with the following characteristics:

1. The expected value function $g(x) := \mathbb{E}_P G(x,W)$ cannot be written in a closed form, and/or its values cannot be easily calculated.
2. The function $G(x,w)$ is easily computable for given x and w.
3. The set $\mathcal{S}$ of feasible solutions, although finite, is very large, so that enumeration approaches are not feasible. For instance, in the example presented in section 4, $\mathcal{S} = \{0,1\}^k$ and hence $|\mathcal{S}| = 2^k$; i.e., the size of the feasible set grows exponentially with the number of variables.

It is well known that many discrete optimization problems are hard to solve. Another difficulty here is that the objective function $g(x)$ can be complicated and/or difficult to compute even approximately. Therefore stochastic discrete optimization problems are difficult indeed and little progress in solving such problems numerically has been reported so far. There is an extensive literature addressing stochastic discrete optimization problems in which the number of feasible solutions is sufficiently small to

*Received by the editors November 1, 1999; accepted for publication (in revised form) May 14, 2001; published electronically December 14, 2001.

http://www.siam.org/journals/siopt/12-2/36322.html

†School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0205 (Anton.Kleywegt@isye.gatech.edu, Alexander.Shapiro@isye.gatech.edu). The first author's work was supported by the National Science Foundation under grant DMI-9875400. The second author's work was supported by the National Science Foundation under grant DMS-0073770.

‡Department of Industrial, Welding and Systems Engineering, The Ohio State University, Columbus, OH 43210-1271 (homem-de-mello.1@osu.edu).

allow estimation of $g(x)$ for each solution x. Examples of this literature are Hochberg and Tamhane [12]; Bechhofer, Santner, and Goldsman [2]; Futschik and Pflug [7, 8]; and Nelson et al. [17]. Another approach that has been studied consists of modifying the well-known simulated annealing method in order to account for the fact that the objective function values are not known exactly. Work on this topic includes Gelfand and Mitter [9], Alrefaei and Andradóttir [1], Fox and Heine [6], Gutjahr and Pflug [10], and Homem-de-Mello [13]. A discussion of two-stage stochastic integer programming problems with recourse can be found in Birge and Louveaux [3]. A branch and bound approach to solving stochastic integer programming problems was suggested by Norkin, Ermoliev, and Ruszczyński [18] and Norkin, Pflug, and Ruszczyński [19]. Schultz, Stougie, and Van der Vlerk [20] suggested an algebraic approach to solving stochastic programs with integer recourse by using a framework of Gröbner basis reductions.

In this paper we study a Monte Carlo simulation–based approach to stochastic discrete optimization problems. The basic idea is simple indeed—a random sample of W is generated and the expected value function is approximated by the corresponding sample average function. The obtained sample average optimization problem is solved, and the procedure is repeated several times until a stopping criterion is satisfied. The idea of using sample average approximations for solving stochastic programs is a natural one and was used by various authors over the years. Such an approach was used in the context of a stochastic knapsack problem in a recent paper of Morton and Wood [16].

The organization of this paper is as follows. In the next section we discuss a statistical inference of the sample average approximation method. In particular, we show that with probability approaching 1 exponentially fast with increase of the sample size, an optimal solution of the sample average approximation problem provides an exact optimal solution of the "true" problem (1.1). In section 3 we outline an algorithm design for the sample average approximation approach to solving (1.1), and in particular we discuss various stopping rules. In section 4 we present a numerical example of the sample average approximation method applied to a stochastic knapsack problem, and section 5 gives conclusions.

2. Convergence results. As mentioned in the introduction, we are interested in solving stochastic discrete optimization problems of the form (1.1). Let $W^1, \ldots, W^N$ be an independently and identically distributed (i.i.d.) random sample of N realizations of the random vector W. Consider the sample average function

$$\hat{g}_N(x) := \frac{1}{N} \sum_{j=1}^{N} G(x, W^j)$$

and the associated problem

$$\min_{x \in \mathcal{S}} \ \hat{g}_N(x). \tag{2.1}$$

We refer to (1.1) and (2.1) as the "true" (or expected value) and sample average approximation (SAA) problems, respectively. Note that $\mathbb{E}[\hat{g}_N(x)] = g(x)$.

Since the feasible set $\mathcal{S}$ is finite, problems (1.1) and (2.1) have nonempty sets of optimal solutions, denoted $\mathcal{S}^*$ and $\hat{\mathcal{S}}_N$, respectively. Let v^* and $\hat{v}_N$ denote the optimal values,

$$v^* := \min_{x \in \mathcal{S}} g(x) \quad \text{and} \quad \hat{v}_N := \min_{x \in \mathcal{S}} \hat{g}_N(x),$$

of the respective problems. We also consider sets of ε-optimal solutions. That is, for $\varepsilon \geq 0$, we say that $\bar{x}$ is an ε-optimal solution of (1.1) if $\bar{x} \in \mathcal{S}$ and $g(\bar{x}) \leq v^* + \varepsilon$. The sets of all ε-optimal solutions of (1.1) and (2.1) are denoted by $\mathcal{S}^\varepsilon$ and $\hat{\mathcal{S}}_N^\varepsilon$, respectively. Clearly for $\varepsilon = 0$ set $\mathcal{S}^\varepsilon$ coincides with $\mathcal{S}^*$, and $\hat{\mathcal{S}}_N^\varepsilon$ coincides with $\hat{\mathcal{S}}_N$.

2.1. Convergence of objective values and solutions. The following proposition establishes convergence with probability one (w.p.1) of the above statistical estimators. By the statement "an event happens w.p.1 for N large enough" we mean that for P—almost every realization $\omega = \{W^1, W^2, \ldots\}$ of the random sequence—there exists an integer $N(\omega)$ such that the considered event happens for all samples $\{W^1, \ldots, W^n\}$ from ω with $n \geq N(\omega)$. Note that in such a statement the integer $N(\omega)$ depends on the sequence ω of realizations and therefore is random.

PROPOSITION 2.1. *The following two properties hold:* (i) $\hat{v}_N \to v^*$ *w.p.*1 *as* $N \to \infty$, *and* (ii) *for any* $\varepsilon \geq 0$ *the event* $\{\hat{\mathcal{S}}_N^\varepsilon \subset \mathcal{S}^\varepsilon\}$ *happens w.p.*1 *for* N *large enough.*

Proof. It follows from the (strong) law of large numbers that for any $x \in \mathcal{S}$, $\hat{g}_N(x)$ converges to $g(x)$ w.p.1 as $N \to \infty$. Since the set $\mathcal{S}$ is finite and the union of a finite number of sets each of measure zero also has measure zero, it follows that, w.p.1, $\hat{g}_N(x)$ converges to $g(x)$ uniformly in $x \in \mathcal{S}$. That is,

$$\delta_N := \max_{x \in \mathcal{S}} |\hat{g}_N(x) - g(x)| \to 0, \quad \text{w.p.1 as} \quad N \to \infty. \tag{2.2}$$

Since $|\hat{v}_N - v^*| \leq \delta_N$, it follows that, w.p.1, $\hat{v}_N \to v^*$ as $N \to \infty$.

For a given $\varepsilon \geq 0$ consider the number

$$\rho(\varepsilon) := \min_{x \in \mathcal{S} \setminus \mathcal{S}^\varepsilon} g(x) - v^* - \varepsilon. \tag{2.3}$$

Since for any $x \in \mathcal{S} \setminus \mathcal{S}^\varepsilon$ it holds that $g(x) > v^* + \varepsilon$ and the set $\mathcal{S}$ is finite, it follows that $\rho(\varepsilon) > 0$.

Let N be large enough such that $\delta_N < \rho(\varepsilon)/2$. Then $\hat{v}_N < v^* + \rho(\varepsilon)/2$, and for any $x \in \mathcal{S} \setminus \mathcal{S}^\varepsilon$ it holds that $\hat{g}_N(x) > v^* + \varepsilon + \rho(\varepsilon)/2$. It follows that if $x \in \mathcal{S} \setminus \mathcal{S}^\varepsilon$, then $\hat{g}_N(x) > \hat{v}_N + \varepsilon$ and hence x does not belong to the set $\hat{\mathcal{S}}_N^\varepsilon$. The inclusion $\hat{\mathcal{S}}_N^\varepsilon \subset \mathcal{S}^\varepsilon$ follows, which completes the proof. □

Note that if δ is a number such that $0 \leq \delta \leq \varepsilon$, then $\mathcal{S}^\delta \subset \mathcal{S}^\varepsilon$ and $\hat{\mathcal{S}}_N^\delta \subset \hat{\mathcal{S}}_N^\varepsilon$. Consequently it follows by the above proposition that for any $\delta \in [0, \varepsilon]$ the event $\{\hat{\mathcal{S}}_N^\delta \subset \mathcal{S}^\varepsilon\}$ happens w.p.1 for N large enough. It also follows that if $\mathcal{S}^\varepsilon = \{x^*\}$ is a singleton, then $\hat{\mathcal{S}}_N^\varepsilon = \{x^*\}$ w.p.1 for N large enough. In particular, if the true problem (1.1) has a unique optimal solution x^*, then w.p.1 for sufficiently large N the approximating problem (2.1) has a unique optimal solution $\hat{x}_N$ and $\hat{x}_N = x^*$. Also consider the set $A := \{g(x) - v^* : x \in \mathcal{S}\}$. The set A is a subset of the set $\mathbb{R}_+$ of nonnegative numbers and $|A| \leq |\mathcal{S}|$, and hence A is finite. It follows from the above analysis that for any $\varepsilon \in \mathbb{R}_+ \setminus A$ the event $\{\hat{\mathcal{S}}_N^\varepsilon = \mathcal{S}^\varepsilon\}$ happens w.p.1 for N large enough.

2.2. Convergence rates. The above results do not say anything about the rates of convergence of $\hat{v}_N$ and $\hat{\mathcal{S}}_N^\delta$ to their true counterparts. In this section we investigate such rates of convergence. By using the theory of large deviations (LD), we show that, under mild regularity conditions and $\delta \in [0, \varepsilon]$, the probability of the event $\{\hat{\mathcal{S}}_N^\delta \subset \mathcal{S}^\varepsilon\}$ approaches 1 exponentially fast as $N \to \infty$. Next we briefly outline some background of the LD theory.

Consider a random (real valued) variable X having mean $\mu := \mathbb{E}[X]$. Its moment-generating function $M(t) := \mathbb{E}[e^{tX}]$ is viewed as an extended valued function, i.e., it can take value $+\infty$. It holds that $M(t) > 0$ for all $t \in \mathbb{R}$, $M(0) = 1$, and the domain $\{t : M(t) < +\infty\}$ of the moment-generating function is an interval containing zero. The conjugate function

$$I(z) := \sup_{t \in \mathbb{R}} \{tz - \Lambda(t)\}, \tag{2.4}$$

of the logarithmic moment-generating function $\Lambda(t) := \log M(t)$, is called the (LD) *rate* function of X. It is possible to show that both functions $\Lambda(\cdot)$ and $I(\cdot)$ are convex.

Consider an i.i.d. sequence $X_1, \ldots, X_N$ of replications of the random variable X, and let $Z_N := N^{-1} \sum_{i=1}^{N} X_i$ be the corresponding sample average. Then for any real numbers a and $t \geq 0$ it holds that $P(Z_N \geq a) = P(e^{tZ_N} \geq e^{ta})$, and hence it follows from Chebyshev's inequality that

$$P(Z_N \geq a) \ \leq \ e^{-ta} \mathbb{E}\left[e^{tZ_N}\right] \ = \ e^{-ta}[M(t/N)]^N.$$

By taking the logarithm of both sides of the above inequality, changing variables $t' := t/N$, and minimizing over $t' \geq 0$, it follows for $a \geq \mu$ that

$$\frac{1}{N} \log\left[P(Z_N \geq a)\right] \ \leq \ -I(a). \tag{2.5}$$

Note that for $a \geq \mu$ it suffices to take the supremum in the definition (2.4) of $I(a)$ for $t \geq 0$, and therefore this constraint is omitted. Inequality (2.5) corresponds to the upper bound of Cramér's LD theorem.

The constant $I(a)$ in (2.5) gives, in a sense, the best possible exponential rate at which the probability $P(Z_N \geq a)$ converges to zero for $a > \mu$. This follows from the lower bound

$$\liminf_{N \to \infty} \frac{1}{N} \log\left[P(Z_N \geq a)\right] \ \geq \ -I(a) \tag{2.6}$$

of Cramér's LD theorem. A simple sufficient condition for (2.6) to hold is that the moment-generating function $M(t)$ is finite valued for all $t \in \mathbb{R}$. For a thorough discussion of the LD theory, the interested reader is referred to Dembo and Zeitouni [5].

The rate function $I(z)$ has the following properties: The function $I(z)$ is convex and attains its minimum at $z = \mu$, and $I(\mu) = 0$. Moreover, suppose that the moment-generating function $M(t)$ is finite valued for all t in a neighborhood of $t = 0$. Then X has finite moments, and it follows by the dominated convergence theorem that $M(t)$, and hence the function $\Lambda(t)$, are infinitely differentiable at $t = 0$, and $\Lambda'(0) = \mu$. Consequently for $a > \mu$ the derivative of $\psi(t) := ta - \Lambda(t)$ at $t = 0$ is greater than zero, and hence $\psi(t) > 0$ for $t > 0$ small enough. In that case it follows that $I(a) > 0$. Also, $I'(\mu) = 0$ and $I''(\mu) = \sigma^{-2}$, and hence by Taylor's expansion

$$I(a) = \frac{(a - \mu)^2}{2\sigma^2} + o(|a - \mu|^2). \tag{2.7}$$

Consequently, for a close to μ one can approximate $I(a)$ by $(a-\mu)^2/(2\sigma^2)$. Moreover, for any $\tilde{\epsilon} > 0$ there is a neighborhood $\mathcal{N}$ of μ such that

$$I(a) \geq \frac{(a - \mu)^2}{(2 + \tilde{\epsilon})\sigma^2} \quad \forall \ a \in \mathcal{N}. \tag{2.8}$$

In particular, one can take $\tilde{\epsilon} = 1$.

Now we return to problems (1.1) and (2.1). Consider numbers $\varepsilon \geq 0$, $\delta \in [0, \varepsilon]$, and the event $\{\hat{\mathcal{S}}_N^\delta \subset \mathcal{S}^\varepsilon\}$. It holds that

$$\left\{\hat{\mathcal{S}}_N^\delta \not\subset \mathcal{S}^\varepsilon\right\} = \bigcup_{x \in \mathcal{S} \setminus \mathcal{S}^\varepsilon} \bigcap_{y \in \mathcal{S}} \{\hat{g}_N(x) \leq \hat{g}_N(y) + \delta\}, \tag{2.9}$$

and hence

$$P\left(\hat{\mathcal{S}}_N^\delta \not\subset \mathcal{S}^\varepsilon\right) \leq \sum_{x \in \mathcal{S} \setminus \mathcal{S}^\varepsilon} P\left(\bigcap_{y \in \mathcal{S}} \{\hat{g}_N(x) \leq \hat{g}_N(y) + \delta\}\right). \tag{2.10}$$

Consider a mapping $u : \mathcal{S} \setminus \mathcal{S}^\varepsilon \mapsto \mathcal{S}$. It follows from (2.10) that

$$P\left(\hat{\mathcal{S}}_N^\delta \not\subset \mathcal{S}^\varepsilon\right) \leq \sum_{x \in \mathcal{S} \setminus \mathcal{S}^\varepsilon} P\Big(\hat{g}_N(x) - \hat{g}_N(u(x)) \leq \delta\Big). \tag{2.11}$$

We assume that the mapping $u(x)$ is chosen in such a way that for some $\varepsilon^* > \varepsilon$

$$g(u(x)) \leq g(x) - \varepsilon^* \quad \text{for all} \quad x \in \mathcal{S} \setminus \mathcal{S}^\varepsilon. \tag{2.12}$$

Note that if $u(\cdot)$ is a mapping from $\mathcal{S} \setminus \mathcal{S}^\varepsilon$ into the set $\mathcal{S}^*$, i.e., $u(x) \in \mathcal{S}^*$ for all $x \in \mathcal{S} \setminus \mathcal{S}^\varepsilon$, then (2.12) holds with

$$\varepsilon^* := \min_{x \in \mathcal{S} \setminus \mathcal{S}^\varepsilon} g(x) - v^*, \tag{2.13}$$

and that $\varepsilon^* > \varepsilon$ since the set $\mathcal{S}$ is finite. Therefore a mapping $u(\cdot)$ that satisfies condition (2.12) always exists.

For each $x \in \mathcal{S} \setminus \mathcal{S}^\varepsilon$, let

$$H(x, w) := G(u(x), w) - G(x, w).$$

Note that $\mathbb{E}[H(x, W)] = g(u(x)) - g(x)$, and hence $\mathbb{E}[H(x, W)] \leq -\varepsilon^*$. Let $W^1, \ldots, W^N$ be an i.i.d. random sample of N realizations of the random vector W, and consider the sample average function

$$\hat{h}_N(x) := \frac{1}{N} \sum_{j=1}^N H(x, W^j) = \hat{g}_N(u(x)) - \hat{g}_N(x).$$

It follows from (2.11) that

$$P\left(\hat{\mathcal{S}}_N^\delta \not\subset \mathcal{S}^\varepsilon\right) \leq \sum_{x \in \mathcal{S} \setminus \mathcal{S}^\varepsilon} P\left(\hat{h}_N(x) \geq -\delta\right). \tag{2.14}$$

Let $I_x(\cdot)$ denote the LD rate function of $H(x, W)$. Inequality (2.14) together with (2.5) implies that

$$P\left(\hat{\mathcal{S}}_N^\delta \not\subset \mathcal{S}^\varepsilon\right) \leq \sum_{x \in \mathcal{S} \setminus \mathcal{S}^\varepsilon} e^{-N I_x(-\delta)}. \tag{2.15}$$

It is important to note that the above inequality (2.15) is not asymptotic and is valid for any random sample of size N.

Assumption (A). For every $x \in \mathcal{S}$ the moment-generating function of the random variable $H(x, W)$ is finite valued in a neighborhood of 0.

The above assumption (A) holds, for example, if $H(x, W)$ is a bounded random variable, or if $H(x, \cdot)$ grows at most linearly and W has a distribution from the exponential family.

PROPOSITION 2.2. *Let ε and δ be nonnegative numbers such that $\delta \leq \varepsilon$. Then*

$$P\left(\hat{\mathcal{S}}_N^{\delta} \not\subset \mathcal{S}^{\varepsilon}\right) \;\leq\; |\mathcal{S} \setminus \mathcal{S}^{\varepsilon}| e^{-N\gamma(\delta,\varepsilon)}, \tag{2.16}$$

where

$$\gamma(\delta, \varepsilon) := \min_{x \in \mathcal{S} \setminus \mathcal{S}^{\varepsilon}} I_x(-\delta). \tag{2.17}$$

Moreover, if Assumption (A) *holds, then $\gamma(\delta, \varepsilon) > 0$.*

Proof. Inequality (2.16) is an immediate consequence of inequality (2.15). It holds that $-\delta > -\varepsilon^* \geq \mathbb{E}[H(x, W)]$, and hence it follows by Assumption (A) that $I_x(-\delta) > 0$ for every $x \in \mathcal{S} \setminus \mathcal{S}^{\varepsilon}$. This implies that $\gamma(\delta, \varepsilon) > 0$. □

The following asymptotic result is an immediate consequence of inequality (2.16),

$$\limsup_{N \to \infty} \frac{1}{N} \log\left[1 - P(\hat{\mathcal{S}}_N^{\delta} \subset \mathcal{S}^{\varepsilon})\right] \;\leq\; -\gamma(\delta, \varepsilon). \tag{2.18}$$

Inequality (2.18) means that the probability of the event $\{\hat{\mathcal{S}}_N^{\delta} \subset \mathcal{S}^{\varepsilon}\}$ approaches 1 *exponentially fast* as $N \to \infty$. This suggests that Monte Carlo sampling, combined with an efficient method for solving the deterministic SAA problem, can efficiently solve the type of problems under study, provided that the constant $\gamma(\delta, \varepsilon)$ is not "too small."

It follows from (2.7) that

$$I_x(-\delta) \;\approx\; \frac{\left(-\delta - \mathbb{E}[H(x, W)]\right)^2}{2\sigma_x^2} \;\geq\; \frac{(\varepsilon^* - \delta)^2}{2\sigma_x^2}, \tag{2.19}$$

where ε^* is defined in (2.13) and

$$\sigma_x^2 \;:=\; \mathrm{Var}[H(x, W)] \;=\; \mathrm{Var}[G(u(x), W) - G(x, W)].$$

Therefore the constant $\gamma(\delta, \varepsilon)$, given in (2.17), can be approximated by

$$\gamma(\delta, \varepsilon) \;\approx\; \min_{x \in \mathcal{S} \setminus \mathcal{S}^{\varepsilon}} \frac{(-\delta - \mathbb{E}[H(x, W)])^2}{2\sigma_x^2} \;\geq\; \frac{(\varepsilon^* - \delta)^2}{2\sigma_{\max}^2} \;>\; \frac{(\varepsilon - \delta)^2}{2\sigma_{\max}^2}, \tag{2.20}$$

where

$$\sigma_{\max}^2 \;:=\; \max_{x \in \mathcal{S} \setminus \mathcal{S}^{\varepsilon}} \mathrm{Var}[G(u(x), W) - G(x, W)]. \tag{2.21}$$

A result similar to the one of Proposition 2.2 was derived in [14] by using slightly different arguments. The LD rate functions of the random variables $G(x, W)$ were used there, which resulted in estimates of the exponential constant similar to the estimate (2.20) but with σ_x^2 replaced by the variance of $G(x, W)$. Due to a positive correlation between $G(x, W)$ and $G(u(x), W)$, the variance of $G(x, W) - G(u(x), W)$ tends to be smaller than the variance of $G(x, W)$, thereby providing a smaller upper

bound on $P(\hat{\mathcal{S}}_N^\delta \not\subset \mathcal{S}^\varepsilon)$, especially when $u(x)$ is chosen to minimize $\mathrm{Var}[G(x,W) - G(u(x),W)]/[g(x) - g(u(x))]^2$. This suggests that the estimate given in (2.20) could be more accurate than the one obtained in [14].

To illustrate some implications of the bound (2.16) for issues of the complexity of solving stochastic problems, let us fix a significance level $\alpha \in (0,1)$, and estimate the sample size N which is needed for the probability $P(\hat{\mathcal{S}}_N^\delta \subset \mathcal{S}^\varepsilon)$ to be at least $1-\alpha$. By requiring that the right-hand side of (2.16) be less than or equal to α, we obtain that

$$N \;\geq\; \frac{1}{\gamma(\delta,\varepsilon)} \log\left(\frac{|\mathcal{S} \setminus \mathcal{S}^\varepsilon|}{\alpha}\right). \tag{2.22}$$

Moreover, it follows from (2.8) and (2.17) that $\gamma(\delta,\varepsilon) \geq (\varepsilon-\delta)^2/(3\sigma_{\max}^2)$ for all $\varepsilon \geq 0$ sufficiently small. Therefore it holds that for all $\varepsilon > 0$ small enough and $\delta \in [0,\varepsilon)$, a sufficient condition for (2.22) is that

$$N \;\geq\; \frac{3\sigma_{\max}^2}{(\varepsilon-\delta)^2} \log\left(\frac{|\mathcal{S}|}{\alpha}\right). \tag{2.23}$$

It appears that the bound (2.23) may be too conservative for practical estimates of the required sample sizes (see the discussion in section 4.2). However, the estimate (2.23) has interesting consequences for complexity issues. A key characteristic of (2.23) is that N depends only *logarithmically* both on the size of the feasible set $\mathcal{S}$ and on the tolerance probability α. An important implication of such behavior is the following. Suppose that (i) the size of the feasible set $\mathcal{S}$ grows at most exponentially in the length of the problem input, (ii) the variance $\sigma_{\max}^2$ grows polynomially in the length of the problem input, and (iii) the complexity of finding a δ-optimal solution for (2.1) grows polynomially in the length of the problem input and the sample size N. Then a solution can be generated in time that grows polynomially in the length of the problem input such that, with probability at least $1-\alpha$, the solution is ε-optimal for (1.1). A careful analysis of these issues is beyond the scope of this paper, and requires further investigation.

Now suppose for a moment that the true problem has unique optimal solution x^*, i.e., $\mathcal{S}^* = \{x^*\}$ is a singleton, and consider the event that the SAA problem (2.1) has unique optimal solution $\hat{x}_N$ and $\hat{x}_N = x^*$. We denote that event by $\{\hat{x}_N = x^*\}$. Furthermore, consider the mapping $u : \mathcal{S} \setminus \mathcal{S}^\varepsilon \mapsto \{x^*\}$, i.e., $u(x) \equiv x^*$, and the corresponding constant $\gamma^* := \gamma(0,0)$. That is,

$$\gamma^* \;=\; \min_{x \in \mathcal{S} \setminus \{x^*\}} I_x(0), \tag{2.24}$$

with $I_x(\cdot)$ being the LD rate function of $G(x^*,W) - G(x,W)$. Note that $\mathbb{E}[G(x^*,W) - G(x,W)] = g(x^*) - g(x)$, and hence $\mathbb{E}[G(x^*,W) - G(x,W)] < 0$ for every $x \in \mathcal{S} \setminus \{x^*\}$. Therefore, if Assumption (A) holds, i.e., the moment-generating function of $G(x^*,W) - G(x,W)$ is finite valued in a neighborhood of 0, then $\gamma^* > 0$.

PROPOSITION 2.3. *Suppose that the true problem has unique optimal solution x^* and the moment-generating function of each random variable $G(x^*,W) - G(x,W)$, $x \in \mathcal{S} \setminus \{x^*\}$, is finite valued on $\mathbb{R}$. Then*

$$\lim_{N\to\infty} \frac{1}{N} \log\left[1 - P(\hat{x}_N = x^*)\right] \;=\; -\gamma^*. \tag{2.25}$$

Proof. It follows from (2.18) that

$$\limsup_{N\to\infty} \frac{1}{N} \log\left[1 - P(\hat{x}_N = x^*)\right] \;\leq\; -\gamma^*. \tag{2.26}$$

Consider the complement of the event $\{\hat{x}_N = x^*\}$, which is denoted $\{\hat{x}_N \neq x^*\}$. The event $\{\hat{x}_N \neq x^*\}$ is equal to the union of the events $\{\hat{g}_N(x) \leq \hat{g}_N(x^*)\}$, $x \in \mathcal{S} \setminus \{x^*\}$. Therefore, for any $x \in \mathcal{S} \setminus \{x^*\}$,

$$P\left(\hat{x}_N \neq x^*\right) \;\geq\; P\left(\hat{g}_N(x) \leq \hat{g}_N(x^*)\right).$$

By using the lower bound (2.6) of Cramér's LD theorem, it follows that the inequality

$$\liminf_{N\to\infty} \frac{1}{N} \log\left[1 - P(\hat{x}_N = x^*)\right] \;\geq\; -I_x(0) \tag{2.27}$$

holds for every $x \in \mathcal{S} \setminus \{x^*\}$. Inequalities (2.26) and (2.27) imply (2.25). □

Suppose that $\mathcal{S}^* = \{x^*\}$ and consider the number

$$\kappa \;:=\; \max_{x\in\mathcal{S}\setminus\{x^*\}} \frac{\mathrm{Var}[G(x,W) - G(x^*,W)]}{[g(x) - g(x^*)]^2}. \tag{2.28}$$

It follows from (2.7) and (2.24) that $\kappa \approx 1/(2\gamma^*)$. One can view κ as a *condition number* of the true problem. That is, the sample size required for the event $\{\hat{x}_N = x^*\}$ to happen with a given probability is roughly proportional to κ. The number defined in (2.28) can be viewed as a discrete version of the condition number introduced in [22] for piecewise linear continuous problems.

For a problem with a large feasible set $\mathcal{S}$, the number $\min_{x\in\mathcal{S}\setminus\{x^*\}} g(x) - g(x^*)$, although positive if $\mathcal{S}^* = \{x^*\}$, tends to be small. Therefore the sample size required to calculate the exact optimal solution x^* with a high probability could be very large, even if the optimal solution x^* is unique. For ill-conditioned problems it makes sense to search for approximate (ε-optimal) solutions of the true problem. In that respect the bound (2.16) is more informative since the corresponding constant $\gamma(\delta,\varepsilon)$ is guaranteed to be at least of the order $(\varepsilon-\delta)^2/(2\sigma^2_{\max})$.

It is also insightful to note the behavior of the condition number κ for a discrete optimization problem with linear objective function $G(x,W) := \sum_{i=1}^k W_i x_i$ and feasible set $\mathcal{S}$ given by the vertices of the unit hypercube in $\mathbb{R}^k$, i.e., $\mathcal{S} := \{0,1\}^k$. In that case the corresponding true optimization problem is

$$\min_{x\in\{0,1\}^k} \left\{ g(x) = \sum_{i=1}^k \bar{w}_i x_i \right\},$$

where $\bar{w}_i := E[W_i]$. Suppose that $\bar{w}_i > 0$ for all $i \in \{1,\ldots,k\}$, and hence the origin is the unique optimal solution of the true problem, i.e., $\mathcal{S}^* = \{0\}$. Let

$$\vartheta_i^2 \;:=\; \frac{\mathrm{Var}[W_i]}{(E[W_i])^2}$$

denote the squared coefficient of variation of W_i, and let

$$\rho_{ij} \;:=\; \frac{\mathrm{Cov}[W_i, W_j]}{\sqrt{\mathrm{Var}[W_i]}\sqrt{\mathrm{Var}[W_j]}}$$

denote the correlation coefficient between W_i and W_j. It follows that for any $x \in \{0,1\}^k \setminus \{0\}$,

$$\frac{\mathrm{Var}\left[\sum_{i=1}^k W_i x_i\right]}{\left[\sum_{i=1}^k \bar{w}_i x_i\right]^2} \;=\; \frac{\sum_{i=1}^k \sum_{j=1}^k \rho_{ij}\vartheta_i \bar{w}_i x_i \vartheta_j \bar{w}_j x_j}{\sum_{i=1}^k \sum_{j=1}^k \bar{w}_i x_i \bar{w}_j x_j} \;\le\; \max_{i\in\{1,\ldots,k\}} \vartheta_i^2.$$

Thus

$$\kappa \;=\; \max_{x\in\{0,1\}^k\setminus\{0\}} \frac{\mathrm{Var}\left[\sum_{i=1}^k W_i x_i\right]}{\left[\sum_{i=1}^k \bar{w}_i x_i\right]^2} \;=\; \max_{i\in\{1,\ldots,k\}} \vartheta_i^2.$$

The last equality follows because the maximum is attained by setting $x_i = 1$ for the index i for which W_i has the maximum squared coefficient of variation ϑ_i^2, and setting $x_j = 0$ for the remaining variables. Thus, in this example the condition number κ is equal to the maximum squared coefficient of variation of the W_i's.

2.3. Asymptotics of sample objective values. Next we discuss the asymptotics of the SAA optimal objective value $\hat{v}_N$. For any subset $\mathcal{S}'$ of $\mathcal{S}$ the inequality $\hat{v}_N \le \min_{x\in\mathcal{S}'} \hat{g}_N(x)$ holds. In particular, by taking $\mathcal{S}' = \mathcal{S}^*$, it follows that $\hat{v}_N \le \min_{x\in\mathcal{S}^*} \hat{g}_N(x)$, and hence

$$\mathbb{E}[\hat{v}_N] \;\le\; \mathbb{E}\left\{\min_{x\in\mathcal{S}^*} \hat{g}_N(x)\right\} \;\le\; \min_{x\in\mathcal{S}^*} \mathbb{E}[\hat{g}_N(x)] \;=\; v^*.$$

That is, the estimator $\hat{v}_N$ has a negative bias (cf. Norkin, Pflug, and Ruszczyński [19] and Mak, Morton, and Wood [15]).

It follows from Proposition 2.1 that w.p.1, for N sufficiently large, the set $\hat{\mathcal{S}}_N$ of optimal solutions of the SAA problem is included in $\mathcal{S}^*$. In that case it holds that

$$\hat{v}_N \;=\; \min_{x\in\hat{\mathcal{S}}_N} \hat{g}_N(x) \;\ge\; \min_{x\in\mathcal{S}^*} \hat{g}_N(x).$$

Since the opposite inequality always holds, it follows that, w.p.1, $\hat{v}_N - \min_{x\in\mathcal{S}^*} \hat{g}_N(x) = 0$ for N large enough. Multiplying both sides of this equation by $\sqrt{N}$ it follows that, w.p.1, $\sqrt{N}\,[\hat{v}_N - \min_{x\in\mathcal{S}^*} \hat{g}_N(x)] = 0$ for N large enough, and hence

$$\lim_{N\to\infty} \sqrt{N}\left[\hat{v}_N - \min_{x\in\mathcal{S}^*} \hat{g}_N(x)\right] \;=\; 0 \quad \text{w.p.1.} \tag{2.29}$$

Since convergence w.p.1 implies convergence in probability, it follows from (2.29) that $\sqrt{N}\,[\hat{v}_N - \min_{x\in\mathcal{S}^*} \hat{g}_N(x)]$ converges in probability to zero, i.e.,

$$\hat{v}_N \;=\; \min_{x\in\mathcal{S}^*} \hat{g}_N(x) + o_p(N^{-1/2}).$$

Furthermore, since $v^* = g(x)$ for any $x \in \mathcal{S}^*$, it follows that

$$\sqrt{N}\left[\min_{x\in\mathcal{S}^*} \hat{g}_N(x) - v^*\right] \;=\; \sqrt{N} \min_{x\in\mathcal{S}^*} [\hat{g}_N(x) - v^*] \;=\; \min_{x\in\mathcal{S}^*} \left\{\sqrt{N}\,[\hat{g}_N(x) - g(x)]\right\}.$$

Suppose that for every $x \in \mathcal{S}$ the variance

$$\sigma^2(x) \;:=\; \mathrm{Var}[G(x,W)] \tag{2.30}$$

exists. Then it follows by the central limit theorem (CLT) that, for any $x \in \mathcal{S}$, $\sqrt{N}[\hat{g}_N(x) - g(x)]$ converges in distribution to a normally distributed variable $Z(x)$ with zero mean and variance $\sigma^2(x)$. Moreover, again by the CLT, random variables $Z(x)$ have the same covariance function as $G(x, W)$, i.e., the covariance between $Z(x)$ and $Z(x')$ is equal to the covariance between $G(x, W)$ and $G(x', W)$ for any $x, x' \in \mathcal{S}$. Hence the following result is obtained (it is similar to an asymptotic result for stochastic programs with continuous decision variables which was derived in [21]). We use "$\Rightarrow$" to denote convergence in distribution.

PROPOSITION 2.4. *Suppose that variances $\sigma^2(x)$, defined in* (2.30)*, exist for every $x \in \mathcal{S}^*$. Then*

$$\sqrt{N}(\hat{v}_N - v^*) \;\Rightarrow\; \min_{x \in \mathcal{S}^*} Z(x), \tag{2.31}$$

where $Z(x)$ are normally distributed random variables with zero mean and the covariance function given by the corresponding covariance function of $G(x, W)$. In particular, if $\mathcal{S}^* = \{x^*\}$ is a singleton, then

$$\sqrt{N}(\hat{v}_N - v^*) \;\Rightarrow\; N(0, \sigma^2(x^*)). \tag{2.32}$$

Although for any given x the mean (expected value) of $Z(x)$ is zero, the expected value of the minimum of $Z(x)$ over a subset $\mathcal{S}'$ of $\mathcal{S}$ can be negative and tends to be smaller for a larger set $\mathcal{S}'$. Therefore, it follows from (2.31) that for ill-conditioned problems, where the set of optimal or nearly optimal solutions is large, the estimate $\hat{v}_N$ of v^* tends to be heavily biased. Note that convergence in distribution does not necessarily imply convergence of the corresponding means. Under mild additional conditions it follows from (2.31) that $\sqrt{N}[\mathbb{E}(\hat{v}_N) - v^*] \to \mathbb{E}[\min_{x \in \mathcal{S}^*} Z(x)]$.

3. Algorithm design. In the previous section we established a number of convergence results for the SAA method. The results describe how the optimal value $\hat{v}_N$ and the set $\hat{\mathcal{S}}_N^\varepsilon$ of ε-optimal solutions of the SAA problem converge to their true counterparts v^* and $\mathcal{S}^\varepsilon$, respectively, as the sample size N increases. These results provide some theoretical justification for the proposed method. When designing an algorithm for solving stochastic discrete optimization problems, many additional issues have to be addressed. Some of these issues are discussed in this section.

3.1. Selection of the sample size. In an algorithm, a finite sample size N or a sequence of finite sample sizes has to be chosen, and the algorithm has to stop after a finite amount of time. An important question is how these choices should be made. Estimate (2.23) gives a bound on the sample size required to find an ε-optimal solution with probability at least $1-\alpha$. This estimate has two shortcomings for computational purposes. First, for many problems it is not easy to compute the estimate, because $\sigma^2_{\max}$ and in some problems also $|\mathcal{S}|$ may be hard to compute. Second, as demonstrated in section 4.2, the bound may be far too conservative to obtain a practical estimate of the required sample size. To choose N, several trade-offs should be taken into account. With larger N, the objective function of the SAA problem tends to be a more accurate estimate of the true objective function, an optimal solution of the SAA problem tends to be a better solution, and the corresponding bounds on the optimality gap, discussed later, tend to be tighter. However, depending on the SAA problem (2.1) and the method used for solving the SAA problem, the computational complexity for solving the SAA problem increases at least linearly, and often exponentially, in the sample size N. Thus, in the choice of sample size N, the trade-off between the quality

of an optimal solution of the SAA problem and the bounds on the optimality gap, on the one hand, and computational effort, on the other hand, should be taken into account. Also, the choice of sample size N may be adjusted dynamically, depending on the results of preliminary computations. This issue is addressed in more detail later.

Typically, estimating the objective value $g(x)$ of a feasible solution $x \in \mathcal{S}$ by the sample average $\hat{g}_N(x)$ requires much less computational effort than solving the SAA problem (for the same sample size N). Thus, although computational complexity considerations motivate one to choose a relatively small sample size N for the SAA problem, it makes sense to choose a larger sample size N' to obtain an accurate estimate $\hat{g}_{N'}(\hat{x}_N)$ of the objective value $g(\hat{x}_N)$ of an optimal solution $\hat{x}_N$ of the SAA problem. A measure of the accuracy of a sample average estimate $\hat{g}_{N'}(\hat{x}_N)$ of $g(\hat{x}_N)$ is given by the corresponding sample variance $S^2_{N'}(\hat{x}_N)/N'$, which can be calculated from the same sample of size N'. Again the choice of N' involves a trade-off between computational effort and accuracy, measured by $S^2_{N'}(\hat{x}_N)/N'$.

3.2. Replication. If the computational complexity of solving the SAA problem increases faster than linearly in the sample size N, it may be more efficient to choose a smaller sample size N and to generate and solve several SAA problems with i.i.d. samples, that is, to replicate generating and solving SAA problems.

With such an approach, several issues have to be addressed. One question is whether there is a guarantee that an optimal (or ε-optimal) solution for the true problem will be produced if a sufficient number of SAA problems, based on independent samples of size N, are solved. One can view such a procedure as Bernoulli trials with probability of success $p = p(N)$. Here "success" means that a calculated optimal solution $\hat{x}_N$ of the SAA problem is an optimal solution of the true problem. It follows from Proposition 2.1 that this probability p tends to 1 as $N \to \infty$, and, moreover, by Proposition 2.2 it tends to 1 exponentially fast if Assumption (A) holds. However, for a finite N the probability p can be small or even zero. The probability of producing an optimal solution of the true problem at least once in M trials is $1 - (1-p)^M$, and this probability tends to one as $M \to \infty$, provided p is positive. Thus a relevant question is whether there is a guarantee that p is positive for a given sample size N. The following example shows that the sample size N required for p to be positive is problem-specific, cannot be bounded by a function that depends only on the number of feasible solutions, and can be arbitrarily large.

Example. Suppose that $\mathcal{S} := \{-1, 0, 1\}$, that W can take two values w_1 and w_2 with respective probabilities $1-\gamma$ and γ, and that $G(-1, w_1) := -1$, $G(0, w_1) := 0$, $G(1, w_1) := 2$, and $G(-1, w_2) := 2k$, $G(0, w_2) := 0$, $G(1, w_2) := -k$, where k is an arbitrary positive number. Let $\gamma = 1/(k+1)$. Then $g(x) = (1-\gamma)G(x, w_1) + \gamma G(x, w_2)$, and thus $g(-1) = k/(k+1)$, $g(0) = 0$, and $g(1) = k/(k+1)$. Therefore $x^* = 0$ is the unique optimal solution of the true problem. If the sample does not contain any observations w_2, then $\hat{x}_N = -1 \neq x^*$. Suppose the sample contains at least one observation w_2. Then $\hat{g}_N(1) \leq [2(N-1) - k]/N$. Thus $\hat{g}_N(1) < 0 = \hat{g}_N(0)$ if $N \leq k/2$, and $\hat{x}_N = 1 \neq x^*$. Thus a sample of size $N > k/2$ at least is required, in order for $x^* = 0$ to be an optimal solution of the SAA problem. (Note that $\mathrm{Var}[G(-1, W) - G(0, W)]$ and $\mathrm{Var}[G(1, W) - G(0, W)]$ are $\Theta(k)$, which causes the problem to become harder as k increases.)

Another issue that has to be addressed is the choice of the number M of replications. In a manner similar to the choice of sample size N, the number M of replications may be chosen dynamically. One approach to doing this is discussed next. For sim-

plicity of presentation, suppose that each SAA replication produces one candidate solution, which can be an optimal (ε-optimal) solution of the SAA problem. Let $\hat{x}_N^m$ denote the candidate solution produced by the mth SAA replication (trial). The optimality gap $g(\hat{x}_N^m) - v^*$ can be estimated, as described in the next section. If a stopping criterion based on the optimality gap estimate is satisfied, then no more replications are performed. Otherwise, additional SAA replications with the same sample size N are performed, or the sample size N is increased. The following argument provides a simple guideline as to whether an additional SAA replication with the same sample size N is likely to provide a better solution than the best solution found so far.

Note that, by construction, the random variables $g(\hat{x}_N^m)$, $m = 1, \ldots$, are i.i.d., and their common probability distribution has a finite support because the set $\mathcal{S}$ is finite. Suppose that M replications with sample size N have been performed so far. If the probability distribution of $g(\hat{x}_N)$ were continuous, then the probability that the $(M+1)$th SAA replication with the same sample size would produce a better solution than the best of the solutions produced by the M replications so far would be equal to $1/(M+1)$. Because in fact the distribution of $g(\hat{x}_N)$ is discrete, this probability is less than or equal to $1/(M+1)$. Thus, when $1/(M+1)$ becomes sufficiently small, additional SAA replications with the same sample size are not likely to be worth the effort, and either the sample size N should be increased or the procedure should be stopped.

3.3. Performance bounds. To assist in making stopping decisions, as well as for other performance evaluation purposes, one would like to compute the optimality gap $g(\hat{x}) - v^*$ for a given solution $\hat{x} \in \mathcal{S}$. Unfortunately, the very reason for the approach described in this paper implies that both terms of the optimality gap are hard to compute. As before,

$$\hat{g}_{N'}(\hat{x}) \;:=\; \frac{1}{N'} \sum_{j=1}^{N'} G(\hat{x}, W^j)$$

is an unbiased estimator of $g(\hat{x})$, and the variance of $\hat{g}_{N'}(\hat{x})$ is estimated by $S_{N'}^2(\hat{x})/N'$, where $S_{N'}^2(\hat{x})$ is the sample variance of $G(\hat{x}, W^j)$, based on the sample of size N'.

An estimator of v^* is given by

$$\bar{v}_N^M \;:=\; \frac{1}{M} \sum_{m=1}^{M} \hat{v}_N^m,$$

where $\hat{v}_N^m$ denotes the optimal objective value of the mth SAA replication. Note that $\mathbb{E}[\bar{v}_N^M] = \mathbb{E}[\hat{v}_N]$, and hence the estimator $\bar{v}_N^M$ has the same negative bias as $\hat{v}_N$. Proposition 2.4 indicates that this bias tends to be bigger for ill-conditioned problems with larger sets of optimal, or nearly optimal, solutions. Consider the corresponding estimator $\hat{g}_{N'}(\hat{x}) - \bar{v}_N^M$ of the optimality gap $g(\hat{x}) - v^*$, at the point $\hat{x}$. Since

$$\mathbb{E}\left[\hat{g}_{N'}(\hat{x}) - \bar{v}_N^M\right] \;=\; g(\hat{x}) - \mathbb{E}[\hat{v}_N] \;\geq\; g(\hat{x}) - v^*, \tag{3.1}$$

it follows that on average the above estimator overestimates the optimality gap $g(\hat{x}) - v^*$. It is possible to show (Norkin, Pflug, and Ruszczyński [19], and Mak, Morton, and Wood [15]) that the bias $v^* - \mathbb{E}[\hat{v}_N]$ is monotonically decreasing in the sample size N.

The variance of $\bar{v}_N^M$ is estimated by

$$\frac{S_M^2}{M} = \frac{1}{M(M-1)} \sum_{m=1}^{M} \left(\hat{v}_N^m - \bar{v}_N^M\right)^2. \tag{3.2}$$

If the M samples, of size N, and the evaluation sample, of size N', are independent, then the variance of the optimality gap estimator $\hat{g}_{N'}(\hat{x}) - \bar{v}_N^M$ can be estimated by $S_{N'}^2(\hat{x})/N' + S_M^2/M$.

An estimator of the optimality gap $g(\hat{x}) - v^*$ with possibly smaller variance is $\bar{g}_N^M(\hat{x}) - \bar{v}_N^M$, where

$$\bar{g}_N^M(\hat{x}) := \frac{1}{M} \sum_{m=1}^{M} \hat{g}_N^m(\hat{x})$$

and $\hat{g}_N^m(\hat{x})$ is the sample average objective value at $\hat{x}$ of the mth SAA sample of size N,

$$\hat{g}_N^m(\hat{x}) := \frac{1}{N} \sum_{j=1}^{N} G(\hat{x}, W^{mj}).$$

The variance of $\bar{g}_N^M(\hat{x}) - \bar{v}_N^M$ is estimated by

$$\frac{\bar{S}_M^2}{M} = \frac{1}{M(M-1)} \sum_{m=1}^{M} \left[\left(\hat{g}_N^m(\hat{x}) - \hat{v}_N^m\right) - \left(\bar{g}_N^M(\hat{x}) - \bar{v}_N^M\right)\right]^2.$$

Which estimator of the optimality gap has the least variance depends on the correlation between $\hat{g}_N^m(\hat{x})$ and $\hat{v}_N^m$, as well as on the sample sizes N, N', and M. For many applications, one would expect positive correlation between $\hat{g}_N^m(\hat{x})$ and $\hat{v}_N^m$. The additional computational effort to compute $\hat{g}_N^m(\hat{x})$ for $m = 1, \ldots, M$ should also be taken into account when evaluating any such variance reduction. Either way, the CLT can be applied to the optimality gap estimators $\hat{g}_{N'}(\hat{x}) - \bar{v}_N^M$ and $\bar{g}_N^M(\hat{x}) - \bar{v}_N^M$, so that the accuracy of an optimality gap estimator can be taken into account by adding a multiple z_α of its estimated standard deviation to the gap estimator. Here $z_\alpha := \Phi^{-1}(1-\alpha)$, where $\Phi(z)$ is the cumulative distribution function of the standard normal distribution. For example, if $\hat{x} \in \mathcal{S}$ denotes the candidate solution with the best value of $\hat{g}_{N'}(\hat{x})$ found after M replications, then an optimality gap estimator taking accuracy into account is given by either

$$\hat{g}_{N'}(\hat{x}) - \bar{v}_N^M + z_\alpha \left(\frac{S_{N'}^2(\hat{x})}{N'} + \frac{S_M^2}{M}\right)^{1/2}$$

or

$$\bar{g}_N^M(\hat{x}) - \bar{v}_N^M + z_\alpha \frac{\bar{S}_M}{\sqrt{M}}.$$

For algorithm control, it is useful to separate an optimality gap estimator into its components. For example,

$$\begin{aligned} &\hat{g}_{N'}(\hat{x}) - \bar{v}_N^M + z_\alpha \left(\frac{S_{N'}^2(\hat{x})}{N'} + \frac{S_M^2}{M}\right)^{1/2} \\ &\quad = \left(\hat{g}_{N'}(\hat{x}) - g(\hat{x})\right) + \left(g(\hat{x}) - v^*\right) + \left(v^* - \bar{v}_N^M\right) + z_\alpha \left(\frac{S_{N'}^2(\hat{x})}{N'} + \frac{S_M^2}{M}\right)^{1/2}. \end{aligned} \tag{3.3}$$

In the four terms on the right-hand side of the above equation, the first term has expected value zero; the second term is the true optimality gap; the third term is the bias term, which has positive expected value decreasing in the sample size N; and the fourth term is the accuracy term, which is decreasing in the number M of replications and the sample size N'. Thus a disadvantage of these optimality gap estimators is that the gap estimator may be large if M, N, or N' is small, even if $\hat{x}$ is an optimal solution, i.e., $g(\hat{x}) - v^* = 0$.

3.4. Postprocessing, screening, and selection. Suppose a decision has been made to stop, for example when the optimality gap estimator has become small enough. At this stage the candidate solution $\hat{x} \in \mathcal{S}$ with the best value of $\hat{g}_{N'}(\hat{x})$ can be selected as the chosen solution. However, it may be worthwhile to perform a more detailed evaluation of the candidate solutions produced during the replications. There are several statistical screening and selection methods for selecting subsets of solutions or a single solution, among a (reasonably small) finite set of solutions, using samples of the objective values of the solutions. Many of these methods are described in Hochberg and Tamhane [12] and Bechhofer, Santner, and Goldsman [2]. In the numerical tests described in section 4, a combined procedure was used, as described in Nelson et al. [17]. During the first stage of the combined procedure, a subset $\mathcal{S}''$ of the candidate solutions $\mathcal{S}' := \{\hat{x}_N^1, \ldots, \hat{x}_N^M\}$ is chosen (called screening) for further evaluation, based on its sample average values $\hat{g}_{N'}(\hat{x}_N^m)$. During the second stage, sample sizes $N'' \geq N'$ are determined for more detailed evaluation, based on the sample variances $S^2_{N'}(\hat{x}_N^m)$. Then $N'' - N'$ additional observations are generated, and the candidate solution $\hat{x} \in \mathcal{S}''$ with the best value of $\hat{g}_{N''}(\hat{x})$ is selected as the chosen solution. The combined procedure guarantees that the chosen solution $\hat{x}$ has objective value $g(\hat{x})$ within a specified tolerance δ of the best value $\min_{\hat{x}_N^m \in \mathcal{S}'} g(\hat{x}_N^m)$ over all candidate solutions $\hat{x}_N^m$ with probability at least equal to specified confidence level $1 - \alpha$.

3.5. Algorithm. Next we state a proposed algorithm for the type of stochastic discrete optimization problem studied in this paper.

SAA Algorithm for Stochastic Discrete Optimization.

1. Choose initial sample sizes N and N', a decision rule for determining the number M of SAA replications (possibly involving a maximum number M' of SAA replications with the same sample size, such that $1/(M'+1)$ is sufficiently small), a decision rule for increasing the sample sizes N and N' if needed, and tolerance ε.
2. For $m = 1, \ldots, M$, do steps 2.1 through 2.3.
   - 2.1 Generate a sample of size N and solve the SAA problem (2.1) with objective value $\hat{v}_N^m$ and ε-optimal solution $\hat{x}_N^m$.
   - 2.2 Estimate the optimality gap $g(\hat{x}_N^m) - v^*$ and the variance of the gap estimator.
   - 2.3 If the optimality gap and the variance of the gap estimator are sufficiently small, go to step 4.
3. If the optimality gap or the variance of the gap estimator is too large, increase the sample sizes N and/or N', and return to step 2.
4. Choose the best solution $\hat{x}$ among all candidate solutions $\hat{x}_N^m$ produced, using a screening and selection procedure. Stop.

4. Numerical tests. In this section we describe an application of the SAA method to an optimization problem. The purposes of these tests are to investigate

the viability of the SAA approach, as well as to study the effects of problem parameters, such as the number of decision variables and the condition number κ, on the performance of the method.

4.1. Resource allocation problem. We apply the method to the following resource allocation problem. A decision maker has to choose a subset of k known alternative projects to take on. For this purpose a known quantity q of relatively low-cost resource is available to be allocated. Any additional amount of resource required can be obtained at a known incremental cost of c per unit of resource. The amount W_i of resource required by each project i is not known at the time the decision has to be made, but we assume that the decision maker has an estimate of the probability distribution of $W = (W_1, \ldots, W_k)$. Each project i has an expected net reward (expected revenue minus expected resource use times the lower cost) of r_i. Thus the optimization problem can be formulated as follows:

$$\max_{x \in \{0,1\}^k} \left\{ \sum_{i=1}^{k} r_i x_i - c\,\mathbb{E}\left[\sum_{i=1}^{k} W_i x_i - q \right]^+ \right\}, \tag{4.1}$$

where $[x]^+ := \max\{x, 0\}$. This problem can also be described as a knapsack problem, where a subset of k items has to be chosen, given a knapsack of size q into which to fit the items. The size W_i of each item i is random, and a per unit penalty of c has to be paid for exceeding the capacity of the knapsack. For this reason the problem is called the *static stochastic knapsack problem* (SSKP).

This problem was chosen for several reasons. First, expected value terms similar to that in the objective function of (4.1) occur in many interesting stochastic optimization problems. One such example is airline crew scheduling. An airline crew schedule is made up of crew pairings, where each crew pairing consists of a number of consecutive days (duties) of flying by a crew. Let $\{p_1, \ldots, p_k\}$ denote the set of pairings that can be chosen from. Then a crew schedule can be denoted by the decision vector $x \in \{0,1\}^k$, where $x_i = 1$ means that pairing p_i is flown. The cost $C_i(x)$ of a crew pairing p_i is given by

$$C_i(x) \;=\; \max \left\{ \sum_{d \in p_i} b_d(x), f t_i(x), g n_i \right\},$$

where $b_d(x)$ denotes the cost of duty d in pairing p_i, $t_i(x)$ denotes the total time duration of pairing p_i, n_i denotes the number of duties in pairing p_i, and f and g are constants determined by contracts. Even ignoring airline recovery actions such as cancellations and rerouting, $b_d(x)$ and $t_i(x)$ are random variables. The optimization problem is then

$$\min_{x \in \mathcal{X} \subset \{0,1\}^k} \; \sum_{i=1}^{k} \mathbb{E}[C_i(x)] x_i,$$

where $\mathcal{X}$ denotes the set of feasible crew schedules. Thus the objective function of the crew pairing problem can be written in a form similar to that of the objective function of (4.1).

Another example is a stochastic shortest path problem, where travel times are random and a penalty is incurred for arriving late at the destination. In this case,

the cost $C(x)$ of a path x is given by

$$C(x) \;=\; \sum_{(i,j)\in x} b_{ij} + c\left[\sum_{(i,j)\in x} t_{ij} - q\right]^+,$$

where b_{ij} is the cost of traversing arc (i,j), t_{ij} is the time of traversing arc (i,j), q is the available time to travel to the destination, and c is the penalty per unit time late. The optimization problem is then

$$\min_{x\in\mathcal{X}} \mathbb{E}[C(x)],$$

where $\mathcal{X}$ denotes the set of feasible paths in the network from the specified origin to the specified destination.

A second reason for choosing the SSKP is that objective functions with terms such as $\mathbb{E}[\sum_{i=1}^k W_i x_i - q]^+$ are interesting for the following reason. For many stochastic optimization problems good solutions can be obtained by replacing the random variables W by their means and then solving the resulting deterministic optimization problem $\max_x G(x, E[W])$, called the expected value problem (Birge and Louveaux [3]). It is easy to see that this may not be the case if the objective contains an expected value term as in (4.1). For a given solution x, this term may be very large but may become small if $W_1, \ldots, W_k$ are replaced by their means. In such a case, the obtained expected value problem may produce very bad solutions for the corresponding stochastic optimization problem.

The SSKP was also chosen because it is of interest by itself. One application is the decision faced by a contractor who can take on several contracts, such as an electricity supplier who can supply power to several groups of customers or a building contractor who can bid on several construction projects. The amount of work that will be required by each contract is unknown at the time the contracting decision has to be made. The contractor has the capacity to do work at a certain rate at relatively low cost, for example to generate electricity at a low-cost nuclear power plant. However, if the amount of work required exceeds the capacity, additional capacity has to be obtained at high cost, for example additional electricity can be generated at high-cost oil or natural gas–fired power plants. Norkin, Ermoliev, and Ruszczyński [18] also give several interesting applications of stochastic discrete optimization problems.

Note that the SAA problem of the SSKP can be formulated as the following integer linear program:

$$\begin{array}{lll} \max_{x,z} & \sum_{i=1}^k r_i x_i - \frac{c}{N}\sum_{j=1}^N z_j & \\ \text{subject to} & z_j \;\geq\; \sum_{i=1}^k W_i^j x_i - q, & j = 1,\ldots,N, \\ & x_i \;\in\; \{0,1\}, & i = 1,\ldots,k, \\ & z_j \;\geq\; 0, & j = 1,\ldots,N. \end{array} \tag{4.2}$$

This problem can be solved with the branch and bound method, using the linear programming relaxation to provide upper bounds.

4.2. Numerical results. We present results for two sets of instances of the SSKP. The first set of instances has 10 decision variables, and the second set has 20 decision variables each. For each set we present one instance (called instances 10D and 20D, respectively) that was designed to be hard (large condition number κ), and one randomly generated instance (called instances 10R and 20R, respectively).

TABLE 4.1
Condition numbers κ, optimal values v^*, and values $g(\bar{x})$ of optimal solutions $\bar{x}$ of expected value problems $\max_x G(x, E[W])$, for instances presented.

Instance	Condition number κ	Optimal value v^*	Expected value $g(\bar{x})$
10D	107000	42.7	26.2
10R	410	46.3	28.2
20D	954000	96.5	75.9
20R	233	130.3	109.0

Table 4.1 shows the condition numbers, the optimal values v^*, and the values $g(\bar{x})$ of the optimal solutions $\bar{x}$ of the associated expected value problems $\max_x G(x, E[W])$ for the four instances.

For all instances of the SSKP, the size variables W_i are independent normally distributed, for ease of evaluation of the results produced by the SAA method, as described in the next paragraph. For the randomly generated instances, the rewards r_i were generated from the uniform $(10, 20)$ distribution, the mean sizes μ_i were generated from the uniform $(20, 30)$ distribution, and the size standard deviations σ_i were generated from the uniform $(5, 15)$ distribution. For all instances, the per unit penalty $c = 4$.

If $W_i \sim N(\mu_i, \sigma_i^2)$, $i = 1, \ldots, k$, are independent normally distributed random variables, then the objective function of (4.1) can be written in closed form. That is, the random variable $Z(x) := \sum_{i=1}^k W_i x_i - q$ is normally distributed with mean $\mu(x) = \sum_{i=1}^k \mu_i x_i - q$ and variance $\sigma(x)^2 = \sum_{i=1}^k \sigma_i^2 x_i^2$. It is also easy to show, since $Z(x) \sim N(\mu(x), \sigma(x)^2)$, that

$$\mathbb{E}[Z(x)]^+ \;=\; \mu(x)\Phi\left(\frac{\mu(x)}{\sigma(x)}\right) + \frac{\sigma(x)}{\sqrt{2\pi}} \exp\left(\frac{-\mu(x)^2}{2\sigma(x)^2}\right),$$

where Φ denotes the standard normal cumulative distribution function. Thus, it follows that

$$g(x) \;=\; \sum_{i=1}^k r_i x_i - c\left[\mu(x)\Phi\left(\frac{\mu(x)}{\sigma(x)}\right) + \frac{\sigma(x)}{\sqrt{2\pi}} \exp\left(-\frac{\mu(x)^2}{2\sigma(x)^2}\right)\right]. \tag{4.3}$$

The benefit of such a closed form expression is that the objective value $g(x)$ can be computed quickly and accurately, which is useful for solving small instances of the problem by enumeration or branch and bound (cf. Cohn and Barnhart [4]) and for evaluation of solutions produced by the SAA Algorithm. Good numerical approximations are available for computing $\Phi(x)$, such as *Applied Statistics* Algorithm AS66 (Hill [11]). The SAA Algorithm was executed without the benefit of a closed form expression for $g(x)$, as would be the case for most probability distributions; (4.3) was used only to evaluate the solutions produced by the SAA Algorithm.

The first numerical experiment was conducted to observe how the exponential convergence rate established in Proposition 2.2 applies in the case of the SSKP, and to investigate how the convergence rate is affected by the number of decision variables and the condition number κ. Figures 4.1 and 4.2 show the estimated probability that an SAA optimal solution $\hat{x}_N$ has objective value $g(\hat{x}_N)$ within relative tolerance d of the optimal value v^*, i.e., $\hat{P}[v^* - g(\hat{x}_N) \leq d\, v^*]$, as a function of the sample size N, for different values of d. The experiment was conducted by generating $M = 1000$ independent SAA replications for each sample size N, computing SAA optimal

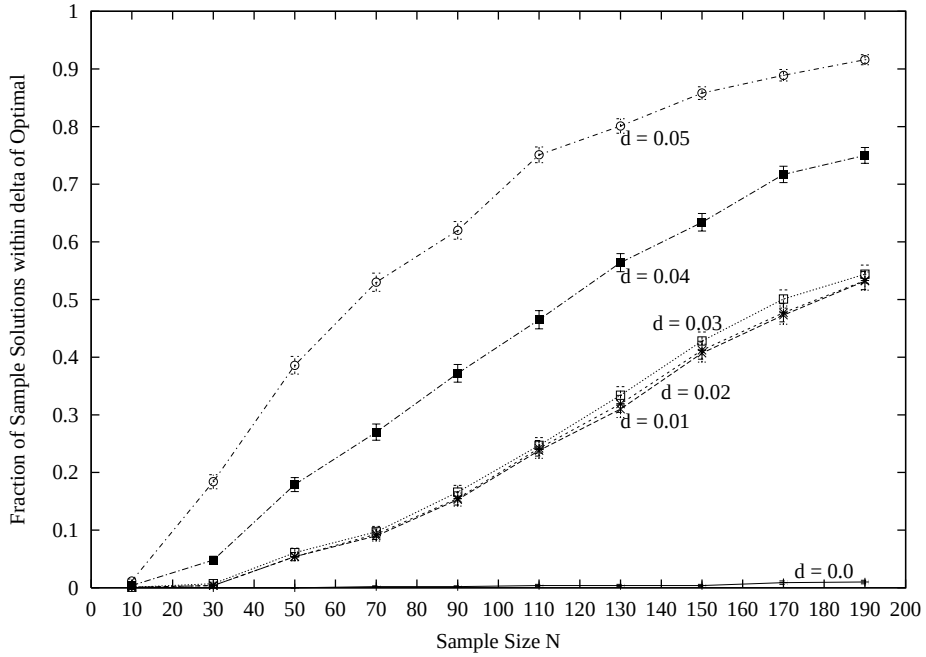


FIG. 4.1. *Probability of* SAA *optimal solution* $\hat{x}_N$ *having objective value* $g(\hat{x}_N)$ *within relative tolerance* d *of the optimal value* v^*, $\hat{P}[v^* - g(\hat{x}_N) \leq d\, v^*]$, *as a function of sample size* N *for different values of* d, *for instance* 20D.

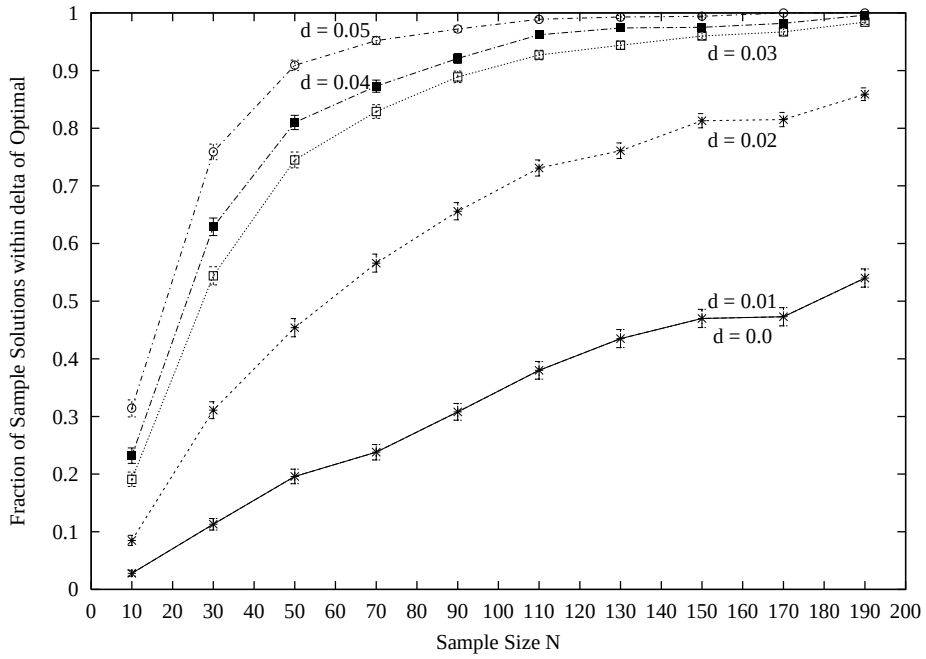


FIG. 4.2. *Probability of* SAA *optimal solution* $\hat{x}_N$ *having objective value* $g(\hat{x}_N)$ *within relative tolerance* d *of the optimal value* v^*, $\hat{P}[v^* - g(\hat{x}_N) \leq d\, v^*]$, *as a function of sample size* N *for different values of* d, *for instance* 20R.

solutions $\hat{x}_N^m$, $m = 1, \ldots, M$, and their objective values $g(\hat{x}_N^m)$ using (4.3), and then counting the number M_d of times that $v^* - g(\hat{x}_N^m) \leq d\, v^*$. Then the probability was estimated by $\hat{P}[v^* - g(\hat{x}_N) \leq d\, v^*] = M_d/M$, and the variance of this estimator was estimated by

$$\widehat{\mathrm{Var}}[\hat{P}] \;=\; \frac{M_d(1 - M_d/M)}{M(M-1)}.$$

The figures also show error bars of length $2(\widehat{\mathrm{Var}}[\hat{P}])^{1/2}$ on each side of the point estimate M_d/M.

One noticeable effect is that the probability that an SAA replication generates an optimal solution ($d = 0$) increases much more slowly with increase in the sample size N for the harder instances (10D and 20D) with poor condition numbers κ than for the randomly generated instances with better condition numbers. However, the probability that an SAA replication generates a reasonably good solution (e.g., $d = 0.05$) increases quite quickly with increase in the sample size N for both the harder instances and for the randomly generated instances.

The second numerical experiment demonstrates how the objective values $g(\hat{x}_N^m)$ of SAA optimal solutions $\hat{x}_N^m$ change as the sample size N increases, and how this change is affected by the number of decision variables and the condition number κ. In this experiment, the maximum number of SAA replications with the same sample size N was chosen as $M' = 50$. Additionally, after $M'' = 20$ replications with the same sample size N, the variance $S^2_{M''}$ of $\hat{v}_N^m$ was computed as in (3.2), because it is an important term in the optimality gap estimator (3.3). If $S^2_{M''}$ was too large, it indicated that the optimality gap estimate would be too large and that the sample size N should be increased. Otherwise, if $S^2_{M''}$ was not too large, then SAA replications were performed with the same sample size N until M' SAA replications had occurred. If the optimality gap estimate was greater than a specified tolerance, then the sample size N was increased and the procedure was repeated. Otherwise, if the optimality gap estimate was less than a specified tolerance, then a screening and selection procedure was applied to all the candidate solutions $\hat{x}_N^m$ generated, and the best solution among these was chosen.

Figures 4.3 and 4.4 show the objective values $g(\hat{x}_N^m)$ of SAA optimal solutions $\hat{x}_N^m$ produced during the course of the algorithm. There were several noticeable effects. First, good and often optimal solutions were produced early in the execution of the algorithm, but the sample size N had to be increased several times thereafter before the optimality gap estimate became sufficiently small for stopping, without any improvement in the quality of the generated solutions. Second, for the randomly generated instances a larger proportion of the SAA optimal solutions $\hat{x}_N^m$ were optimal or had objective values close to optimal, and optimal solutions were produced with smaller sample sizes N than were required for the harder instances. For example, for the harder instance with 10 decision variables (instance 10D), the optimal solution was first produced after $m = 6$ replications with sample size $N = 120$; and for instance 10R, the optimal solution was first produced after $m = 2$ replications with sample size $N = 20$. Also, for the harder instance with 20 decision variables (instance 20D), the optimal solution was not produced in any of the 270 total number of replications (but the second-best solution was produced 3 times); and for instance 20R, the optimal solution was first produced after $m = 15$ replications with sample size $N = 50$. Third,

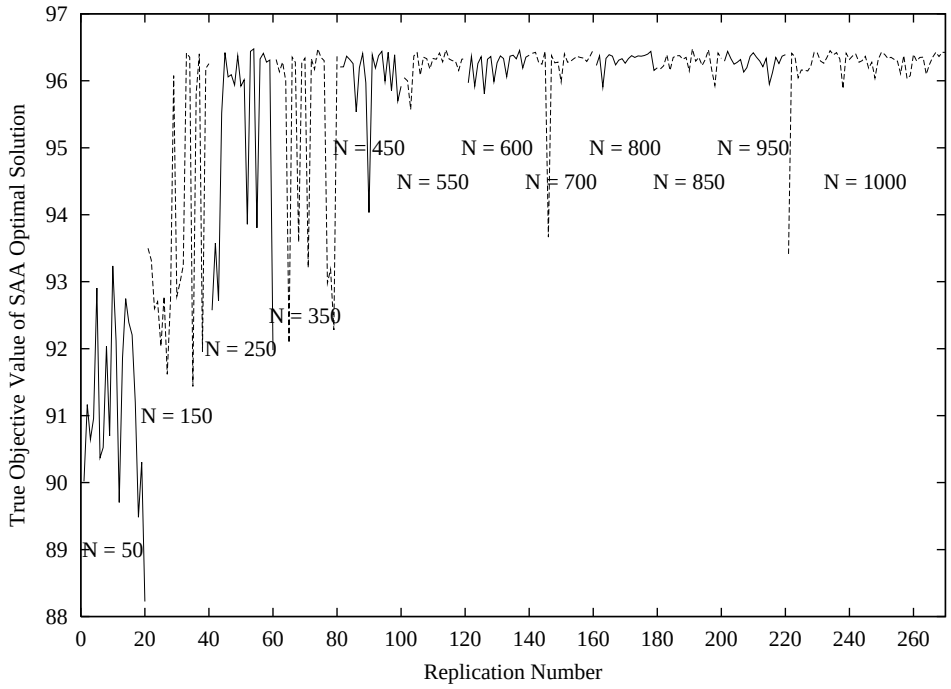


FIG. 4.3. *Improvement of objective values* $g(\hat{x}_N^m)$ *of* SAA *optimal solutions* $\hat{x}_N^m$ *as the sample size* N *increases, for instance* 20D.

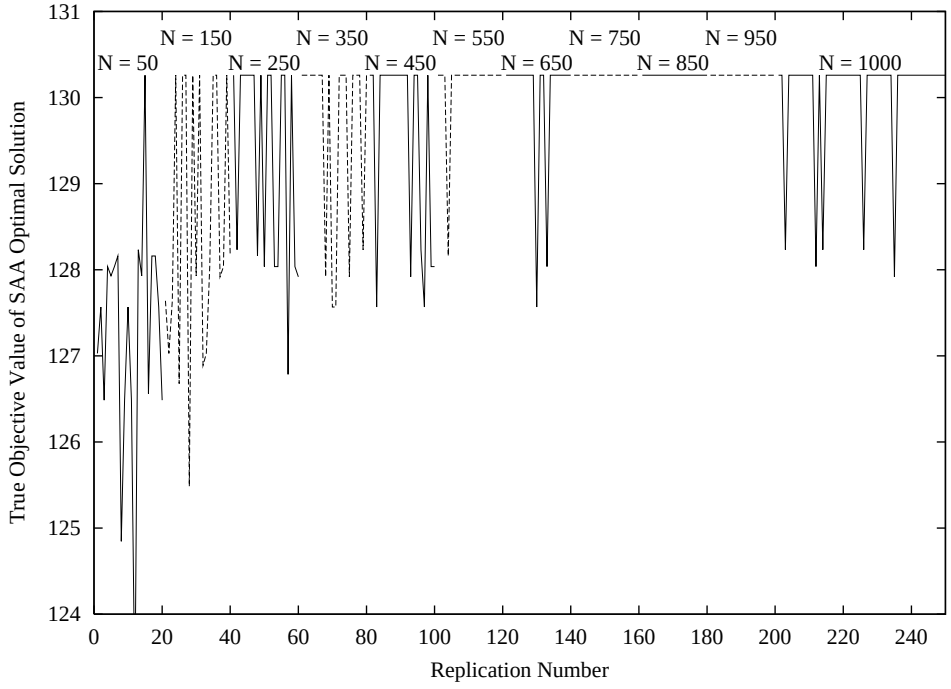


FIG. 4.4. *Improvement of objective values* $g(\hat{x}_N^m)$ *of* SAA *optimal solutions* $\hat{x}_N^m$ *as the sample size* N *increases, for instance* 20R.

for each of the instances, the expected value problem $\max_x G(x, E[W])$ was solved, with its optimal solution denoted by $\bar{x}$. The objective value $g(\bar{x})$ of each $\bar{x}$ is shown in Table 4.1. It is interesting to note that even with small sample sizes N, every solution $\hat{x}_N^m$ produced had a better objective value $g(\hat{x}_N^m)$ than $g(\bar{x})$.

As mentioned above, in the second numerical experiment it was noticed that often the optimality gap estimate is large, even if an optimal solution has been found, i.e., $v^* - g(\hat{x}) = 0$. (This is also a common problem in deterministic discrete optimization.) Consider the components of the optimality gap estimator $\hat{g}_{N'}(\hat{x}) - \bar{v}_N^M$ given in (3.3). The first component $g(\hat{x}) - \hat{g}_{N'}(\hat{x})$ can be made small with relatively little computational effort by choosing N' sufficiently large. The second component, the true optimality gap $v^* - g(\hat{x})$, is often small after only a few replications m with a small sample size N. The fourth component $z_\alpha(S_{N'}^2(\hat{x})/N' + S_M^2/M)^{1/2}$ can also be made small with relatively little computational effort by choosing N' and M sufficiently large. The major part of the problem seems to be caused by the third term $\bar{v}_N^M - v^*$ and by the fact that $\mathbb{E}[\bar{v}_N^M] - v^* \geq 0$, as identified in (3.1). It was also mentioned that the bias $\mathbb{E}[\bar{v}_N^M] - v^*$ decreases as the sample size N increases. However, the second numerical experiment indicated that a significant bias can persist even if the sample size N is increased far beyond the sample size needed for the SAA method to produce an optimal solution.

The third numerical experiment demonstrates the effect of the number of decision variables and the condition number κ on the bias in the optimality gap estimator. Figures 4.5 and 4.6 show how the relative bias $\bar{v}_N^M/v^*$ of the optimality gap estimate changes as the sample size N increases, for different instances. The most noticeable effect is that the bias decreases much more slowly for the harder instances than for the randomly generated instances as the sample size N increases. This is in accordance with the asymptotic result (2.31) of Proposition 2.4.

Two estimators of the optimality gap $v^* - g(\hat{x})$ were discussed in section 3.3, namely, $\bar{v}_N^M - \hat{g}_{N'}(\hat{x})$ and $\bar{v}_N^M - \bar{g}_N^M(\hat{x})$. It was mentioned that the second estimator may have smaller variance than the first, especially if there is positive correlation between $\hat{g}_N^m(\hat{x})$ and $\hat{v}_N^m$. It was also pointed out that the second estimator requires additional computational effort, because after $\hat{x}$ is produced by solving the SAA problem for one sample, the second estimator requires the computation of $\hat{g}_N^m(\hat{x})$ for all the remaining samples $m = 1, \ldots, M$. The fourth numerical experiment compares the optimality gap estimates and their variances. Sample sizes of $N = 50$ and $N' = 2000$ were used, and $M = 50$ replications were performed.

Table 4.2 shows the optimality gap estimates $\bar{v}_N^M - \hat{g}_{N'}(\hat{x})$ and $\bar{v}_N^M - \bar{g}_N^M(\hat{x})$, with their variances $\widehat{\mathrm{Var}}[\bar{v}_N^M - \hat{g}_{N'}(\hat{x})] = S_{N'}^2(\hat{x})/N' + S_M^2/M$ and $\widehat{\mathrm{Var}}[\bar{v}_N^M - \bar{g}_N^M(\hat{x})] = \bar{S}_M^2/M$, respectively; the correlation $\widehat{\mathrm{Cor}}[\bar{v}_N^M, \bar{g}_N^M(\hat{x})]$; and the computation times of the gap estimates. In each case, the bias $\bar{v}_N^M - v^*$ formed the major part of the optimality gap estimate; the standard deviations of the gap estimators were small compared with the bias. There was positive correlation between $\bar{v}_N^M$ and $\bar{g}_N^M(\hat{x})$, and the second gap estimator had smaller variances, but this benefit is obtained at the expense of relatively large additional computational effort.

In section 2.2, an estimate $N \approx 3\sigma_{\max}^2 \log(|\mathcal{S}|/\alpha)/(\varepsilon - \delta)^2$ of the required sample size was derived. For the instances presented here, using $\varepsilon = 0.5$, $\delta = 0$, and $\alpha = 0.01$, these estimates were of the order of 10^6 and thus much larger than the sample sizes that were actually required for the specified accuracy. The sample size estimates using $\sigma_{\max}^2$ were smaller than the sample size estimates using $\max_{x \in \mathcal{S}} \mathrm{Var}[G(x, W)]$ by a factor of approximately 10.

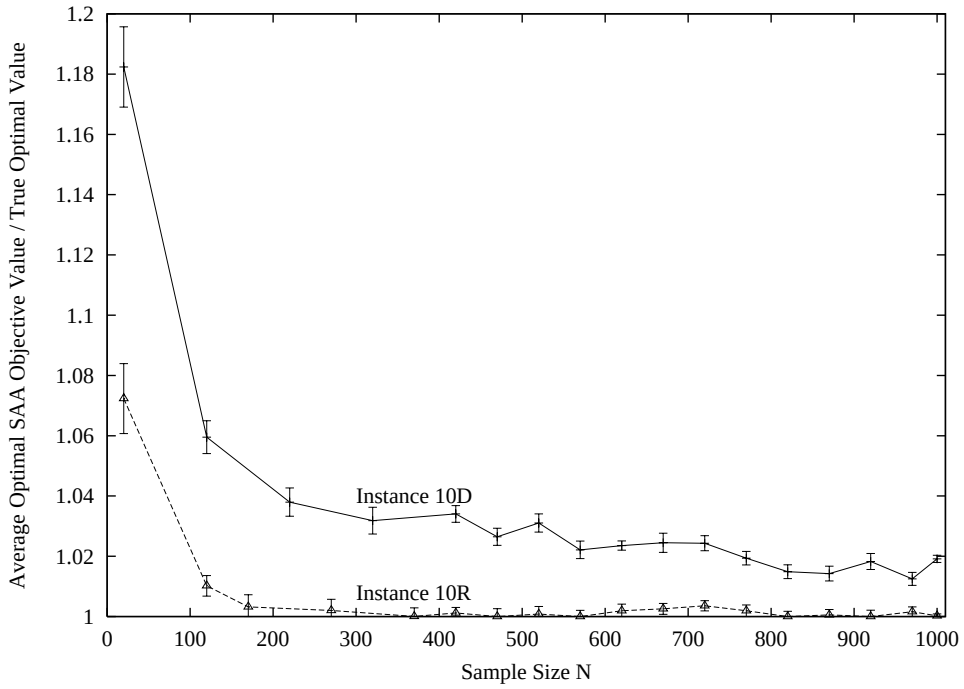


FIG. 4.5. *Relative bias* $\bar{v}_N^M/v^*$ *of the optimality gap estimator as a function of the sample size* N*, for instances* 10D *and* 10R*, with* 10 *decision variables.*

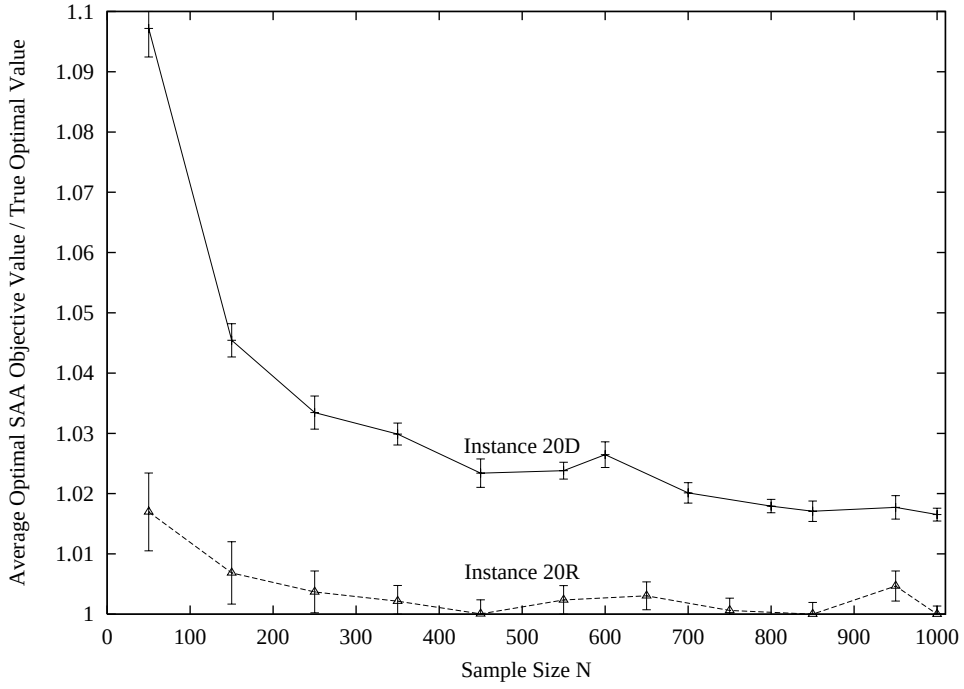


FIG. 4.6. *Relative bias* $\bar{v}_N^M/v^*$ *of the optimality gap estimate as a function of the sample size* N*, for instances* 20D *and* 20R*, with* 20 *decision variables.*

TABLE 4.2
Optimality gap estimates $\bar{v}_N^M - \hat{g}_{N'}(\hat{x})$ and $\bar{v}_N^M - \bar{g}_N^M(\hat{x})$, with their variances and computation times.

Instance	Opt. gap $v^* - g(\hat{x})$	Estimate $\bar{v}_N^M - \hat{g}_{N'}(\hat{x})$	$\widehat{\mathrm{Var}}[\bar{v}_N^M - \hat{g}_{N'}(\hat{x})]$ $= S^2_{N'}(\hat{x})/N' + S^2_M/M$	CPU time
10D	0	3.46	0.200	0.02
10R	0	1.14	0.115	0.01
20D	0.148	8.46	0.649	0.02
20R	0	3.34	1.06	0.02

Instance	Opt. gap $v^* - g(\hat{x})$	Estimate $\bar{v}_N^M - \bar{g}_N^M(\hat{x})$	$\widehat{\mathrm{Var}}[\bar{v}_N^M - \bar{g}_N^M(\hat{x})]$ $= \bar{S}^2_M/M$	Correlation $\widehat{\mathrm{Cor}}[\bar{v}_N^M, \bar{g}_N^M(\hat{x})]$	CPU time
10D	0	3.72	0.121	0.203	0.24
10R	0	1.29	0.035	0.438	0.24
20D	0.148	9.80	0.434	0.726	0.49
20R	0	3.36	0.166	0.844	0.47

Several variance reduction techniques can be used. Compared with simple random sampling, Latin hypercube sampling reduced the variances by factors varying between 1.02 and 2.9 and increased the computation time by a factor of approximately 1.2. Also, to estimate $g(x)$ for any solution $x \in \mathcal{S}$, it is natural to use $\sum_{i=1}^k W_i x_i$ as a control variate, because $\sum_{i=1}^k W_i x_i$ should be correlated with $[\sum_{i=1}^k W_i x_i - q]^+$, and the mean of $\sum_{i=1}^k W_i x_i$ is easy to compute. Using this control variate reduced the variances of the estimators of $g(x)$ by factors between 2.0 and 3.0 and increased the computation time by a factor of approximately 2.0.

5. Conclusion. We proposed a sample average approximation method for solving stochastic discrete optimization problems, and we studied some theoretical as well as practical issues important for the performance of this method. It was shown that the probability that a replication of the SAA method produces an optimal solution increases at an exponential rate in the sample size N. It was found that this convergence rate depends on the conditioning of the problem, which in turn tends to become poorer with an increase in the number of decision variables. It was also shown that the sample size required for a specified accuracy increases proportional to the logarithm of the number of feasible solutions. It was found that for many instances the SAA method produces good and often optimal solutions with only a few replications and a small sample size. However, the optimality gap estimator considered here was in each case too weak to indicate that a good solution had been found. Consequently the sample size had to be increased substantially before the optimality gap estimator indicated that the solutions were good. Thus, a more efficient optimality gap estimator can make a substantial contribution toward improving the performance guarantees of the SAA method during execution of the algorithm. The SAA method has the advantage of ease of use in combination with existing techniques for solving deterministic optimization problems.

The proposed method involves solving several replications of the SAA problem (2.1), and possibly increasing the sample size several times. An important issue is the behavior of the computational complexity of the SAA problem (2.1) as a function of the sample size. Current research aims at investigating this behavior for particular classes of problems.

REFERENCES

[1] M. H. Alrefaei and S. Andradóttir, *A simulated annealing algorithm with constant temperature for discrete stochastic optimization*, Management Science, 45 (1999), pp. 748–764.

[2] R. E. Bechhofer, T. J. Santner, and D. M. Goldsman, *Design and Analysis of Experiments for Statistical Selection, Screening and Multiple Comparisons*, John Wiley, New York, NY, 1995.

[3] J. R. Birge and F. Louveaux, *Introduction to Stochastic Programming*, Springer Ser. Oper. Res., Springer-Verlag, New York, NY, 1997.

[4] A. Cohn and C. Barnhart, *The stochastic knapsack problem with random weights: A heuristic approach to robust transportation planning*, in Proceedings of the Triennial Symposium on Transportation Analysis (TRISTAN III), San Juan, PR, 1998.

[5] A. Dembo and O. Zeitouni, *Large Deviations Techniques and Applications*, Springer-Verlag, New York, NY, 1998.

[6] B. L. Fox and G. W. Heine, *Probabilistic search with overrides*, Ann. Appl. Probab., 5 (1995), pp. 1087–1094.

[7] A. Futschik and G. C. Pflug, *Confidence sets for discrete stochastic optimization*, Ann. Oper. Res., 56 (1995), pp. 95–108.

[8] A. Futschik and G. C. Pflug, *Optimal allocation of simulation experiments in discrete stochastic optimization and approximative algorithms*, European J. Oper. Res., 101 (1997), pp. 245–260.

[9] S. B. Gelfand and S. K. Mitter, *Simulated annealing with noisy or imprecise energy measurements*, J. Optim. Theory Appl., 62 (1989), pp. 49–62.

[10] W. Gutjahr and G. C. Pflug, *Simulated annealing for noisy cost functions*, J. Global Optim., 8 (1996), pp. 1–13.

[11] I. D. Hill, *Algorithm AS66: The normal integral*, Applied Statistics, 22 (1973), pp. 424–427.

[12] Y. Hochberg and A. Tamhane, *Multiple Comparison Procedures*, John Wiley, New York, NY, 1987.

[13] T. Homem-de-Mello, *Variable-Sample Methods and Simulated Annealing for Discrete Stochastic Optimization*, manuscript, Department of Industrial, Welding and Systems Engineering, The Ohio State University, Columbus, OH, 1999.

[14] T. Homem-de-Mello, *Monte Carlo methods for discrete stochastic optimization*, in Stochastic Optimization: Algorithms and Applications, S. Uryasev and P. M. Pardalos, eds., Kluwer Academic Publishers, Norwell, MA, 2000, pp. 95–117.

[15] W. K. Mak, D. P. Morton, and R. K. Wood, *Monte Carlo bounding techniques for determining solution quality in stochastic programs*, Oper. Res. Lett., 24 (1999), pp. 47–56.

[16] D. P. Morton and R. K. Wood, *On a stochastic knapsack problem and generalizations*, in Advances in Computational and Stochastic Optimization, Logic Programming, and Heuristic Search: Interfaces in Computer Science and Operations Research, D. L. Woodruff, ed., Kluwer Academic Publishers, Dordrecht, the Netherlands, 1998, pp. 149–168.

[17] B. L. Nelson, J. Swann, D. M. Goldsman, and W. Song, *Simple procedures for selecting the best simulated system when the number of alternatives is large,* Oper. Res., to appear.

[18] V. I. Norkin, Y. M. Ermoliev, and A. Ruszczyński, *On optimal allocation of indivisibles under uncertainty*, Oper. Res., 46 (1998), pp. 381–395.

[19] V. I. Norkin, G. C. Pflug, and A. Ruszczyński, *A branch and bound method for stochastic global optimization*, Math. Programming, 83 (1998), pp. 425–450.

[20] R. Schultz, L. Stougie, and M. H. Van der Vlerk, *Solving stochastic programs with integer recourse by enumeration: A framework using Gröbner basis reductions*, Math. Programming, 83 (1998), pp. 229–252.

[21] A. Shapiro, *Asymptotic analysis of stochastic programs*, Ann. Oper. Res., 30 (1991), pp. 169–186.

[22] A. Shapiro, T. Homem-de-Mello, and J. C. Kim, *Conditioning of Convex Piecewise Linear Stochastic Programs*, manuscript, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 2000.

SIAM J. OPTIM.
Vol. 12, No. 2, pp. 503–521

RANK-TWO RELAXATION HEURISTICS FOR MAX-CUT AND OTHER BINARY QUADRATIC PROGRAMS*

SAMUEL BURER†, RENATO D. C. MONTEIRO‡, AND YIN ZHANG§

Abstract. The Goemans–Williamson randomized algorithm guarantees a high-quality approximation to the MAX-CUT problem, but the cost associated with such an approximation can be excessively high for large-scale problems due to the need for solving an expensive semidefinite relaxation. In order to achieve better practical performance, we propose an alternative, rank-two relaxation and develop a specialized version of the Goemans–Williamson technique. The proposed approach leads to continuous optimization heuristics applicable to MAX-CUT as well as other binary quadratic programs, for example the MAX-BISECTION problem.

A computer code based on the rank-two relaxation heuristics is compared with two state-of-the-art semidefinite programming codes that implement the Goemans–Williamson randomized algorithm, as well as with a purely heuristic code for effectively solving a particular MAX-CUT problem arising in physics. Computational results show that the proposed approach is fast and scalable and, more importantly, attains a higher approximation quality in practice than that of the Goemans–Williamson randomized algorithm. An extension to MAX-BISECTION is also discussed, as is an important difference between the proposed approach and the Goemans–Williamson algorithm; namely, that the new approach does not guarantee an upper bound on the MAX-CUT optimal value.

Key words. binary quadratic programs, MAX-CUT and MAX-BISECTION, semidefinite relaxation, rank-two relaxation, continuous optimization heuristics

AMS subject classifications. 90C06, 90C27, 90C30

PII. S1052623400382467

1. Introduction.

Many combinatorial optimization problems can be formulated as quadratic programs with binary variables, a simple example being the MAX-CUT problem. Since such problems are usually NP-hard, which means that exact solutions are difficult to obtain, different heuristic or approximation algorithms have been proposed, often based on continuous relaxations of the original discrete problems. A relatively new relaxation scheme is called the semidefinite programming relaxation (or SDP relaxation), in which a vector-valued binary variable is replaced by a matrix-valued continuous variable, resulting in a convex optimization problem called a semidefinite program (SDP) that can be solved to a prescribed accuracy in polynomial time. Some early ideas related to such a relaxation can be found in a number of works, including [10, 23, 24, 26, 27].

Based on solving the SDP relaxation, Goemans and Williamson [18] proposed a randomized algorithm for the MAX-CUT problem and established the celebrated 0.878 performance guarantee. Since then, SDP relaxation has become a powerful and popular theoretical tool for devising polynomial-time approximation algorithms

*Received by the editors December 13, 2000; accepted for publication (in revised form) June 25, 2001; published electronically December 14, 2001. Computational results reported in this paper were obtained on an SGI Origin2000 computer at Rice University, acquired in part with support from NSF grant DMS-9872009.

http://www.siam.org/journals/siopt/12-2/38246.html

†School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332 (burer@math.gatech.edu). This author was supported in part by NSF grants CCR-9902010 and INT-9910084.

‡School of ISyE, Georgia Institute of Technology, Atlanta, GA 30332 (monteiro@ isye.gatech.edu). This author was supported in part by NSF grants CCR-9902010 and INT-9910084.

§Department of Computational and Applied Mathematics, Rice University, Houston, TX 77005 (zhang@caam.rice.edu). This author was supported in part by DOE grant DE-FG03-97ER25331, DOE/LANL contract 03891-99-23, and NSF grant DMS-9973339.

for hard combinatorial optimization problems, and even in cases where performance guarantees are not known, randomized algorithms based on the SDP relaxation can often give good-quality approximate solutions in practice. It is important to note that such Goemans–Williamson-type approaches produce both upper and lower bounds on the optimal value of the underlying discrete problem.

In the meantime, there have been hopes that the SDP relaxation would eventually lead to practically efficient algorithms for solving large-scale combinatorial optimization problems by producing tight lower and upper bounds. In this regard, however, results thus far have not always been encouraging. The main difficulty lies in the fact that the number of variables and/or constraints in an SDP relaxation is one order of magnitude higher than that of the original problem. Hence, the cost of solving such SDP problems grows quickly as the size of the problems increases. In other words, a key issue here is the scalability of the SDP relaxation approach with respect to the problem size.

There has been a great deal of research effort towards improving the efficiency of SDP solvers, including work on exploiting sparsity in more traditional interior-point methods [1, 9, 16, 17, 29] and work on alternative methods [5, 6, 7, 20, 21, 30, 31]. Indeed, the efficiency of SDP solvers has been improved significantly in the last few years. Nevertheless, the scalability problem still remains.

On the other hand, computational studies have continued to affirm that the quality of bounds produced by the SDP relaxation is quite high. For example, the Goemans–Williamson approximation algorithm produces lower bounds (i.e., discrete solutions) that are better than or at least comparable to that of a number of heuristics (see [11], for example). It is thus natural to investigate whether the quality of the SDP relaxation can be preserved while somehow extending its use to problems of very large size.

Can the approaches inspired by Goemans and Williamson, which rely on solving the SDP relaxation, ever become competitive in attacking large-scale problems? In this paper, we hope to provide a partial answer to this question. We will argue that in terms of producing a lower bound, the answer seems to be negative, at least for some problem classes including the familiar MAX-CUT problem. In other words, if one is interested only in obtaining a high-quality approximate solution, then the SDP relaxation does not seem to hold much promise. Our argument is based on strong empirical evidence showing that on a large set of test problems the performance of the SDP relaxation approach is clearly inferior to that of a new rank-two relaxation approach that we will propose and study in this paper. The advantages of this rank-two approach appear not only in terms of computational costs but, more notably, also in terms of the approximation quality.

Based on the proposed rank-two relaxation and a specialized version of the Goemans–Williamson technique, we construct a continuous optimization heuristic for approximating the MAX-CUT problem and establish some properties for this approach that are useful in designing algorithms. We then compare a code based on our heuristic with some state-of-the-art SDP-based approximation codes on a set of MAX-CUT test problems. We also compare our code with a well-established, heuristic code for MAX-CUT on a set of test problems from physics. Finally, we consider extensions to other related problems—in particular, to the MAX-BISECTION problem.

This paper is organized as follows. Section 2 briefly introduces the MAX-CUT problem and its corresponding SDP relaxation. In section 3, we present the rank-two relaxation scheme and study its properties, including a useful characterization for a

maximum cut. In section 4, we present our heuristic algorithm for the MAX-CUT problem, and computational results on MAX-CUT are given in section 5. We extend the heuristic to the MAX-BISECTION problem in section 5.3 and give numerical results as well. Lastly, we conclude the paper in section 7.

2. Max-cut and the semidefinite relaxation. Let an undirected and connected graph $G = (V, E)$, where $V = \{1, 2, \ldots, n\}$ and $E \subset \{(i, j) : 1 \le i < j \le n\}$, be given. Let the edge weights $w_{ij} = w_{ji}$ be given such that $w_{ij} = 0$ for $(i, j) \notin E$, and in particular, let $w_{ii} = 0$. The MAX-CUT problem is to find a bipartition (V_1, V_2) of V so that the sum of the weights of the edges between V_1 and V_2 is maximized. It is well known that the MAX-CUT problem can be formulated as

$$\begin{aligned} \max \quad & \frac{1}{2} \sum_{1 \le i < j \le n} w_{ij}(1 - x_i x_j) \\ \text{subect to (s.t.)} \quad & |x_i| = 1, \ \ i = 1, \ldots, n, \end{aligned} \tag{1}$$

which has the same solution as the following binary quadratic program:

$$\begin{aligned} \min \quad & \sum_{1 \le i < j \le n} w_{ij} x_i x_j \\ \text{s.t.} \quad & |x_i| = 1, \ i = 1, \ldots, n. \end{aligned} \tag{2}$$

Moreover, it is easy to verify that (2) can be rewritten into the matrix optimization problem

$$\begin{aligned} \min \quad & \tfrac{1}{2} \, W \bullet X, \\ \text{s.t.} \quad & \operatorname{diag}(X) = e, \\ & \operatorname{rank}(X) = 1, \\ & X \succeq 0, \end{aligned} \tag{3}$$

where $W = [w_{ij}]$, $W \bullet X = \sum_{i,j=1}^{n} w_{ij} x_{ij}$, $\operatorname{diag}(X)$ is the vector in $\Re^n$ consisting of the diagonal elements of X, e is the vector of all ones, and $X \succeq 0$ means that X is symmetric positive semidefinite.

Since the MAX-CUT problem is NP-hard, various heuristics and approximation algorithms have been proposed to attack it. Recent ground-breaking work comes from Goemans and Williamson [18], who replace the "unit scalars" x_i in (2) by unit vectors $v_i \in \Re^n$ and the scalar products $x_i x_j$ by the inner products $v_i^T v_j$. The resulting problem is the following relaxation of the MAX-CUT problem:

$$\begin{aligned} \min \quad & \sum_{1 \le i < j \le n} w_{ij} v_i^T v_j \\ \text{s.t.} \quad & \|v_i\|_2 = 1, \ i = 1, \ldots, n, \end{aligned} \tag{4}$$

where $v_i \in \Re^n$. Furthermore, a change of variables $X = [v_i^T v_j] \in \Re^{n \times n}$ leads to the following so-called SDP relaxation for the MAX-CUT problem:

$$\begin{aligned} \min \quad & \tfrac{1}{2} \, W \bullet X, \\ \text{s.t.} \quad & \operatorname{diag}(X) = e, \\ & X \succeq 0. \end{aligned} \tag{5}$$

It is well known that such an SDP problem is approximately solvable in polynomial time (see [25], for example). Comparing (5) with (3), we clearly see that the SDP

relaxation is nothing more than the problem obtained from (3) by dropping the rank-one restriction on X.

It is worth observing that a solution $(v_1, \ldots, v_n)$ of (4) consists of n points on the surface of the unit sphere in $\Re^n$, each representing a node in the graph. Goemans and Williamson [18] proposed the following randomized algorithm for generating cuts in the graph: after a solution of (4) is obtained, one randomly partitions the unit sphere into two half-spheres H_1 and H_2 (the boundary in-between can be on either side) and forms the bipartition consisting of $V_1 = \{i : v_i \in H_1\}$ and $V_2 = \{i : v_i \in H_2\}$. Furthermore, Goemans and Williamson established the celebrated result that if all the weights are nonnegative, then the expected value of such randomly generated cuts is at least 0.878 times the maximum cut value. That result gives a strong performance guarantee for this randomization procedure. In fact, it has recently been shown in [13] that the factor 0.878 is indeed the best possible in several senses.

3. A rank-two relaxation. In this section, we present an alternative rank-two relaxation scheme that leads to a nonlinear optimization problem having only n variables but also a nonconvex objective function. Since the number of variables is not increased from the MAX-CUT problem, this approach possesses scalability for relaxing large-scale problems. On the other hand, since the relaxation is nonconvex, we cannot expect to find an optimal solution in practice, and so we can no longer ensure a computable upper bound on the original problem. For solving this problem to gain information about the underlying MAX-CUT problem, the trade-off is obviously between computational efficiency and a theoretical guarantee. When the main objective is to obtain high-quality approximate solutions, however, we hope to demonstrate through computational experiments that the gain clearly outweighs the loss.

We replace the "unit scalar" variables x_i in (2) by unit vectors $v_i \in \Re^2$ (not $\Re^n$), and the scalar products $x_i x_j$ by the inner products $v_i^T v_j$. As before, the constraint $|x_i| = 1$ becomes $\|v_i\|_2 = 1$; namely, all the vectors v_i should be on the unit circle. In this way, we obtain a relaxation of the MAX-CUT problem that has exactly the same form as (4) except that now all vectors v_i are in $\Re^2$ instead of $\Re^n$. Alternatively, this relaxation can be viewed as replacing the rank-one restriction on X in (3) by the rank-two restriction $\operatorname{rank}(X) \leq 2$; hence we call it a rank-two relaxation.

Using polar coordinates, we can represent a set of n unit vectors $v_1, \ldots, v_n$ in $\Re^2$ by means of a vector $\theta = (\theta_1, \ldots, \theta_n)^T \in \Re^n$ consisting of n angles, namely,

$$v_i = \begin{pmatrix} \cos\theta_i \\ \sin\theta_i \end{pmatrix} \quad \forall\ i = 1, \ldots, n.$$

In this case, we have

$$v_i^T v_j \equiv \cos(\theta_i - \theta_j) \quad \forall\ i, j = 1, \ldots, n.$$

Let $T(\theta)$ be the skew-symmetric matrix-valued function of θ defined by

$$T_{ij}(\theta) = \theta_i - \theta_j \quad \forall\ i, j = 1, \ldots, n,$$

and let $f : \Re^n \to \Re$ be the function defined as

$$f(\theta) \equiv \frac{1}{2}\, W \bullet \cos(T(\theta)) \quad \forall\ \theta \in \Re^n, \tag{6}$$

where $\cos(T(\theta))$ is the $n \times n$ matrix whose entries are the cosine of the corresponding entries of $T(\theta)$. Then, in terms of the polar coordinates, we obtain the following

relaxation for the MAX-CUT problem:

$$\min_{\theta \in \Re^n} f(\theta). \tag{7}$$

This is an unconstrained optimization problem with a nonconvex objective function. In general, it has multiple local, nonglobal minima.

The derivatives of the function $f(\theta)$ can be easily computed. Indeed, the first partial derivatives of $f(\theta)$ are given by

$$\frac{\partial f(\theta)}{\partial \theta_j} = \sum_{k=1}^{n} w_{kj} \sin(\theta_k - \theta_j) \quad \forall\, j = 1, \ldots, n,$$

and hence,

$$g(\theta) \equiv \nabla f(\theta) = [W \circ \sin(T(\theta))]^T e, \tag{8}$$

where the notation "$\circ$" indicates the Hadamard, i.e., entrywise, product of W and $\sin(T(\theta))$. The second partial derivatives of $f(\theta)$ are given by

$$\frac{\partial^2 f(\theta)}{\partial \theta_i \partial \theta_j} = \begin{cases} w_{ij} \cos(\theta_i - \theta_j) & \text{if } i \neq j, \\ -\sum_{k \neq j} w_{kj} \cos(\theta_k - \theta_j) & \text{if } i = j \end{cases}$$

for every $i, j = 1, \ldots, n$, and hence the Hessian of $f(\theta)$ is given by

$$H(\theta) \equiv \nabla^2 f(\theta) = W \circ \cos(T(\theta)) - \mathrm{Diag}\left([W \circ \cos(T(\theta))]e\right), \tag{9}$$

where, for any vector v, $\mathrm{Diag}(v)$ is the diagonal matrix with v on its diagonal. Note that the major effort in the evaluation of f, g, and H is the computation of the quantities $W \circ \cos(T(\theta))$ and $W \circ \sin(T(\theta))$.

There are interesting relationships between cuts in the graph and the function $f(\theta)$, which we will now describe. We call a vector $\bar{\theta} \in \Re^n$ an *angular representation* of a cut, or simply a cut, if there exist integers k_{ij} such that

$$\bar{\theta}_i - \bar{\theta}_j = k_{ij}\pi \quad \forall\, i, j = 1, \ldots, n. \tag{10}$$

Clearly, in this case $\cos(\bar{\theta}_i - \bar{\theta}_j) = \pm 1$ and there exists a binary vector $\bar{x} \in \{-1, 1\}^n$ such that

$$\cos(\bar{\theta}_i - \bar{\theta}_j) \equiv \bar{x}_i \bar{x}_j = \pm 1 \quad \forall\, i, j = 1, \ldots, n.$$

Moreover, the cut value corresponding to a cut $\bar{\theta}$ is

$$\psi(\bar{\theta}) \equiv \frac{1}{2} \sum_{i>j} w_{ij}[1 - \cos(\bar{\theta}_i - \bar{\theta}_j)]. \tag{11}$$

We note that the function $f(\theta)$ is invariant with respect to simultaneous, uniform rotation on every component of θ, i.e., $f(\theta) \equiv f(\theta + \tau e)$ for any scalar τ, and is periodic with a period of 2π with respect to each variable θ_i. Modulo the uniform rotation and the periodicity for each variable, there is an obvious one-to-one correspondence between the binary and angular representations of a cut; namely,

$$\bar{\theta}_i = \begin{cases} 0 & \text{if } \bar{x}_i = +1, \\ \pi & \text{if } \bar{x}_i = -1, \end{cases}$$

and vice versa. With the above correspondence in mind, in what follows we will use $\bar{\theta}$ and $\bar{x}$ interchangeably to represent a cut. Moreover, given an angular representation of a cut $\bar{\theta}$ (or a binary one $\bar{x}$), we will use the notation $x(\bar{\theta})$ (or $\theta(\bar{x})$) to denote the corresponding binary (or angular) representation of the same cut.

Since $\sin(\bar{\theta}_i - \bar{\theta}_j) = 0$ for any $\bar{\theta}$ satisfying (10), it follows directly from (8) that $g(\bar{\theta}) = 0$ at any cut $\bar{\theta}$. We state this simple observation in the following proposition.

PROPOSITION 3.1. *Every cut $\bar{\theta} \in \Re^n$ is a stationary point of the function $f(\theta)$.*

We will now derive in the lemma below a characterization of a maximum (minimum) cut which will be useful in the later development. We first need the following definition.

DEFINITION 3.2. *A matrix $M \in \Re^{n\times n}$ is called* nonnegatively summable *if the sum of the entries in every principal submatrix of M is nonnegative, or equivalently, if $u^T M u \geq 0$ for every binary vector $u \in \{0,1\}^n$.*

Clearly, every positive semidefinite matrix is nonnegatively summable. On the other hand, the matrix $ee^T - I$ is nonnegatively summable, but not positive semidefinite.

LEMMA 3.3. *Let $\bar{x} \in \{-1,1\}^n$ be given and consider the matrix $M(\bar{x}) \in \Re^{n\times n}$ defined as*

$$M(\bar{x}) = W \circ (\bar{x}\bar{x}^T) - \mathrm{Diag}([W \circ (\bar{x}\bar{x}^T)]e). \tag{12}$$

Then, $\bar{x}$ is a maximum (respectively, minimum) cut if and only if $M(\bar{x})$ (respectively, $-M(\bar{x})$) is nonnegatively summable.

Proof. Let $q : \Re^n \to \Re$ be the quadratic function defined as $q(x) = (x^T W x)/2$ for all $x \in \Re^n$, and note that $\bar{x}$ is a maximum cut if and only if $\bar{x}$ minimizes $q(x)$ over the set of all $x \in \{-1,1\}^n$. Now, let $x \in \{-1,1\}^n$ be given and observe that

$$\bar{x} - x = 2\delta \circ \bar{x},$$

where "$\circ$" represents the Hadamard product and $\delta \in \Re^n$ is defined as

$$\delta_i \equiv \begin{cases} 0 & \text{if } x_i = \bar{x}_i, \\ 1 & \text{if } x_i \neq \bar{x}_i. \end{cases} \tag{13}$$

Using this identity and the fact that $\delta^T v = \delta^T \mathrm{Diag}(v)\delta$ for any $v \in \Re^n$, we obtain

$$\begin{aligned} q(x) - q(\bar{x}) &= (W\bar{x})^T(x - \bar{x}) + \frac{1}{2}(x - \bar{x})^T W(x - \bar{x}) \\ &= -2\bar{x}^T W(\delta \circ \bar{x}) + 2(\delta \circ \bar{x})^T W(\delta \circ \bar{x}) \\ &= -2\delta^T([W \circ \bar{x}\bar{x}^T]e) + 2\delta^T[W \circ \bar{x}\bar{x}^T]\delta \\ &= -2\delta^T \mathrm{Diag}\left([W \circ \bar{x}\bar{x}^T]e\right)\delta + 2\delta^T[W \circ \bar{x}\bar{x}^T]\delta \;=\; 2\delta^T M(\bar{x})\delta. \end{aligned}$$

Noting that every $x \in \{-1,1\}^n$ corresponds to a unique vector $\delta \in \{0,1\}^n$ via (13), and vice versa, we conclude from the above identity that $\bar{x}$ minimizes $q(x)$ over $x \in \{-1,1\}^n$ if and only if $\delta^T M(\bar{x})\delta \geq 0$ for all $\delta \in \{0,1\}^n$, or equivalently, $M(\bar{x})$ is nonnegatively summable.

The proof of the second equivalence is analogous. Hence, the result follows. ∎

Although every cut is a stationary point of $f(\theta)$, the following theorem guarantees that only the maximum cuts can possibly be local minima of $f(\theta)$. In fact, the theorem gives a complete classification of cuts as stationary points of the function $f(\theta)$.

THEOREM 3.4. *Let $\bar{\theta}$ be a cut and let $\bar{x} \equiv x(\bar{\theta})$ be the associated binary cut. If $\bar{\theta}$ is a local minimum (respectively, local maximum) of $f(\theta)$, then $\bar{x}$ is a maximum (respectively, minimum) cut. Consequently, if $\bar{x}$ is neither a maximum cut nor a minimum cut, then $\bar{\theta}$ must be a saddle point of $f(\theta)$.*

Proof. Since $\bar{x}_i\bar{x}_j = \cos(\bar{\theta}_i - \bar{\theta}_j)$, we have $\nabla^2 f(\bar{\theta}) \equiv M(x(\bar{\theta}))$ due to (9) and (12). If $\bar{\theta}$ is a local minimum of f, then the Hessian $\nabla^2 f(\bar{\theta})$ is positive semidefinite and hence nonnegatively summable. The first implication of the theorem then follows from the first equivalence of Lemma 3.3. The second implication of the theorem can be proved in a similar way using the second equivalence of Lemma 3.3. Hence, the result follows. □

The converses of the two implications in the above theorem do not hold. Indeed, consider the unweighted graph K_3 (the complete graph with three nodes) for which the cut $\bar{x} = [1\ -1\ -1]^T$ is maximum. From (12), we have

$$M(\bar{x}) = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix},$$

which is indeed nonnegatively summable but not positive semidefinite. Hence the corresponding angular representation $\bar{\theta}$ is not a local minimum of the function $f(\theta)$ in view of the fact that $M(\bar{x}) \equiv \nabla^2 f(\bar{\theta})$.

There are indeed instances where maximum cuts are local minima of $f(\theta)$, as indicated by the following observation.

PROPOSITION 3.5. *For a bipartite graph with nonnegative weights, the global minimum value of $f(\theta)$ is attained by a maximum cut.*

Proof. A maximum cut is one that cuts through all the edges in the bipartite graph. For this cut, $\cos(\theta_i - \theta_j) = -1$ for all edges $(i, j) \in E$. Hence the global minimum value of $f(\theta)$ is attained at $-e^T We/2$. □

Obviously, for problems where a maximum cut $\bar{x}$ corresponds to a local minimum of $f(\theta)$, the optimality of $\bar{x}$ can be checked in polynomial time by determining whether $M(\bar{x})$ is positive semidefinite or not.

Since nonmaximum cuts cannot possibly be local minima of $f(\theta)$, a good minimization algorithm would not be attracted to stationary points corresponding to nonmaximum cuts that are either local maxima or saddle points of $f(\theta)$. This fact will play an important role in the construction of our algorithms.

4. A heuristic algorithm for MAX-CUT. To produce an approximate solution to the MAX-CUT problem, we first minimize the function $f(\theta)$ and obtain a local minimum θ corresponding to a distribution of points on the unit circle. Using periodicity, we may easily assume that $\theta_i \in [0, 2\pi)$ for each $i = 1, \ldots, n$. Any partition of the unit circle into two equal halves gives a cut as follows. Pick an angle $\alpha \in [0, \pi)$ and let

$$x_i = \begin{cases} +1 & \text{if } \theta_i \in [\alpha, \alpha + \pi), \\ -1 & \text{otherwise.} \end{cases} \tag{14}$$

The corresponding value of the cut x is given by

$$\gamma(x) \equiv \frac{1}{2} \sum_{i>j} w_{ij}(1 - x_i x_j). \tag{15}$$

An advantage of the rank-two relaxation over the SDP relaxation is that it is straightforward and inexpensive to examine all possible cuts generated in the above fashion,

making it easy to find the best one. The following, deterministic (rather than random) procedure finds a best possible Goemans–Williamson-type cut associated with a given θ. Without loss of generality, let us assume that θ satisfies $\theta_i \in [0, 2\pi),\ i = 1, \ldots, n$, and that

$$\theta_1 \leq \theta_2 \leq \cdots \leq \theta_n,$$

after a reordering if necessary.

PROCEDURE-CUT (input θ, output x^*).

Let $\alpha = 0$, $\Gamma = -\infty$, $i = 1$. Let j be the smallest index such that $\theta_j > \pi$ if there is one; otherwise set $j = n + 1$. Set $\theta_{n+1} = 2\pi$.

While $\alpha \leq \pi$

1. Generate cut x by (14) and compute $\gamma(x)$.
2. If $\gamma(x) > \Gamma$, then let $\Gamma = \gamma(x)$ and $x^* = x$.
3. If $\theta_i \leq \theta_j - \pi$, let $\alpha = \theta_i$ and increment i by 1; otherwise let $\alpha = \theta_j - \pi$ and increment j by 1.

End

Since our rank-two relaxation has the same form as Goemans and Williamson's relaxation (4), except that ours has variables in $\Re^2$ rather than $\Re^n$, the same analysis of Goemans and Williamson, with minimal changes, can be applied to show that the cut value generated by the above procedure is at least 0.878 times the relaxed cut value $\psi(\theta)$ as is defined in (11). That is,

$$\gamma(x^*) \geq 0.878\, \psi(\theta).$$

However, since we cannot guarantee that $\psi(\theta)$ is an upper bound on the maximum cut value, there is no performance guarantee. Nevertheless, we do have the property that, in a weak sense and to some extent, the better the local maximum of $\psi(\theta)$ (or, equivalently, local minimum of $f(\theta)$) we obtain, the better a cut will likely be produced. To see this, let x_a^* and x_b^* be two binary cuts generated by Procedure-CUT from θ_a and θ_b, respectively. If $\gamma(x_a^*) \leq \psi(\theta_a)$ and $\psi(\theta_b) > \frac{1}{0.878}\psi(\theta_a)$, then since

$$\gamma(x_b^*) \geq 0.878\ \psi(\theta_b) > \psi(\theta_a) \geq \gamma(x_a^*),$$

x_b^* is a better cut than x_a^*.

After we minimize the function $f(\theta)$ and obtain a local minimum θ^1, we will call on Procedure-CUT to produce a best possible cut x^1 associated with θ^1. At this point, we may stop and return the generated cut x^1. On the other hand, if we are willing to spend more time, we may try to improve the quality of our approximation.

We know that the angular representation of the cut x^1, $\theta(x^1)$, is a stationary point—most likely a saddle point—of the function $f(\theta)$, but not a minimizer unless it is already a maximum cut. Assuming that $\theta(x^1)$ is in fact a saddle point, it is probable that close by there are local minima of f that are deeper than θ^1 is. Although we cannot restart the minimization directly from the stationary point $\theta(x^1)$, we can certainly restart from a slight perturbation of $\theta(x^1)$ and hopefully escape to a better local minimum θ^2, which in turn would hopefully lead to a better cut x^2 or $\theta(x^2)$. We can continue this process until we reach a cut from which we deem that further improvement seems unlikely. We state this heuristic as the following algorithm.

ALGORITHM 1 (input N, θ^0, output x^*):

Given $\theta^0 \in \Re^n$ and integer $N \geq 0$, let $k = 0$ and $\Gamma = -\infty$.

While $k \leq N$

1. Starting from θ^0, minimize f to get θ.
2. Compute a best cut x associated with θ by Procedure-CUT.
3. If $\gamma(x) > \Gamma$, let $\Gamma = \gamma(x)$, $x^* = x$, and $k = 0$; otherwise set $k = k + 1$.
4. Set θ^0 to a random perturbation of the angular representation of x.

End

The parameter N controls how many consecutive, nonimproving random perturbations are allowed before we stop the algorithm. If so desired, the algorithm can be run M times with multiple starting points θ^0 to increase the chances of achieving better cuts. Generally speaking, the larger N and M are, the longer time the algorithm will take to run, and the better cut it will return.

A geometric interpretation of Algorithm 1 is as follows. After we arrive at a local minimum of f, we search around this local minimum for a nearby saddle point (i.e., a cut) that has the lowest f-value in the neighborhood. We then move to the saddle point and restart the minimization to locate a nearby local minimum that, hopefully, has a smaller f-value than the previous one. We repeat this process until we deem that the search has become unfruitful.

5. Computational results for MAX-CUT. We have implemented Algorithm 1 in a Fortran90 code named "CirCut." For the minimization of $f(\theta)$, we use a simple gradient algorithm with a backtracking Armijo line-search. Since numerical experiments indicate that the accuracy of the minimization is not crucial, we stop the minimization when the relative change in the function value is less than 10^{-4}.

In CirCut, we also include an option for a simple local search in the cut space; that is, after a cut is returned from Procedure-CUT, one has the option to improve it through a quick local search that moves one or two nodes at a time, producing a so-called locally 2-optimal solution. This feature can often slightly improve the quality of a cut and is therefore set to be a default feature unless specified otherwise.

We compare our code CirCut with two SDP codes, SBmethod and DSDP, both implementing the Goemans–Williamson randomized algorithm (along with other features). Since these codes produce both an upper bound and a lower bound, while our code only gives the latter, the comparisons should not be taken at face value. In carrying out such comparisons, we have two objectives in mind. First, since our heuristic is derived from the Goemans–Williamson randomized algorithm by a rank restriction, we want to see how our modifications affect the performance, both time-wise and quality-wise, of generating lower bounds. Second, since the approximation quality of the Goemans–Williamson randomized algorithm has been shown to be at least as good as a number of heuristics [11], through the comparisons we hope to get a good picture of the approximation quality of our heuristic. We select the codes SBmethod and DSDP for our comparisons because they represent the state of the art in solving large-scale SDP problems.

We also compare our code with a state-of-the-art heuristic code for MAX-CUT problems from the Ising spin glass model in physics, developed by Hartmann [19]. The purpose of this comparison is self-evident.

5.1. Comparison with SBmethod. We first report numerical results on the MAX-CUT problem in comparison with SBmethod, an SDP code developed by Helmberg and Rendl [20]. SBmethod solves a large class of semidefinite programs using a specialized bundle method, the so-called spectral bundle method, and in particular is one of the fastest codes for solving MAX-CUT SDP relaxations.

TABLE 1
Statistics for the torus set of MAX-CUT problems.

Graph name	Size	Lower bound	Upper bound	SDP bound
pm3-8-50	(512, 1536)	456	461	527
pm3-15-50	(3375, 10125)	2988	3069.51	3474
g3-8	(512, 1536)	41684814	41684814	45735817
g3-15	(3375, 10125)	2.85790e+8	2.87725e+8	3.1346e+8

TABLE 2
Comparison with SBmethod on MAX-CUT problems from the torus set.

Graph	SBmethod		CirCut ($N = 4$, $M = 100$)		
Name	Value	Time	Avg. value	Avg. time	Best value
pm3-8-50	434	28.72	443	0.218	452
pm3-15-50	2728	2131.89	2888	2.332	2936
g3-8	4.04736e+7	36.03	4.09098e+7	0.298	4.13946e+7
g3-15	2.73412e+8	3604.54	2.74357e+8	2.835	2.77917e+8

The first set of test problems comes from the DIMACS library of mixed semidefinite quadratic linear programs [12]. This set contains four MAX-CUT problems, called the torus problems, which originated from the Ising model of spin glasses in physics (see section 5.3 for details). In Table 1, we give statistics for this set of problems; note that the sizes of the graphs are given as $(|V|, |E|)$. In the table, the columns "Lower bound" and "Upper bound" give the best lower and upper bounds on the maximum cut value known to us to date, and the column "SDP bound" gives the SDP upper bounds on the maximum cut values. All the lower and upper bounds were supplied to us by Michael Jünger and Frauke Liers [22] except for the lower bounds 2988 for `pm3-15-50` and 285790637 for `g3-15`, which were the best cut values obtained so far by CirCut on these two problems, respectively. We mention that for `pm3-8-50` and `g3-8`, the best cut values obtained so far by CirCut are, respectively, 454 and 41684814, and the latter value is optimal.

In Table 2, we present a comparison between the SBmethod and CirCut codes. Since the latest version of SBmethod does not include the functionality of generating cuts by the Goemans–Williamson randomized procedure, we used an earlier version that does. It is quite likely that the latest version of SBmethod would produce better timings than those presented in the table.

We ran both SBmethod and CirCut on an SGI Origin2000 machine with sixteen 300MHZ R12000 processors at Rice University. Since neither code is parallel, however, only one processor was used at a time. For both codes, the cut values were obtained without any postprocessing heuristics, i.e., the simple local search feature of CirCut was not invoked. The default parameter settings were used for SBmethod. In Table 2, the cut value and computation time are reported for each problem. For CirCut, the value of M is the number of times Algorithm 1 was run with random starting points, and the value of N is the parameter required by Algorithm 1. The average time per run, the average cut value, and the best value in the M runs are reported in the last three columns of the table, respectively. All the reported times are in seconds. From the table, it is clear that an average run of CirCut is much faster and produces better quality cuts on all four test problems.

More results are reported in Table 3 for CirCut using different values of N. These results indicate that the variations between the average and best cut values are quite moderate, and they also show that even with $N = 0$ (no further improvement at-

TABLE 3
More CirCut results on MAX-CUT problems from the torus set.

Graph	CirCut ($N = 0$, $M = 100$)			CirCut ($N = 8$, $M = 100$)		
Name	Avg. val.	Avg. time	Best val.	Avg. val.	Avg. time	Best val.
pm3-8-50	430	0.031	444	448	0.386	454
pm3-15-50	2792	0.212	2834	2937	4.272	2964
g3-8	37870328	0.024	40314704	40917332	0.538	41684814
g3-15	253522848	0.154	264732800	277864512	7.880	281029888

tempted after minimization), CirCut gives quite respectable cuts in a minimal amount of time on average. As N increases, CirCut produces better quality cuts and of course uses more time. However, even for $N = 8$, CirCut is still faster by orders of magnitude.

We should bear in mind that in every run SBmethod also produces an upper bound; hence the running times for CirCut and SBmethod are not exactly comparable. They become totally comparable only when the sole objective of the computation is to obtain approximate solutions. These comments also apply to the comparisons presented in the next subsection and in section 6.

5.2. Comparison with DSDP. The second set of test problems are from the so-called G-set graphs, which are randomly generated. Recently, Choi and Ye [9] reported computational results on a subset of G-set graphs that were solved as MAX-CUT problems using their SDP code COPL-DSDP, or simply DSDP. The code DSDP uses a dual-scaling interior-point algorithm and an iterative linear-equation solver. It is currently one of the fastest interior-point codes for solving SDP problems.

We ran CirCut on a subset of G-set graphs as MAX-CUT problems and compared our results with those reported in Choi and Ye [9]. The comparison is given in Table 4, along with graph name and size information. We emphasize that the timing for DSDP was obtained on an HP 9000/785/C3600 machine with a 367 MHZ processor [8], while ours was on the aforementioned SGI Origin2000 machine at Rice University. These two machines seem to have comparable processing speeds. We did not run DSDP on the same computer at Rice University for several reasons: (1) the latest version of DSDP with an iterative linear-equation solver has not yet been made publicly available, (2) since the speeds of DSDP and CirCut are orders of magnitude apart, a precise timing is unnecessary in a qualitative comparison, and (3) it would be excessively time-consuming to rerun DSDP on all the tested problems (as can be see from Table 4).

The first two columns of Table 4 contain information concerning the tested graphs, where the sizes are again given as $(|V|, |E|)$, followed by timing (in seconds) and cut value information. The DSDP results were given as reported in [9]. We ran CirCut using two sets of parameters: "C1" results were for $N = 0$ and $M = 1$ (no further improvement after minimization and a single starting point), and "C2" for $N = 10$ and $M = 5$. Note that in this table the running times listed for C2 include all $M = 5$ runs; i.e., the times are not averaged as in the previous tables.

We observe that C1 took less than 11 seconds to return approximate solutions to all the 27 test problems with a quality that, on average, is nearly as good as that of the DSDP cuts, which required more than 5 days of computation. On the other hand, C2 took more time to generate the cuts, but the quality of the C2 cuts is almost uniformly better than those of DSDP, with one exception. Only on problem G50 did DSDP produce a slightly better cut. We note, however, that CirCut can easily find a cut of the same value on G50 if M is set to a larger value.

TABLE 4
Comparison with DSDP on MAX-CUT problems from the G-set.

Graph		Time			Value		
Name	Size	DSDP	C1	C2	DSDP	C1	C2
G11	(800, 1600)	16.6	0.06	3.88	542	524	554
G12	(800, 1600)	17.7	0.06	3.76	540	512	552
G13	(800, 1600)	18.2	0.06	3.45	564	536	572
G14	(800, 4694)	35.2	0.09	5.53	2922	3016	3053
G15	(800, 4661)	32.1	0.09	5.91	2938	3011	3039
G20	(800, 4672)	32.0	0.11	5.56	838	901	939
G21	(800, 4667)	37.6	0.08	5.56	841	887	921
G22	(2000, 19990)	4123.3	0.36	22.31	12960	13148	13331
G23	(2000, 19990)	3233.5	0.37	18.85	13006	13197	13269
G24	(2000, 19990)	3250.7	0.30	27.30	12933	13195	13287
G30	(2000, 19990)	3718.9	0.32	23.77	3038	3234	3377
G31	(2000, 19990)	3835.7	0.33	19.61	2851	3146	3255
G32	(2000, 4000)	142.6	0.18	13.09	1338	1306	1380
G33	(2000, 4000)	132.5	0.14	12.62	1330	1290	1352
G34	(2000, 4000)	156.7	0.12	9.82	1334	1276	1358
G50	(3000, 6000)	264.6	0.17	15.71	5880	5748	5856
G55	(5000, 12498)	1474.8	0.54	39.73	9960	10000	10240
G56	(5000, 12498)	15618.6	0.46	33.52	3634	3757	3943
G57	(5000, 10000)	1819.8	0.48	32.23	3320	3202	3412
G60	(7000, 17148)	58535.1	0.73	56.75	13610	13765	14081
G61	(7000, 17148)	52719.6	0.51	63.57	5252	5429	5690
G62	(7000, 14000)	5187.2	0.47	47.04	4612	4486	4740
G64	(7000, 41459)	102163.9	0.94	67.56	7624	8216	8575
G70	(10000, 9999)	33116.2	0.37	94.39	9456	9280	9529
G72	(10000, 20000)	12838.1	0.72	86.59	6644	6444	6820
G77	(14000, 28000)	32643.4	0.95	109.41	9418	9108	9670
G81	(20000, 40000)	131778.2	1.49	140.46	13448	12830	13662

5.3. Comparison with a heuristic algorithm from physics. An area of great interest in modern physics is the study of spin glasses [3, 14], and the particular problem of computing the so-called groundstate of an Ising spin glass can be cast as the problem of finding a maximum cut in a edge-weighted graph. In this section, we compare our heuristic CirCut with a successful heuristic by Hartmann [19] for finding an approximation to the groundstate of specially structured spin glasses.

Roughly speaking, a spin glass is a collection of n magnetic spins that possesses various interactions between the spins and also exhibits disorder in its frozen, or low-energy, state. In the collection, each spin can take on one of a finite number of positions. For example, when there are exactly two possible positions, the two positions are imagined as "up" and "down" (or $+1$ and -1). In addition, the interactions between the spins describe how the positions of a given spin and its "neighbor" spins affect the overall energy of the spin glass. For example, in Table 5 we show the energy contributed by two interacting spins i and j for a spin glass in which (i) there are two possible positions for a spin, (ii) all interactions act pairwise between spins, and (iii) each interaction is either positive or negative.

The groundstate, or low-energy state, of a spin glass occurs when the positions of the n spins are chosen so as to minimize the overall energy of the spin glass. Additionally, spin glasses are characterized by the fact that their groundstate is disordered; that is, all interactions cannot be satisfied with zero energy, and hence the overall energy of the system is positive. (Note that the standard physics terminology differs somewhat from—but is equivalent to—our terminology.)

TABLE 5
Energy levels of two interacting spins.

i	j	Interaction	Energy
up	up	+	0
up	down	+	1
down	up	+	1
down	down	+	0
up	up	−	1
up	down	−	0
down	up	−	0
down	down	−	1

A special subclass of spin glasses, called the Ising spin glasses, has been studied extensively. Ising spin glasses satisfy items (i) and (ii) of the previous paragraph, and the so-called $\pm J$ model of Ising spin glasses also satisfies item (iii). It is not difficult to see that this model can be represented by an edge-weighted graph $\bar{G} = (V, E, \bar{W})$, where the vertex set V consists of the n spins, the edge set E describes the pairwise interactions, and the symmetric weight matrix $\bar{W} = (\bar{w}_{ij})$ has $\bar{w}_{ij}$ equal to 1, -1, or 0, respectively, if i and j interact positively, negatively, or not at all. Moreover, if a variable x_i that can take on values $+1$ or -1 is used to represent the position of spin i, then the groundstate of the Ising spin glass can be seen to be the optimal solution of the optimization

$$\begin{aligned} \min \quad & \sum_{(i,j)\in E} \frac{1}{2}(1 - \bar{w}_{ij} x_i x_j) \\ \text{s.t.} \quad & |x_i| = 1, \quad i = 1, \ldots, n. \end{aligned} \tag{16}$$

After some immediate simplifications, (16) can be written in the equivalent form (2), where $w_{ij} = -\bar{w}_{ij}$, that is, (16) is equivalent to the maximum cut problem on the graph $G = (V, E, W)$, where $W = -\bar{W}$.

Many approaches for solving (16) have been investigated in both the physics community and the optimization community (see [2, 28]). Recently, one of the most successful heuristic approaches for solving (16) has been the approach of Hartmann [19], which in particular focuses on finding the groundstates of $\pm J$ Ising spin glasses that can be embedded as square or cubic lattices in two or three dimensions, respectively. The interactions are of the type "nearest neighbor" so that each vertex (or spin) has four neighbors in two dimensions and six in three dimensions. Such lattice graphs lead to regular graphs having a great deal of structure. In addition, Hartmann considers cases in which negative interactions occur as many times as positive interactions, that is, $\sum_{(i,j)\in E} \bar{w}_{ij} = 0$. Hartmann reported strong computational results with square lattices having side length $L = 4, 5, \ldots, 30$ and cubic lattices having length $L = 4, 5, \ldots, 14$. Note that the square lattices have a total of L^2 vertices and that the cubic lattices have a total of L^3 vertices.

Although we refer the reader to [19] for a full description of Hartmann's algorithm, we summarize the basic idea of the method here. Given a feasible solution x to (16), the algorithm tries to find a new feasible solution $\hat{x}$ having less energy by using x to randomly build up a set of nodes $\hat{V}$ for which the groundstate $x_{\hat{V}}$ of the induced graph on $\hat{V}$ can be found in polynomial time using a max-flow min-cut algorithm. Then $\hat{x}$ is formed from x by setting $\hat{x}_i = (x_{\hat{V}})_i$ if $i \in \hat{V}$ and $\hat{x}_i = x_i$ if $i \notin V$. The energy of $\hat{x}$ is guaranteed to be no worse than that of x, and so this procedure can be iterated

TABLE 6
Comparison of CirCut and Hartmann's algorithm.

Graph			Cut values				Times			
#	$\|V\|$	$\|E\|$	C1	C2	H1	H2	C1	C2	H1	H2
1	1000	3000	874	880	882	896	5	39	69	9528
2	1000	3000	894	892	892	900	7	47	68	9605
3	1000	3000	878	882	878	892	6	45	68	9537
4	1000	3000	888	894	890	898	7	54	68	9583
5	1000	3000	878	880	876	886	6	48	69	9551
6	1000	3000	866	876	874	888	6	47	68	9555
7	1000	3000	882	894	890	900	8	57	69	9564
8	1000	3000	872	874	870	882	7	53	69	9629
9	1000	3000	884	896	888	902	6	48	68	9551
10	1000	3000	876	888	884	894	5	56	69	9629
11	2744	8232	2396	2410	2382	2446	22	219	236	33049
12	2744	8232	2398	2426	2390	2458	20	170	236	32836
13	2744	8232	2382	2404	2370	2442	20	165	235	33171
14	2744	8232	2398	2418	2394	2450	19	173	236	33136
15	2744	8232	2382	2412	2370	2446	20	177	235	32851
16	2744	8232	2404	2416	2384	2450	23	183	236	33129
17	2744	8232	2390	2406	2384	2444	19	166	234	32999
18	2744	8232	2412	2414	2386	2446	28	171	236	33089
19	2744	8232	2382	2390	2356	2424	31	187	235	32963
20	2744	8232	2410	2422	2388	2458	19	166	236	33140

until the energy exhibits no strict improvement from iteration to iteration. Various parameters of the algorithm can affect its running time and also the quality of solution that is returned; these parameters determine the number of iterations allowed with no improvement, the number of independent times the overall algorithm is run, and, more generally, the exhaustiveness of the search performed by the algorithm.

We ran both CirCut and the algorithm of Hartmann on the same SGI Origin 2000 used for the computational results in the previous subsections. Hartmann's code is written in ANSI C and uses only one processor. In addition, we compiled both codes with the same compiler optimization option. In Table 6, we compare CirCut with the algorithm of Hartmann on twenty graphs arising from twenty cubic lattices having randomly generated interaction magnitudes; these problems are of the same type that Hartmann investigated in [19]. Ten of the graphs have $(L, n, |E|) = (10, 1000, 3000)$, and ten have $(L, n, |E|) = (14, 2744, 8232)$. We note that, for comparison purposes, the output of each algorithm is in terms of the equivalent maximum cut problem. Two versions of CirCut corresponding to the parameter choices $(N, M) = (10, 5)$ and $(N, M) = (50, 10)$ were run on all thirty graphs; the versions are named C1 and C2, respectively. Similarly, two versions H1 and H2 of Hartmann's algorithm were run such that H1 performed a less exhaustive search than H2. We remark that H2 represented the default parameters supplied to us by Hartmann.

Table 6 contains data corresponding to the four algorithms' performance on each of the twenty graphs. The first three columns give the graph number, the size of V, and the size of E. The next four columns give the cut value found by the algorithms, and the final four columns give the times (in seconds) required by each of the algorithms.

It can be seen from the table that on the first ten graphs, C1 had the fastest speed, but the cuts it returned were in a few cases inferior to those produced by H1. On the other hand, C2 was able to produce better cuts than H1 in a considerably shorter amount of time. The overall winner in terms of cut values on graphs 1–10 was H2, but this performance was achieved at the expense of very large computation

times. For the second set of graphs 11–20, we see that both C1 and C2 outperformed H1 in terms of cut values and that C1 was much faster than H1 and C2 was notably faster than H1 as well. Again, H2 returned the best cuts but took a very long time. In all cases, the differences in the quality of cuts generated by the algorithms are small, percentage-wise. For example, on average C1 attained over 98 percent of the cut value of H2 in an amount of time less than one-tenth of a percent of that used by H2.

Overall, the results seem to indicate that C2 is a good choice when quality cuts are needed in a short amount of time. In particular, C2 is at least as effective as H1. In addition, C1 is a good alternative, especially when the size of the graph becomes large. When high quality cuts are needed and time is not an issue, H2 is the best choice. Moreover, we remark that, based on some unreported experimentation, CirCut does not seem to be able to achieve the same cut values as H2 even if CirCut is allowed to search for a very long time.

6. Some extensions. Conceptually, there is little difficulty in extending the rank-two relaxation idea to other combinatorial optimization problems in the form of a binary quadratic program, especially to those arising from graph bipartitioning. For a given problem, however, whether or not the rank-two relaxation will lead to high-performance algorithms, like the one we have demonstrated for MAX-CUT, must be determined by an individual investigation and a careful evaluation. Close attention must also be paid to the specific structure of each problem in order to obtain good algorithms.

In this section, we focus on extending the rank-two relaxation idea to a close relative of MAX-CUT—the MAX-BISECTION problem. MAX-BISECTION is the same as MAX-CUT except that it has the additional constraint $e^T x = 0$ (i.e., the number of positive ones in x must equal the number of negative ones, hence implying that n should be even), which can also be written as

$$(e^T x)^2 = (ee^T) \bullet (xx^T) = 0.$$

After removal of the rank-one restriction, one obtains the following SDP relaxation of the MAX-BISECTION problem (comparable to (5)):

$$\begin{array}{ll} \min & \frac{1}{2} W \bullet X \\ \text{s.t.} & \operatorname{diag}(X) = e, \\ & ee^T \bullet X = 0, \\ & X \succeq 0. \end{array} \tag{17}$$

Randomized procedures similar to the Goemans–Williamson technique for MAX-CUT have been proposed with different performance guarantees for MAX-BISECTION; see [15, 32], for example.

In an approach analogous to that used for MAX-CUT, using the rank-two relaxation and polar coordinates, we obtain a new relaxation for MAX-BISECTION:

$$\begin{array}{ll} \min & f(\theta) \\ \text{s.t.} & e^T \cos(T(\theta)) e = 0. \end{array} \tag{18}$$

Suppose that we have obtained a (local or global) minimizer θ for (18). How do we generate a bisection? Without loss of generality, let us assume that n is even and that θ satisfies $\theta_i \in [0, 2\pi),\ i = 1, \ldots, n$. We may also assume that, after a reordering,

$$\theta_1 \leq \theta_2 \leq \cdots \leq \theta_n.$$

Then, to generate a bisection, we pick any integer $k \in [1, n/2)$ and let

$$x_i = \begin{cases} 1 & \text{if } i \in [k, k+n/2), \\ -1 & \text{otherwise.} \end{cases} \tag{19}$$

The following procedure efficiently considers all possible values of k in (19) and saves the best resultant bisection.

PROCEDURE-BIS (input θ, output x^*).

Given $\theta \in \Re^n$ such that $0 \le \theta_1 \le \cdots \le \theta_n < 2\pi$, let $\Gamma = -\infty$.

For $k = 1, \ldots, n/2 - 1$

1. Generate a cut x by (19) and compute $\gamma(x)$.
2. If $\gamma(x) > \Gamma$, then let $\Gamma = \gamma(x)$ and $x^* = x$.

End

Instead of solving the constrained relaxation (18), we have found through numerical experiments that solving the unconstrained relaxation (7) can generate the same or better quality bisections while taking less time. Intuitively, this is not hard to understand since the best bisection generated by Procedure-BIS for a given θ is dependent only on the ordering of the points along the circle and independent of the actual locations of the points. In fact, it is easy to verify that the constraint in (18) is equivalent to $\|[v_1 \ \cdots \ v_n]e\|^2 = 0$, where $v_i = [\cos(\theta_i) \ \sin(\theta_i)]^T$; that is, the n vectors on the unit circle must sum up to zero. So by itself, the constraint puts a restriction on the locations of points but has nothing to do with their ordering. Hence, whether a given θ satisfies the constraint or not has no bearing on the quality of the bisection x^* generated by Procedure-BIS. On the other hand, the quality of x^* depends greatly on the objective value $f(\theta)$. Since it is more likely to obtain lower function values at unconstrained local minima than at constrained ones, we are more likely to obtain better bisections without the constraint.

In view of this, we construct our heuristic algorithm based on minimizing $f(\theta)$ without the additional constraint. We simply replace Procedure-CUT in Algorithm-1 by Procedure-BIS and obtain a heuristic algorithm for the MAX-BISECTION problem, which we call Algorithm 2. In Algorithm 2, we also have the option of improving a cut by a minimal local search that allows swapping only a pair of nodes at a time and is set to be a default feature.

We ran Algorithm-2 of CirCut on a subset of the G-set problems plus two additional test problems. These extra problems were contained in a test set used by Choi and Ye [9] and are publicly available.

In Table 7, we compare the results of CirCut with the results of DSDP reported in [9]. Again, we mention that the timing for DSDP was obtained on an HP 9000/785/C3600 computer with a 367 MHZ processor, while ours was on an SGI Origin2000 machine with sixteen 300 MHZ processors at Rice University. (Note, however, that both codes always use a single processor.)

Again, the first two columns of Table 7 contain the information on the tested graphs, followed by timing (in seconds) and cut value information. We ran CirCut using two sets of parameters: C1 results were for $N = 0$ and $M = 1$ (no further improvement after minimization and a single starting point); and C2 for $N = 5$ and $M = 1$.

C1 took less than 22 seconds to return approximate solutions to all 13 test problems with a quality that is on average superior to that of DSDP. While C2 took more time to generate the bisections, the quality of the bisections generated by C2 is better than that of DSDP on all but one problem: G50. Again, we mention that if N and

TABLE 7
Comparison with DSDP on MAX-BISECTION problems.

Graph		Time			Value		
Name	Size	DSDP	C1	C2	DSDP	C1	C2
G50	(3000, 6000)	462.2	0.29	2.29	5878	5690	5830
G55	(5000,12498)	1793.4	0.46	4.32	9958	10007	10171
G56	(5000,12498)	20793.5	0.44	3.36	3611	3672	3835
G57	(5000,10000)	2090.8	0.32	2.98	3322	3146	3382
G60	(7000,17148)	48949.9	0.54	4.66	13640	13759	13945
G61	(7000,17148)	42467.2	0.62	7.16	5195	5312	5545
G62	(7000,14000)	5446.0	0.50	4.98	4576	4402	4706
G64	(7000,41459)	123409.7	0.92	12.05	7700	8056	8431
G72	(10000,20000)	15383.9	0.76	7.34	6628	6314	6736
G77	(14000,28000)	36446.7	1.15	11.38	6560	8980	9638
G81	(20000,40000)	334824.2	1.54	26.87	9450	12582	13618
bm1	(882,4711)	33.9	0.08	0.65	848	857	863
biomedp	(6514,629839)	46750.7	13.89	37.55	5355	5575	5593

M are set to larger values, CirCut is able to produce a bisection of the same value on G50 as that of DSDP's, within a time still much shorter than that required by DSDP.

6.1. Maximization versus minimization. So far, we have presented only computational results on maximization problems, i.e., the MAX-CUT and MAX-BISECTION problems, which are equivalent to minimizing $f(\theta)$. Moreover, all of the graphs in the test sets have had either all positive edge weights or a combination of both positive and negative weights.

Now let us consider the corresponding minimization problems on these graphs, equivalent to maximizing $f(\theta)$. For those graphs having both positive and negative weights, one can apply the same algorithms to the minimization problems by simply minimizing $-f(\theta)$ instead of $f(\theta)$. Things are not so simple, however, if all the weights are positive. In this case, it is easy to see that the global minimum of $-f(\theta)$ is attained whenever all n points coincide on the unit circle such that $\cos(\theta_i - \theta_j) \equiv 1$. This result makes sense for the MIN-CUT problem in that the minimum cut in a graph with all positive weights is to have all nodes on one side of the cut (i.e., to have no cut at all). On the other hand, this result does not have a meaningful interpretation for MIN-BISECTION, creating a challenge for generating a bisection whenever a global minimum of $-f(\theta)$ is attained (although actually finding a global minimum may not happen often). An obvious possible remedy to this problem is to reinstall the bisection constraint back into the formulation. Further investigation is clearly needed for the MIN-BISECTION problem.

7. Concluding remarks. The computational results presented here indicate that the proposed rank-two relaxation heuristics are effective in approximating the MAX-CUT and MAX-BISECTION problems. Being able to return high-quality approximate solutions in a short amount of time, they are particularly useful in situations where either the problem is very large or time is at a premium.

Several factors have contributed to the performance of the rank-two relaxation approach: (1) the costs of local optimization are extremely low; (2) desirable properties relate the discrete problem to its rank-two relaxation, enabling us to locate high-quality local minima; and (3) good local minima of the rank-two relaxation appear to be sufficient for generating good approximate solutions to the discrete problem.

The proposed heuristics consistently produce better-quality approximate solutions

while taking only a tiny amount of time in comparison to the SDP relaxation approach, particularly on larger problems. This fact suggests that as a practical technique for producing lower bounds, the SDP relaxation approach does not seem to hold much promise, at least for the MAX-CUT and the MAX-BISECTION problems. In addition, the rank-two relaxation heuristic compares favorably to other heuristics, i.e., ones that are not based on the SDP relaxation.

It is known that, besides MAX-CUT, a number of other combinatorial optimization problems can also be formulated as unconstrained binary quadratic programs in the form of (2), such as the MAX-CLIQUE problem (see [4], for example). These are potential candidates for which the rank-two relaxation approach may also produce high-performance heuristic algorithms. Further investigation in this direction will be worthwhile.

Acknowledgments. We are grateful to an anonymous referee for valuable comments and suggestions that have helped us to improve the paper. We would also like to thank Alexander Hartmann for sharing his computer code with us and Professor Franz Rendl for his careful reading of the paper and insightful comments.

REFERENCES

[1] S. J. Benson, Y. Ye, and X. Zhang, *Solving large-scale sparse semidefinite programs for combinatorial optimization*, SIAM J. Optim., 10 (2000), pp. 443–461.

[2] B. A. Berg and T. Celik, *New approach to spin-glass simulations*, Phys. Rev. Lett., 69 (1992), pp. 2292–2295.

[3] K. Binder and A. P. Young, *Spin-glasses—experimental facts, theoretical concepts and open questions*, Rev. Modern Phys., 58 (1986), pp. 801–977.

[4] I. Bomze, M. Budinich, P. Pardalos, and M. Pelillo, *The maximum clique problem*, in Handbook of Combinatorial Optimization, Vol. 4, D.-Z. Du and P. M. Pardalos, eds., Kluwer Academic Publishers, Boston, MA, 1999, pp. 1–74.

[5] S. Burer and R. D. C. Monteiro, *A projected gradient algorithm for solving the Maxcut SDP relaxation*, Optim. Methods Softw., to appear.

[6] S. Burer, R. D. C. Monteiro, and Y. Zhang, *Solving a Class of Semidefinite Programs via Nonlinear Programming*, Tech. report TR99-17, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1999.

[7] S. Burer, R. D. C. Monteiro, and Y. Zhang, *Interior-Point Algorithms for Semidefinite Programming Based on a Nonlinear Programming Formulation*, Tech. report TR99-27, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1999.

[8] C. Choi, *private communication*, University of Iowa, Iowa City, IA, 2000.

[9] C. Choi and Y. Ye, *Solving Sparse Semidefinite Programs Using the Dual Scaling Algorithm with an Iterative Solver*, working paper, Department of Management Science, University of Iowa, IA, 2000.

[10] C. Delorme and S. Poljak, *Laplacian eigenvalues and the maximum cut problem*, Math. Programming, 62 (1993), pp. 557–574.

[11] O. Dolezal, T. Hofmeister, and H. Lefmann, *A Comparison of Approximation Algorithms for the Maxcut Problem*, manuscript, Universität Dortmund, Lehrstuhl Informatik 2, Dortmund, Germany, 1999.

[12] *The DIMACS Library of Mixed Semidefinite-Quadratic-Linear Programs,* http://dimacs.rutgers.edu/Challenges/Seventh/Instances/.

[13] U. Feige and G. Schechtman, *On the Optimality of the Random Hyperplane Rounding Technique for MAX CUT*, manuscript, Faculty of Mathematics and Computer Science, Weizmann Institute, Rehovot, Israel, 2000.

[14] K. H. Fisher and J. A. Hertz, *Spin Glasses*, Cambridge University Press, Cambridge, UK, 1991.

[15] A. Frieze and M. Jerrum, *Improved algorithms for Max K-cut and Max bisection*, Algorithmica, 18 (1997), pp. 67–81.

[16] K. Fujisawa, M. Kojima, and K. Nakata, *Exploiting sparsity in primal-dual interior-point methods for semidefinite programming*, Math. Programming, 79 (1997), pp. 235–253.

[17] M. Fukuda, M. Kojima, K. Murota, and K. Nakata, *Exploiting sparsity in semidefinite programming via matrix completion* I: *General framework*, SIAM J. Optim., 11 (2001), pp. 647–674.
[18] M. X. Goemans and D. P. Williamson, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, J. ACM, 42 (1995), pp. 1115–1145.
[19] A. K. Hartmann, *Cluster-exact approximation of spin glass groundstates*, Phys. A, 224 (1996), pp. 480–488.
[20] C. Helmberg and F. Rendl, *A spectral bundle method for semidefinite programming*, SIAM J. Optim., 10 (2000), pp. 673–696.
[21] M. Peinado and S. Homer, *Design and performance of parallel and distributed approximation algorithms for maxcut*, J. Parallel and Distributed Comput., 46 (1997), pp. 48–61.
[22] M. Jünger and F. Liers, *private communication,* Universität Köln, Cologne, Germany, 2000.
[23] L. Lovász, *On the Shannon capacity of a graph*, IEEE Trans. Inform. Theory, 25 (1979), pp. 1–7.
[24] L. Lovász and A. Schrijver, *Cones of matrices and set-functions and* 0-1 *optimization*, SIAM J. Optim., 1 (1991), pp. 166–190.
[25] Y. E. Nesterov and A. S. Nemirovskii, *Interior-Point Polynomial Algorithms, in Convex Programming,* SIAM, Philadelphia, 1994.
[26] S. Poljak, and F. Rendl, *Nonpolyhedral relaxations of graph-bisection problems*, SIAM J. Optim., 5 (1995), pp. 467–487.
[27] N. Z. Shor, *Quadratic optimization problems*, Soviet J. Comput. Systems Sci., 25 (1987), pp. 1–11; Tekhnicheskaya Kibernetika, 1 (1987), pp. 128–139 (in Russian).
[28] C. De Simone, M. Diehl, M. Jünger, P. Mutzel, G. Reinelt, and G. Rinaldi, *Exact ground states of Ising spin glasses: New experimental results with a branch and cut algorithm*, J. Statist. Phys., 80 (1995), pp. 487–496.
[29] K.-C. Toh and M. Kojima, *Solving Some Large Scale Semidefinite Programs via the Conjugate Residual Method*, research report, Department of Mathematics, National University of Singapore, Singapore, 2000.
[30] R. J. Vanderbei and H. Yurttan Benson, *On Formulating Semidefinite Programming Problems as Smooth Convex Nonlinear Optimization Problems*, Techn. report ORFE 99-01, Department of Operations Research and Financial Engineering, Princeton, University, Princeton NJ, 1999.
[31] S. Vavasis, *A Note on Efficient Computation of the Gradient in Semidefinite Programming*, working paper, Department of Computer Science, Cornell University, Ithaca, NY, 1999.
[32] Y. Ye, *A* .699*-approximation algorithm for max-bisection*, Math. Program., 90 (2001), pp. 101–111.

SIAM J. OPTIM.
Vol. 12, No. 2, pp. 522–554

A PRIMAL-DUAL ALGORITHM FOR SOLVING POLYHEDRAL CONIC SYSTEMS WITH A FINITE-PRECISION MACHINE*

FELIPE CUCKER† AND JAVIER PEÑA‡

Abstract. We describe a primal-dual interior-point algorithm that determines which one of two alternative systems,

$$Ax = 0, \quad x \geq 0,$$

and

$$A^{\mathrm{T}}y \leq 0,$$

is strictly feasible, provided that this pair of systems is well-posed. Furthermore, when the second system is strictly feasible, the algorithm returns a strict solution y; when the first system is strictly feasible, the algorithm returns a strict forward-approximate solution x. Here $A \in \mathbb{R}^{m\times n}$ is given. Our algorithm works with finite-precision arithmetic. The amount of precision required is adjusted as the algorithm progresses and remains bounded by a measure of well-posedness $C(A)$ of the pair of systems of constraints. The algorithm halts in at most $\mathcal{O}((m+n)^{1/2}(\log(m+n)+\log(C(A))+|\log\gamma|))$ interior-point iterations, where $\gamma \in (0,1)$ is a parameter specifying the desired degree of accuracy of the forward-approximate solution for the first system. If the feasible system is the second one, the term $|\log\gamma|$ in the bound on the number of iterations can be dropped.

Key words. linear conic systems, finite-precision algorithms

AMS subject classifications. 90C05, 90C51

PII. S1052623401386794

1. Introduction.

1.1. Let $A \in \mathbb{R}^{m\times n}$ be given and consider the two systems

$$Ax = 0, \quad x \geq 0, \tag{1.1}$$

and

$$A^{\mathrm{T}}y \leq 0. \tag{1.2}$$

It is well known that one of these systems has a strict solution (one for which the satisfied inequality is strict) if and only if the other has no nontrivial solutions. (A solution to (1.2) is nontrivial if it satisfies $A^{\mathrm{T}}y \neq 0$.) This is a generic property. Indeed, except for a set of Lebesgue measure zero, for any $A \in \mathbb{R}^{m\times n}$ the pair (1.1)–(1.2) is *well-posed*: one of the systems has a strict solution, and the same system continues to have a strict solution even if the matrix A is slightly perturbed. Formally speaking, the pair (1.1)–(1.2) is well-posed if the distance from A to ill-posedness (defined below) is strictly positive.

*Received by the editors March 26, 2001; accepted for publication (in revised form) July 13, 2001; published electronically December 14, 2001.
http://www.siam.org/journals/siopt/12-2/38679.html

†Department of Mathematics, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong, P. R. of CHINA (macucker@math.cityu.edu.hk). This author's work was partially supported by CERG grant 9040393.

‡Graduate School of Industrial Administration, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213-3890 (jfp@andrew.cmu.edu). This author's work was partially supported by NSF grant CCR-0092655. Most of this paper was written while Javier Peña was visiting City University of Hong Kong in Summer, 1999, and Fall, 2000.

We propose an algorithm that, for well-posed pairs, will decide which of (1.1) and (1.2) has a strict solution and produce such a solution. A key feature of our algorithm is that we do not assume infinite precision for real number arithmetic. These kinds of computations are to be performed with finite precision. The machine precision, though, will vary during the computation. It is initially small and subsequently needs to be gradually sharpened, but remains bounded and is not much larger than the minimal precision that any algorithm would require.

The assumption of finite precision sets some limitations on the kinds of results we may obtain. If system (1.2) has strict solutions, then we will obtain, after sufficiently refining the precision, a strict solution $y \in \mathbb{R}^m$ of (1.2). On the other hand, if the system having a strict solution is (1.1), then there is no hope of exactly computing one such solution x, since the set of solutions is thin in $\mathbb{R}^n$ (i.e., has empty interior). In such a case there is no way to ensure that the errors produced by the use of finite precision will not move any candidate solution out of this set. We can, however, compute good approximations, namely, forward-approximate solutions. The following notion is partly inspired by the discussion of forward solutions in [15, section 5].

DEFINITION 1.1. *Let $\gamma \in (0,1)$. A point $\widehat{x} \in \mathbb{R}^n$ is a* γ-forward solution *of the system $Ax = 0$, $x \geq 0$, if $\widehat{x} \geq 0$, $\widehat{x} \neq 0$, there exists $\overline{x} \in \mathbb{R}^n$ such that*

$$A\overline{x} = 0, \quad \overline{x} \geq 0,$$

and, for $i = 1, \ldots, n$,

$$|\widehat{x}_i - \overline{x}_i| \leq \gamma \widehat{x}_i.$$

The point $\overline{x}$ is said to be an associated solution *for $\widehat{x}$. A point is a* forward-approximate solution *of $Ax = 0$, $x \geq 0$, if it is a γ-forward solution of the system for some $\gamma \in (0,1)$.*

In case system (1.1) has a strict solution, our algorithm will find a forward-approximate solution. Actually, if the desired accuracy γ of this approximation is given to the algorithm, the returned solution will be a γ-forward solution.

A central theme in numerical analysis (especially in numerical linear algebra) is the dependence of both the precision and the running time required by an algorithm to perform a computation on the condition of its input (measured by a positive real called the condition number). The results of our paper follow this theme. A main role is played by the condition number for linear programs introduced by Renegar (cf. [16]), which we now recall. Let $\rho_P(A)$ and $\rho_D(A)$ be the distances to infeasibility of (1.1) and (1.2), defined by

$$\rho_P(A) = \inf\{\|\Delta A\|_{1,\infty} : (A + \Delta A)x = 0,\ x \geq 0,\ x \neq 0 \text{ is infeasible}\}$$

and

$$\rho_D(A) = \inf\{\|\Delta A\|_{1,\infty} : (A + \Delta A)^{\mathrm{T}} y \leq 0,\ (A + \Delta A)^{\mathrm{T}} y \neq 0 \text{ is infeasible}\}.$$

Here $\|\cdot\|_{1,\infty}$ denotes the operator norm[1]

$$\|A\|_{1,\infty} := \sup\{\|Ax\|_1 : \|x\|_\infty \leq 1\}.$$

[1]We use this norm to make distances to infeasibility compatible with the norms in our formulation. This choice of norm is not critical, but different norms would complicate most expressions in our results with additional constants.

Note that only one of $\rho_D(A)$ and $\rho_P(A)$ can be positive. We say that A is *ill-posed* when both of them are zero. In this case either system can be made without nontrivial solutions by taking arbitrarily small perturbations on A. When this occurs, we do not expect our algorithm to yield any solution. Indeed, if A is ill-posed, then the algorithm will not halt. If we define

$$\rho(A) = \rho_P(A) + \rho_D(A) = \max\{\rho_P(A), \rho_D(A)\},$$

then it is easy to see that $\rho(A)$ is the *distance to ill-posedness*, i.e.,

$$\rho(A) = \inf\{\|\Delta A\|_{1,\infty} : A + \Delta A \text{ is ill-posed}\}.$$

It is also easy to see that if $\rho_P(A) > 0$, then (1.1) has strict feasible solutions. Likewise, if $\rho_D(A) > 0$, then (1.2) has strict feasible solutions. Furthermore, generically one of these conditions always holds, as it can be shown that the set of matrices A such that $\rho(A) = 0$ has Lebesgue measure zero.

Remark 1.1. The implications above can be rephrased as

$$\{A \in \mathbb{R}^{m\times n} : \rho_P(A) > 0\} \subseteq \{A \in \mathbb{R}^{m\times n} : Ax = 0,\ x > 0 \text{ is feasible}\}$$

and

$$\{A \in \mathbb{R}^{m\times n} : \rho_D(A) > 0\} \subseteq \{A \in \mathbb{R}^{m\times n} : A^{\mathrm{T}}y < 0 \text{ is feasible}\}.$$

It is not difficult to see that the second inclusion is actually an equality. The first one is strict though. For instance, for the matrix

$$\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$$

system (1.1) has strict solutions while $\rho_P(A) = 0$. Actually, it follows from [19] that

$$\{A : \rho_P(A) > 0\} = \{A : Ax = 0,\ x > 0 \text{ is feasible, and } A \text{ has full rank}\}.$$

We note, however, that the difference between the two sets in the first inclusion has measure zero, and the matrices in such difference are ill-posed.

The *condition number* $C(A)$ of the pair (1.1)–(1.2) is defined to be $C(A) = \|A\|_{1,\infty}/\rho(A)$ if $\rho(A) > 0$ and $C(A) = \infty$ otherwise. It satisfies $C(A) \geq 1$ and is invariant under positive scaling of A. For a detailed discussion on the distance to ill-posedness and condition numbers, see [16, 14].

The *round-off unit* or *machine precision* of a machine is a number $u \in \mathbb{R}$, $0 < u < 1$, such that real numbers x in the machine are systematically replaced by approximations $r(x)$ satisfying $|r(x) - x| \leq u|x|$. Roughly, $|\log u|$ corresponds with the number of digits of the mantissa in the floating-point representation of $r(x)$.

In our analysis we will estimate both the number of iterations of the algorithm and the precision required as functions of m, n, and $C(A)$. Our main result can be stated as follows.

THEOREM 1.2. *There exists a round-off machine which, with input of a matrix $A \in \mathbb{R}^{m\times n}$ and a number $\gamma \in (0, 1)$, finds either a strict γ-forward solution $x \in \mathbb{R}^n$ of $Ax = 0$, $x \geq 0$, or a strict solution $y \in \mathbb{R}^m$ of the system $A^{\mathrm{T}}y \leq 0$. The machine precision varies during the execution of the algorithm. The finest required precision is*

$$u = \frac{1}{\mathbf{c}(m+n)^{12}C(A)^2},$$

where **c** *is a universal constant. The number of main (interior-point) iterations of the algorithm is bounded by*

$$\mathcal{O}((m+n)^{1/2}(\log(m+n)+\log(C(A))+|\log\gamma|))$$

if $\rho_P(A)>0$*, and by the same expression without the* $|\log\gamma|$ *term if* $\rho_D(A)>0$.

The complexity bound in Theorem 1.2 cannot be written as a function of m and n solely, due to the unboundedness of $C(A)$. One can eliminate the occurrence of $\log C(A)$ in the bound above at the cost of trading worst-case for average-case complexity. In [5] it is shown that, for Gaussian matrices (i.e., matrices whose entries are independently and identically distributed (i.i.d.) normal random variables), the expected value of $\log C(A)$ is $\mathcal{O}(\min\{n, m\log n\})$ if $n>m$, and $\mathcal{O}(\log m)$ if $n\leq m$. Using this result, the following corollary follows.

COROLLARY 1.3. *For Gaussian* $m\times n$ *matrices, the expected number of main (interior-point) iterations of the algorithm in Theorem* 1.2 *is bounded by*

$$\begin{cases} \mathcal{O}\left((m+n)^{1/2}m(\log(m+n)+|\log\gamma|)\right) & \text{if } n>m,\\ \mathcal{O}\left((m+n)^{1/2}(\log(m+n)+|\log\gamma|)\right) & \text{if } n\leq m. \end{cases}$$

Although our results do not make any assumption about which of m and n is greater, the case $n>m$ is the interesting one. Not only is this the case that naturally arises in practice, but also it is the one in which both (1.1) and (1.2) may be strictly feasible. If $n\leq m$, then system (1.2) has strict solutions except when A is ill-posed. Hence the situation of $n\leq m$ is much simpler and relatively uninteresting.

Remark 1.2. At this stage some observations about complexity are necessary. Most of the work related to finite precision assumes that this precision is fixed. This implies a fixed cost for each arithmetic operation and therefore a total cost for the algorithm, which is, up to a constant, the number of arithmetic operations performed during the computation. This is the so-called *algebraic complexity* and is the measure underlying the complexity theory developed in [3]. In this fixed-precision context, every instance of algorithm analysis includes (or should include) a result bounding the accuracy of the solution as a function of the input size, the input condition, and the machine precision.

Theorem 1.2 does not belong to the context above, since the algorithm therein works with variable precision. This allows the algorithm (as long as $C(A)<\infty$) to return a true strict solution of (1.2), if (1.2) is strictly feasible, or a γ-forward solution of (1.1) for a prespecified γ, if (1.1) is strictly feasible. Needless to say, this is at the cost of increasing the precision. Thus, to be fair, one needs to associate some cost measure with this precision increase. At this point one notices that the fixed cost for each arithmetic operation is no longer a reasonable model for variable precision. A more realistic assumption assigns cost $(\log u)^2$ to any multiplication or division between two floating-point numbers with round-off unit u, since this is roughly the number of elementary operations performed by the computer to multiply or divide these numbers. For an addition, subtraction, or comparison the cost is $|\log u|$. The cost of the integer arithmetic necessary for computing variables' addresses and other quantities related with data management may be (and is customarily) ignored.

A closer look at our algorithm shows that at each iteration the algorithm performs $\mathcal{O}((m+n)^3)$ arithmetic operations. Therefore, the algebraic complexity of the algorithm is bounded by $\mathcal{O}((m+n)^{3.5}(\log(m+n)+\log C(A)+|\log\gamma|))$. Using the cost model described above we obtain a bound for the total cost of the algorithm of

$$\mathcal{O}((m+n)^{3.5}(\log(m+n)+\log C(A)+|\log\gamma|)^3).$$

Also, if we consider A to be Gaussian, then the expected cost is bounded by

$$\begin{cases} \mathcal{O}\left((m+n)^{3.5}m^3(\log(m+n)+|\log\gamma|)^3\right) & \text{if } n > m, \\ \mathcal{O}\left((m+n)^{3.5}(\log(m+n)+|\log\gamma|)^3\right) & \text{if } n \le m. \end{cases}$$

1.2. The effects of finite precision when solving linear programming problems have been noticed for many years. In [24], Wolfe suggested some strategies to control round-off errors for the simplex method. Other papers (e.g., [6, 20]) also dealt with the issue of round-off errors for the simplex method, but without providing rigorous results. A first formal treatment of this issue appears in [1], which shows some form of stabilization for the simplex method when replacing the Gauss–Jordan elimination without pivot selection by the Hessenberg-LU decomposition. Other papers related to round-off or inexact computations and linear programming include [19, 13].

The usual round-off analysis, however, as it is done in numerical linear algebra, has been scarce for linear programming (LP). This is only natural since it was not until very recently that condition numbers for LP problems were proposed (e.g., [16, 21, 4]). These condition numbers have been shown to control the size of solution sets and of particular solutions and speed of convergence of some iterative algorithms. They have been little used, however, for round-off analysis. A recent paper dealing with inexact computations, but not with round-off errors, in LP is [8]. As far as we are aware, the only round-off analysis for LP as described above is a recent paper by Vera [22].

Vera analyzes the computational complexity of using a logarithmic barrier method to solve

$$\begin{array}{ll} \min & c^{\mathrm{T}}x \\ \text{s.t.} & Ax \le b \end{array}$$

under finite-precision arithmetic. This is along the lines of our work. However, his work relies on the availability of the following information: (1) the condition number of the linear program (or an upper bound on it) and (2) an initial point $\overline{x}$ satisfying $A\overline{x} < b$. Vera's complexity bounds depend on the estimate of the condition number and the centrality of the initial point. In contrast, our algorithm requires no further knowledge of the problem beyond the input matrix A. The precision required is learned and adjusted as the algorithm progresses.

2. Main ideas.

2.1. Reformulating the problem. Our algorithm is based on a relaxation scheme introduced by Peña and Renegar [15]. The primal-dual perspective is partly motivated by Vavasis and Ye's formulation in [21].

We approach the feasibility problems $Ax = 0$, $x \ge 0$, and $A^{\mathrm{T}}y \le 0$ by studying the related pair of optimization problems

$$\begin{array}{ll} \min & \|\widetilde{x}\|_1 \\ \text{s.t.} & Ax + \widetilde{x} = 0, \\ & x \ge 0, \\ & \|x\|_\infty \le 1, \end{array} \tag{2.1}$$

and

$$\begin{array}{ll} \min & \|\widetilde{y}\|_1 \\ \text{s.t.} & A^{\mathrm{T}}y + \widetilde{y} \le 0, \\ & \widetilde{y} \le 0, \\ & \|y\|_\infty \le 1. \end{array} \tag{2.2}$$

We can recast these problems as the following primal-dual pair of linear programs:

$$\begin{aligned} \min \quad & e^{\mathrm{T}}x' + e^{\mathrm{T}}x'' \\ \text{s.t.} \quad & \begin{bmatrix} A & I_m & -I_m & \\ I_n & & & I_n \end{bmatrix} \begin{bmatrix} x \\ x' \\ x'' \\ x''' \end{bmatrix} = \begin{bmatrix} 0 \\ e \end{bmatrix}, \\ & x, x', x'', x''' \geq 0, \end{aligned} \tag{2.3}$$

and

$$\begin{aligned} \max \quad & e^{\mathrm{T}}y' \\ \text{s.t.} \quad & \begin{bmatrix} A^{\mathrm{T}} & I_n \\ I_m & \\ -I_m & \\ & I_n \end{bmatrix} \begin{bmatrix} y \\ y' \end{bmatrix} + \begin{bmatrix} s \\ s' \\ s'' \\ s''' \end{bmatrix} = \begin{bmatrix} 0 \\ e \\ e \\ 0 \end{bmatrix}, \\ & s, s', s'', s''' \geq 0. \end{aligned} \tag{2.4}$$

We shall apply a primal-dual interior-point method to the pair (2.3)–(2.4). A basic feature of interior-point methods is to generate iterates that are pushed away from the boundary of the feasible region. In addition, for the pair (2.3)–(2.4), it is obvious that at any optimal solution the variables x', x'', y' are all zero. Hence it is intuitively clear that an interior-point algorithm applied to (2.3)–(2.4) will yield a strict solution for either $Ax = 0$, $x \geq 0$, or $A^{\mathrm{T}}y \leq 0$, provided that the pair of systems is well-posed (i.e., $\rho(A) > 0$). Propositions 3.6 to 3.9 in section 3 formalize this statement.

In order to simplify our exposition, we will use the following notation:

$$\vec{x} = \begin{bmatrix} x \\ x' \\ x'' \\ x''' \end{bmatrix}, \quad \vec{s} = \begin{bmatrix} s \\ s' \\ s'' \\ s''' \end{bmatrix}, \quad \vec{y} = \begin{bmatrix} y \\ y' \end{bmatrix}$$

and

$$\mathcal{A} = \begin{bmatrix} A & I_m & -I_m & \\ I_n & & & I_n \end{bmatrix}, \quad \vec{b} = \begin{bmatrix} 0 \\ e \end{bmatrix}, \quad \vec{c} = \begin{bmatrix} 0 \\ e \\ e \\ 0 \end{bmatrix}.$$

Thus, we can write our primal-dual pair (2.3)–(2.4) in the more compact and standard LP form

$$\begin{aligned} \min \quad & \vec{c}^{\mathrm{T}}\vec{x} \\ \text{s.t.} \quad & \mathcal{A}\vec{x} = \vec{b}, \\ & \vec{x} \geq 0, \end{aligned} \tag{2.5}$$

and

$$\begin{aligned} \max \quad & \vec{b}^{\mathrm{T}}\vec{y} \\ \text{s.t.} \quad & \mathcal{A}^{\mathrm{T}}\vec{y} + \vec{s} = \vec{c}, \\ & \vec{s} \geq 0. \end{aligned} \tag{2.6}$$

Remark 2.1. Problem (2.1) is closely related to the formulation

$$\begin{array}{ll} \min & \delta \\ \text{s.t.} & Ax + \widetilde{x} = 0, \\ & x \geq 0, \\ & \|x\| \leq 1, \\ & \|\widetilde{x}\| \leq \delta, \end{array}$$

which is a particular case—the one obtained for the nonnegative orthant cone—of the formulation proposed by Peña and Renegar in [15] for general conic systems. Indeed, many of the ideas and arguments in this paper rely on results developed in [15] and the related work by Renegar [17]. The approach in [15, 17] is purely primal and in principle only applies to the primal constraints $Ax = 0,\ x \geq 0$. One of the main features of this paper is the primal-dual approach, which yields a unified treatment for the dual system $A^{\mathrm{T}}y \leq 0$ as well.

Unlike the formulation in [15], where the Euclidean norm is used to bound the primal and relaxation variables, we here choose the $\|\cdot\|_1$ and $\|\cdot\|_\infty$ norms. This choice of norms yields two main advantages. On the one hand, it allows us to state the formulation as a standard linear program. On the other hand, it readily yields a primal-dual formulation for both $Ax = 0,\ x \geq 0$, and $A^{\mathrm{T}}y \leq 0$.

For some time we will make the following assumption on the input matrix A.

Assumption 1. For $i = 1, \ldots, n$

$$\|Ae_i\|_1 = 1.$$

This is equivalent to the assumption that, if A_i^{T} denotes the ith row of A^{T}, $\|A_i^{\mathrm{T}}\|_1 = 1$ for $i = 1, \ldots, n$. This assumption is trivial from a computational viewpoint; it takes a few operations to reduce the matrix to this form. The condition number of the new matrix may have changed, however. Most of this paper will be devoted to proving Theorem 1.2 for matrices satisfying the above assumption. In section 11 we will extend the result to arbitrary matrices.

Notice that, as a consequence of Assumption 1, $1 \leq \|A\|_{1,\infty} \leq n$.

2.2. The central path. The primal-dual pair (2.5)–(2.6) can be straightforwardly solved via a primal-dual interior-point algorithm. For our purposes, we shall apply a short-step path-following algorithm (cf. [12, 25]). Our primary goal is to analyze such an algorithm in the presence of finite-precision arithmetic. Traditional convergence analyses of interior-point methods do not address this issue. We shall show that, in spite of the presence of finite precision, the fundamental steps of a typical convergence analysis can still be carried through.

Recall that the *central path* $\mathcal{C}$ of the pair (2.5)–(2.6) is the set of solutions of the nonlinear system of equations

$$\begin{array}{rcl} \mathcal{A}\vec{x} & = & \vec{b}, \\ \mathcal{A}^{\mathrm{T}}\vec{y} + \vec{s} & = & \vec{c}, \\ \vec{X}\vec{S}e & = & \mu e, \end{array} \tag{2.7}$$

with $\vec{x}, \vec{s} \geq 0$ for all values of the parameter $\mu > 0$. We have used here the following common notational convention. If a lowercase letter denotes a point in $\mathbb{R}^n$, then the corresponding uppercase letter will denote the $n \times n$ diagonal matrix whose diagonal elements are the coordinates of the given point. Thus, $\vec{X}$ denotes the diagonal matrix with $\vec{x}$ in the diagonal. We shall use this convention throughout the paper.

Let w denote a generic point $(\vec{x}, \vec{y}, \vec{s})$, and for such a point define

$$\mu(w) := \frac{e^{\mathrm{T}}\vec{X}\vec{S}e}{2(m+n)} = \frac{1}{2(m+n)} \sum_{i=1}^{2(m+n)} \vec{x}_i \vec{s}_i.$$

Note that if $w \in \mathcal{C}$ for a certain value of μ, then $\mu(w) = \mu$. We may sometimes write μ for $\mu(w)$ when w is clear from the context.

2.3. The algorithm. We are now ready to describe our primal-dual algorithm. This is essentially a standard primal-dual short-step algorithm (cf. [12] or [25, Chapter 5]) enhanced with two additional features. One of these features is the stopping criteria and the other one is the presence of finite precision and the adjustment of this precision as the algorithm progresses. To ensure the correctness of the algorithm, the precision will be set to

$$\phi(\mu(w)) := \min\{\mu(w)^2, 1\} \frac{1}{\mathbf{c}(m+n)^{12}}$$

at each iteration. Here $\mathbf{c}$ is a universal constant.

Let $\beta = 1/4$ and $\xi = 1/12$.

ALGORITHM FPPD(A, γ).

(i) Set the machine precision to $u := 1/\mathbf{c}(m+n)^{12}$,
$K := \frac{2mn}{\beta}$,
$w := \left(\frac{1}{2}e, Ke, \frac{1}{2}Ae + Ke, \frac{1}{2}e, 0, -2Ke, 2Ke, e, e, 2Ke\right)$.

(ii) Set the machine precision to $u := \phi(\mu(w))$.

(iii) If $A^{\mathrm{T}}y < -2u\left(\lceil \log_2 m \rceil + 1\right)e$, then HALT and return y as a feasible solution for $A^{\mathrm{T}}y < 0$.

(iv) If $\sigma_{\min}(X^{1/2}S^{-1/2}A^{\mathrm{T}}) > \frac{3(m+n)\mu(w)^{1/2}}{\gamma(1-2\beta)}$, then HALT and return x as a γ-forward solution for $Ax = 0,\ x > 0$.

(v) Set $\overline{\mu} := \left(1 - \frac{\xi}{\sqrt{2(m+n)}}\right)\mu(w)$.

(vi) Update w by solving a linearization of (2.7) for $\mu = \overline{\mu}$.

(vii) Go to (ii).

Remark 2.2.

(i) The expression FPPD stands for Finite Precision Primal Dual.

(ii) We already remarked that when $n \leq m$, the system (1.2) has strict solutions except when A is ill-posed. Therefore, in this case, Algorithm FPPD will systematically skip step (iv).

(iii) The choice of β and ξ above is somehow arbitrary. Any constants $\beta, \xi \in (0, 1/4]$ satisfying

$$\frac{(1-\beta)^3}{2(m+n)+1} \geq \frac{1}{10(m+n)}, \tag{2.8}$$

$$\frac{(\beta+\xi)^2}{\sqrt{2}(1-2\beta)} \leq \left(1 - \frac{3\xi}{2\sqrt{2(m+n)}}\right)\beta - \frac{5\xi}{6}, \tag{2.9}$$

$$\xi \leq \beta \leq 10\xi, \tag{2.10}$$

will ensure the correctness of our algorithm.

2.4. Plan of the paper. The rest of this paper is organized as follows. In section 3 we present several results that show the correctness of Algorithm FPPD, i.e., Theorem 1.2. This section develops two crucial pieces in our work. The first one is the way in which the update of w in step (vi) is performed. The second one is Theorem 3.3, which encapsulates the essence of the effects of finite precision in our computations.

Sections 3 to 6 mostly deal with convergence properties of Algorithm FPPD. Arguments here are of the kind used in convergence analysis of interior-point methods. Round-off analysis is delayed until sections 7 to 9. In this way, that part of the paper is clearly separated from the convergence analysis mentioned before, and the reader can get the conceptual content of the latter without being encumbered with the technicalities of the former.

Section 11 concludes the paper with some final remarks and details.

3. Proof of the main theorem.

3.1. On the update of w. A key step in the analysis of our algorithm is understanding the properties of the update of w performed at step (vi). The update w^+ is defined as $w^+ = w - \Delta w$, where $\Delta w = (\Delta\vec{x}, \Delta\vec{y}, \Delta\vec{s})$ solves the linear system

$$\begin{bmatrix} \mathcal{A} & & \\ & \mathcal{A}^{\mathrm{T}} & I \\ \vec{S} & & \vec{X} \end{bmatrix} \begin{bmatrix} \Delta\vec{x} \\ \Delta\vec{y} \\ \Delta\vec{s} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vec{X}\vec{S}e - \overline{\mu}e \end{bmatrix}. \tag{3.1}$$

Via some elementary row operations, the solution of (3.1) can be obtained as follows. First, solve the reduced system

$$(\mathcal{A}\vec{S}^{-1}\vec{X}\mathcal{A}^{\mathrm{T}})\Delta\vec{y} = -\mathcal{A}\vec{S}^{-1}(\vec{X}\vec{S}e - \overline{\mu}e). \tag{3.2}$$

Then set

$$\Delta\vec{s} = -\mathcal{A}^{\mathrm{T}}\Delta\vec{y} \tag{3.3}$$

and

$$\Delta\vec{x} = \vec{S}^{-1}\vec{X}\mathcal{A}^{\mathrm{T}}\Delta\vec{y} + \vec{S}^{-1}(\vec{X}\vec{S}e - \overline{\mu}e). \tag{3.4}$$

If (3.2), (3.3), and (3.4) were performed with exact arithmetic, then $(\Delta\vec{x}, \Delta\vec{y}, \Delta\vec{s})$ would solve (3.1) exactly. In such an ideal case, the convergence of Algorithm FPPD would readily follow from an analysis that is fairly standard and well-documented today; see, e.g., [12] or [25, Chapter 5]. Due to the use of finite precision we will have to proceed a bit differently.

Because of round-off errors, the computed Δw might not satisfy the first two constraints in (3.1). To circumvent this difficulty we exploit the role of some of the variables in our formulation as slack variables. The following construction formalizes this idea.

Given a point $w = (\vec{x}, \vec{y}, \vec{s})$, let us denote by $\underline{w} := (\underline{\vec{x}}, \vec{y}, \underline{\vec{s}})$ the *exact* vector

$$(x, x', Ax + x', e - x, y, y', -A^{\mathrm{T}}y - y', e - y, e + y, -y').$$

Notice that although this mathematical object is perfectly defined, we would not be able to explicitly represent all of its components in a finite-precision machine. Our analysis, however, will crucially rely on $\underline{w}$ and its properties.

An immediate consequence of this construction is that for any w, $\underline{w} := (\underline{\vec{x}}, \vec{y}, \underline{\vec{s}})$ satisfies $\mathcal{A}\underline{\vec{x}} = \vec{b}$ and $\mathcal{A}^{\mathrm{T}}\vec{y} + \underline{\vec{s}} = \vec{c}$.

The precise way in which we update w in step (vi) will be as follows. (All computations below are performed with precision $\phi(\mu(w))$.)

(a) Compute a solution $\Delta\vec{y}$ of

$$(\mathcal{A}\vec{S}^{-1}\vec{X}\mathcal{A}^{\mathrm{T}})\Delta\vec{y} = -\mathcal{A}\vec{S}^{-1}(\vec{X}\vec{S}e - \overline{\mu}e)$$

and let $\vec{y} := \vec{y} - \Delta\vec{y}$.

(b) Let

$$\begin{bmatrix} \Delta x \\ \Delta x' \end{bmatrix} := \begin{bmatrix} XS^{-1} & \\ & X'(S')^{-1} \end{bmatrix} \begin{bmatrix} A^{\mathrm{T}} & I \\ I & 0 \end{bmatrix} \Delta\vec{y} + \begin{bmatrix} S^{-1} & \\ & (S')^{-1} \end{bmatrix} \left(\begin{bmatrix} XSe \\ X'S'e \end{bmatrix} - \overline{\mu}e \right),$$

$x := x - \Delta x$,
$x' := x' - \Delta x'$.

(c) Let
$x'' := Ax + x'$,
$x''' := e - x$,
$s := -A^{\mathrm{T}}y - y'$,
$s' := e - y$,
$s'' := e + y$,
$s''' := -y'$.

If (c) were carried out with exact arithmetic, then the new iterate w^+ would satisfy $w^+ = \underline{w^+}$. This will not be the case because of the finite precision on the computations. However, the difference between these vectors can be bounded componentwise, as the following proposition states.

PROPOSITION 3.1. *If* (c) *above is performed with precision* $\phi(\mu(w))$, *then the new iterate* w^+ *satisfies*

$$\|w^+ - \underline{w^+}\|_\infty \leq \frac{\beta \min\{\mu(w^+), 1\}}{20(m+n)^2}. \tag{3.5}$$

Proof. For the proof of this proposition, see section 10. □

The considerations above suggest defining the following two enlargements of the central path. They will play a central role in our development.

DEFINITION 3.2. *Given* $\beta \in (0, \frac{1}{4}]$, *the* central neighborhood $\mathcal{N}_\beta$ *is defined as the set of points* $w = (\vec{x}, \vec{y}, \vec{s})$, *with* $\vec{x},\ \vec{s} > 0$, *such that the following constraints hold:*

$$\begin{aligned} \mathcal{A}\vec{x} &= \vec{b}, \\ \mathcal{A}^{\mathrm{T}}\vec{y} + \vec{s} &= \vec{c}, \\ \|\vec{X}\vec{S}e - \mu(w)e\| &\leq \beta\mu(w). \end{aligned}$$

The extended central neighborhood $\overline{\mathcal{N}_\beta}$ *is thus defined by*

$$\overline{\mathcal{N}_\beta} := \left\{ w : \underline{w} \in \mathcal{N}_\beta \text{ and } \|w - \underline{w}\|_\infty \leq \frac{\beta \min\{\mu(w), 1\}}{20(m+n)^2} \right\}.$$

Remark 3.1. Ideally, one would like to generate a sequence of points on the central path $\mathcal{C}$ with values of μ decreasing to zero. Limitations on our ability to solve

nonlinear systems have led interior-point methods to generate the sequence above in the central neighborhood $\mathcal{N}_\beta$. The additional limitations arising from the use of finite precision lead us to generate this sequence in the extended central neighborhood $\overline{\mathcal{N}_\beta}$.

We are now ready to present stepping stones towards the proof of Theorem 1.2.

THEOREM 3.3. *Let $w \in \overline{\mathcal{N}_\beta}$, suppose $w^+ = w - \Delta w$ is obtained using steps* (a)–(c) *above, and let $\Delta\underline{w} = (\Delta\underline{\vec{x}}, \Delta\vec{y}, \Delta\underline{\vec{s}}) := \underline{w} - \underline{w^+}$. Then*

$$\begin{aligned} \mathcal{A}\Delta\underline{\vec{x}} &= 0, \\ \mathcal{A}^{\mathrm{T}}\Delta\vec{y} + \Delta\underline{\vec{s}} &= 0, \\ \vec{S}\Delta\underline{\vec{x}} + \vec{X}\Delta\underline{\vec{s}} &= \vec{X}\vec{S}e - \overline{\mu}e + r, \end{aligned}$$

with $\|r\| \leq \frac{\xi\mu(w)}{3}$.

Proof. For the proof see section 4. ∎

PROPOSITION 3.4. *Let $w \in \overline{\mathcal{N}_\beta}$ and suppose that $w^+ = w - \Delta w$ is obtained using steps* (a)–(c) *above. Then*

(i)

$$\left(1 - \frac{3\xi}{2\sqrt{2(m+n)}}\right)\mu(w) \leq \mu(w^+) \leq \left(1 - \frac{\xi}{2\sqrt{2(m+n)}}\right)\mu(w)$$

and

(ii) $w^+ \in \overline{\mathcal{N}_\beta}$.

Proof. For the proof see section 10. ∎

PROPOSITION 3.5. *For $K \geq \frac{2mn}{\beta}$ the point $w_0 = (\vec{x}, \vec{y}, \vec{s})$, defined as follows, belongs to $\mathcal{N}_\beta$:*

$$x = x''' = \frac{1}{2}e, \quad x' = Ke, \quad x'' = \frac{1}{2}Ae + Ke,$$

$$y = 0, \quad s' = s'' = e, \quad s = s''' = -y' = 2Ke.$$

In addition, if this point is computed with precision $\mathbf{c}(m+n)^{-12}$, then the computed point $\mathtt{fl}(w_0)$ belongs to $\overline{\mathcal{N}_\beta}$.

Proof. For the proof see section 7. ∎

PROPOSITION 3.6. *If in step* (iii) *the algorithm above yields (with round-off errors) $A^{\mathrm{T}}y < -2u\,(\lceil \log_2 m \rceil + 1)\,e$, then y is a strict solution of* (1.2), *i.e., $A^{\mathrm{T}}y < 0$.*

Proof. The proof can be found in section 7. ∎

PROPOSITION 3.7. *Assume $w \in \overline{\mathcal{N}_\beta}$. If in step* (iv) *the algorithm above yields (with round-off errors)*

$$\sigma_{\min}(X^{1/2}S^{-1/2}A^{\mathrm{T}}) > \frac{3(m+n)\mu(w)^{1/2}}{\gamma(1-2\beta)},$$

then x is a γ-forward solution of (1.1), *and the projection*

$$\overline{x} = x - XS^{-1}A^{\mathrm{T}}(AXS^{-1}A^{\mathrm{T}})^{-1}Ax$$

is an associated solution for x.

Proof. The proof can be found in section 9. ∎

PROPOSITION 3.8. *Assume* $\rho_D(A) > 0$ *and* $w \in \overline{\mathcal{N}_\beta}$. *If* $\mu(w) \le \frac{\rho_D(A)}{20(n+m)^2}$, *then* $A^{\mathrm{T}}y < -4u(\lceil \log_2 m \rceil + 1)e$ *holds exactly and the algorithm halts in step* (iii).

Proof. For the proof see section 7. ∎

PROPOSITION 3.9. *Let* $\gamma \in (0,1)$, *assume* $w \in \overline{\mathcal{N}_\beta}$ *and* $\rho_P(A) > 0$. *If*

$$\mu(w) \le \frac{(1-2\beta)^2 \rho_P(A)}{20(m+n)^{5/2}} \left(1 + \frac{1}{\gamma}\right)^{-1},$$

then $\sigma_{\min}(X^{1/2}S^{-1/2}A^{\mathrm{T}}) > \frac{4(m+n)\mu(w)^{1/2}}{\gamma(1-2\beta)}$ *holds exactly and the algorithm halts in step* (iv).

Proof. For the proof see section 9. ∎

3.2. Proof of Theorem 1.2. Propositions 3.6 to 3.9 show that, for $w \in \overline{\mathcal{N}_\beta}$ with $\mu(w)$ sufficiently small, the algorithm halts and yields a solution. Propositions 3.4 and 3.5 show that the algorithm generates such a point $w \in \overline{\mathcal{N}_\beta}$ in at most

$$\mathcal{O}((m+n)^{1/2}(\log(m+n) + \log(C(A)) + |\log\gamma|))$$

iterations when $\rho_P(A) > 0$, and in at most

$$\mathcal{O}((m+n)^{1/2}(\log(m+n) + \log(C(A))))$$

iterations when $\rho_D(A) > 0$. ∎

4. Proof of Theorem 3.3. Recall that in section 3.1 we defined $\Delta\vec{y}$ to be the solution of

$$(\mathcal{A}\vec{S}^{-1}\vec{X}\mathcal{A}^{\mathrm{T}})\Delta\vec{y} = -\mathcal{A}\vec{S}^{-1}(\vec{X}\vec{S}e - \overline{\mu}e)$$

computed with precision $\phi(\mu(w))$. Therefore, $\Delta\vec{y}$ *does not* actually satisfy the equality above. To prove Theorem 3.3, however, we do not need this equality to be satisfied. It suffices that both sides of it are close enough. The following proposition quantifies this closeness.

PROPOSITION 4.1. *Let* $w \in \overline{\mathcal{N}_\beta}$. *With precision* $u = \phi(\mu)$ *in all arithmetic operations, we can obtain a vector* $\Delta\vec{y}$ *such that*

$$\|(\mathcal{A}\vec{S}^{-1}\vec{X}\mathcal{A}^{\mathrm{T}})\Delta\vec{y} + \mathcal{A}\vec{S}^{-1}(\vec{X}\vec{S}e - \overline{\mu}e)\| \le \frac{\xi \min\{1/(m+n), \mu\}}{7}.$$

We will prove Proposition 4.1 in section 8. Its proof is the heart of our round-off analysis.

Remark 4.1. We next prove Theorem 3.3. The proof we give ignores the round-off error present in the computation of $\Delta x, \Delta x'$. Formally speaking, $\Delta x, \Delta x'$ do not satisfy (b) exactly. Instead we actually have

$$\begin{bmatrix} \Delta x \\ \Delta x' \end{bmatrix} = \begin{bmatrix} XS^{-1} & \\ & X'(S')^{-1} \end{bmatrix} \begin{bmatrix} A^{\mathrm{T}} & I \\ I & 0 \end{bmatrix} \Delta\vec{y} + \begin{bmatrix} S^{-1} & \\ & (S')^{-1} \end{bmatrix} \left(\begin{bmatrix} XSe \\ X'S'e \end{bmatrix} - \overline{\mu}e \right) + \eta$$

for some error vector η. However, it is easy to show (see section 11.1) that this error vector satisfies

$$\left\| \vec{S} \begin{bmatrix} \eta \\ 0 \end{bmatrix} \right\| \le \frac{\xi\mu(w)}{10},$$

and hence the proof below can readily be amended. We ignore this minor detail to avoid obscuring the essence of the argument.

Proof of Theorem 3.3. By construction, $(\Delta\underline{\vec{x}}, \Delta\vec{y}, \Delta\underline{\vec{s}})$ satisfies

$$\begin{aligned} \mathcal{A}\Delta\underline{\vec{x}} &= 0, \\ \mathcal{A}^{\mathrm{T}}\Delta\vec{y} + \Delta\underline{\vec{s}} &= 0. \end{aligned}$$

In addition, simple algebraic manipulations show that the construction of $(\Delta\underline{\vec{x}}, \Delta\vec{y}, \Delta\underline{\vec{s}})$ also imposes the condition that $\Delta\underline{\vec{x}}$ be equal to

$$\vec{S}^{-1}(-\vec{X}\Delta\underline{\vec{s}} + (\vec{X}\vec{S}e - \overline{\mu}e)) + \begin{bmatrix} 0 \\ 0 \\ \begin{bmatrix} I & \\ & -I \end{bmatrix} ((\mathcal{A}\vec{S}^{-1}\vec{X}\mathcal{A}^{\mathrm{T}})\Delta\vec{y} + \mathcal{A}\vec{S}^{-1}(\vec{X}\vec{S}e - \overline{\mu}e)) \end{bmatrix}.$$

Hence,

$$\vec{S}\Delta\underline{\vec{x}} + \vec{X}\Delta\underline{\vec{s}} - (\vec{X}\vec{S}e - \overline{\mu}e) = \begin{bmatrix} 0 \\ 0 \\ \begin{bmatrix} S'' & \\ & -S''' \end{bmatrix} ((\mathcal{A}\vec{S}^{-1}\vec{X}\mathcal{A}^{\mathrm{T}})\Delta\vec{y} + \mathcal{A}\vec{S}^{-1}(\vec{X}\vec{S}e - \overline{\mu}e)) \end{bmatrix}.$$

Therefore, by Proposition 4.1,

$$\left\|\vec{S}\Delta\underline{\vec{x}} + \vec{X}\Delta\underline{\vec{s}} - (\vec{X}\vec{S}e - \overline{\mu}e)\right\| \leq \left\|\begin{bmatrix} S'' & \\ & S''' \end{bmatrix}\right\| \frac{\xi \min\{1/(m+n), \mu\}}{7}.$$

But $\|s''\|_\infty \leq 1 + \|y\|_\infty + \|s'' - \underline{s}''\|_\infty \leq 2 + \beta/20(m+n)^2$, and therefore $\|S''\| = \|s''\|_\infty \leq 7/3$. Also, $s''' = -y'$ implies $\|S'''\| = \|s'''\|_\infty = \|y'\|_\infty \leq -\vec{b}^{\mathrm{T}}\vec{y} = 2(m+n)\mu - \vec{c}^{\mathrm{T}}\vec{x} \leq 2(m+n)\mu$. Thus

$$\begin{aligned} \|r\| &= \|\vec{S}\Delta\underline{\vec{x}} + \vec{X}\Delta\underline{\vec{s}} - (\vec{X}\vec{S}e - \overline{\mu}e)\| \\ &\leq \max\{\|S''\|, \|S'''\|\} \frac{\xi \min\{1/(m+n), \mu\}}{7} \\ &\leq \max\{7/3, 2(m+n)\mu\} \frac{\xi \min\{1/(m+n), \mu\}}{7} \\ &\leq \frac{\xi\mu}{3}. \qquad \square \end{aligned}$$

5. Some useful bounds. The following proposition states a key property of the points in the neighborhood $\overline{\mathcal{N}_\beta}$ in connection with the systems $Ax = 0,\ x \geq 0$, and $A^{\mathrm{T}}y \leq 0$.

PROPOSITION 5.1. *Let* $w = (\vec{x}, \vec{y}, \vec{s}) \in \overline{\mathcal{N}_\beta}$. *Then*

$$x \geq \frac{\max\{\rho_P(A),\, m\mu(w)\}}{10(n+m)}\, e$$

and

$$s \geq \frac{\max\{\rho_D(A),\, n\mu(w)\}}{10(n+m)}\, e.$$

To prove Proposition 5.1 we will rely on the following characterization of $\rho_P(A)$, $\rho_D(A)$ due to Renegar. (For a detailed discussion, see [14, 16].)

PROPOSITION 5.2. *Let $A \in \mathbb{R}^{m\times n}$. Then*

$$\rho_P(A) = \inf\{\|\overline{y}\|_1 : \nexists x \text{ such that } Ax = \overline{y},\ x \geq 0,\ \|x\|_\infty \leq 1\}$$

and

$$\rho_D(A) = \inf\{\|\overline{x}\|_1 : \nexists y \text{ such that } A^{\mathrm{T}}y - \overline{x} \leq 0,\ \|y\|_\infty \leq 1\}. \qquad \square$$

We will also use the following two technical lemmas. The first one summarizes some basic properties satisfied by the points in $\mathcal{N}_\beta$ and $\overline{\mathcal{N}_\beta}$. These bounds will be used throughout the rest of the paper.

LEMMA 5.3. *Let $w \in \mathcal{N}_\beta$. Then*

(i) *For $i = 1, \ldots, 2(n+m)$, $\vec{x}_i\vec{s}_i \geq (1-\beta)\mu(w)$;*
(ii) $\vec{c}^{\mathrm{T}}\vec{x} \geq (1-\beta)m\mu(w)$;
(iii) $-\vec{b}^{\mathrm{T}}\vec{y} \geq (1-\beta)n\mu(w)$;
(iv) $\|x'\|_1 + \|x''\|_1 = \vec{c}^{\mathrm{T}}\vec{x} \leq 2(m+n)\mu(w)$.

Furthermore, if $w \in \overline{\mathcal{N}_\beta}$, then similar bounds hold with $(1-2\beta)$ instead of $(1-\beta)$ in (i), (ii), *and* (iii).

Proof. Since $w \in \mathcal{N}_\beta$, $\|\vec{X}\vec{S}e - \mu(w)e\| \leq \beta\mu(w)$. Thus, for $i = 1, \ldots, 2(n+m)$, $|\vec{x}_i\vec{s}_i - \mu(w)| \leq \beta\mu(w)$. From here it follows that $\vec{x}_i\vec{s}_i \geq \mu(w) - \beta\mu(w)$, and hence we obtain (i).

For part (ii) suppose $\vec{c}^{\mathrm{T}}\vec{x} < (1-\beta)m\mu$; then by the pigeonhole principle

$$x_i < \frac{(1-\beta)\mu}{2}$$

for some $i \in \{n+1, \ldots, n+2m\}$. Since $x_i s_i \geq (1-\beta)\mu$,

$$s_i \geq \frac{(1-\beta)\mu}{x_i} > 2.$$

But this is a contradiction, since $w \in \mathcal{N}_\beta$ implies $s_i \leq 1 + \|y\|_\infty \leq 2$. Thus $\vec{c}^{\mathrm{T}}\vec{x} \geq (1-\beta)m\mu$.

A similar argument shows part (iii).

Finally, from the two equality constraints in the definition of $\mathcal{N}_\beta$ it follows that $\vec{c}^{\mathrm{T}}\vec{x} - \vec{b}^{\mathrm{T}}\vec{y} = \vec{s}^{\mathrm{T}}\vec{x}$. Since $\vec{b}^{\mathrm{T}}\vec{y} = -\|y'\|_1$, we have $\vec{c}^{\mathrm{T}}\vec{x} \leq \vec{s}^{\mathrm{T}}\vec{x}$. Part (iv) then follows since $\vec{s}^{\mathrm{T}}\vec{x} = 2(m+n)\mu(w)$ by definition of $\mu(w)$.

The statements for $w \in \overline{\mathcal{N}_\beta}$ are readily obtained by applying (i)–(iv) to the point $\underline{w} \in \mathcal{N}_\beta$ and using Claim 1 below. $\square$

CLAIM 1. *Suppose that $w = (\vec{x}, \vec{y}, \vec{s})$ is such that $\vec{x},\ \vec{s} > 0$, and $\|w - \underline{w}\|_\infty \leq \epsilon < 1$. Then*

$$\|x\|_\infty,\ \|x'''\|_\infty \leq \frac{1}{1-\epsilon}, \quad \|s'\|_\infty,\ \|s''\|_\infty \leq \frac{2}{1-\epsilon},$$

$$\|x'\|_\infty,\ \|x''\|_\infty,\ \|s'''\|_\infty \leq \frac{2(m+n)\mu(w)}{1-\epsilon}, \quad \|s\|_\infty \leq \frac{2 + 2(m+n)\mu(w)}{1-\epsilon},$$

and

$$|\mu(\underline{w}) - \mu(w)| < \max\{2, 2(m+n)\mu(w)\}\frac{(2+\epsilon)\epsilon}{1-\epsilon}.$$

Proof. This readily follows by using the facts that $\mathcal{A}\underline{\vec{x}} = \vec{b}$ and $\mathcal{A}^{\mathrm{T}}\vec{y} + \underline{\vec{s}} = \vec{c}$ together with Assumption 1. ∎

The following lemma is simply a particular version of a result that holds for self-scaled barrier functions (cf. [18, Theorem 3.5.9]).

LEMMA 5.4. *Let* $u, v \in \mathbb{R}^d$, *with* $u > 0$, $v \geq 0$. *If* $\langle v - u, U^{-1}e \rangle \leq 0$, *then*

$$\|U^{-1}(v-u)\| \leq d. \qquad \square$$

Proof of Proposition 5.1. Let us first assume $w \in \mathcal{N}_\beta$. We shall show a slightly stronger inequality in this case.

Let $i \in \{1, \ldots, n\}$ and $\epsilon > 0$ be given. Choosing $\overline{y}$ as an appropriate multiple of Ae_i and applying Proposition 5.2, it is easy to see that there exists z such that

$$z_i \geq \rho_P(A) - \epsilon$$

and

$$Az = 0, \quad z \geq 0, \ \|z\|_\infty \leq 1.$$

Hence we can readily get $\vec{z} \geq 0$ such that

$$\mathcal{A}\vec{z} = \vec{b}, \quad \vec{z} \geq 0, \quad \vec{c}^{\mathrm{T}}\vec{z} \leq \vec{c}^{\mathrm{T}}\vec{x}, \quad \text{and } z_i \geq \max\{\vec{c}^{\mathrm{T}}\vec{x},\, \rho_P(A) - \epsilon\}.$$

Thus,

$$\left\langle \vec{z} - \mu(w)\vec{S}^{-1}e, \frac{\vec{s}}{\mu(w)} \right\rangle = \frac{1}{\mu(w)} \langle \vec{z} - \vec{x}, \vec{s} \rangle = \langle \vec{z} - \vec{x}, \vec{c} - \mathcal{A}^{\mathrm{T}}y \rangle = \frac{1}{\mu(w)} \langle \vec{z} - \vec{x}, \vec{c} \rangle \leq 0.$$

Hence Lemma 5.4 implies that $\|\frac{1}{\mu(w)}\vec{S}(\vec{z} - \mu(w)\vec{S}^{-1}e)\| \leq 2(m+n)$. A bit of algebra and Lemma 5.3 yield

$$x_i \geq \frac{(1-\beta)\max\{\vec{c}^{\mathrm{T}}\vec{x},\, \rho_P(A) - \epsilon\}}{2(m+n)+1} \geq \frac{(1-\beta)\max\{(1-\beta)m\mu(w),\, \rho_P(A) - \epsilon\}}{2(m+n)+1}.$$

Since $i \in \{1, \ldots, n\}$ and $\epsilon > 0$ are arbitrary,

$$x \geq \frac{(1-\beta)\max\{\rho_P(A),\, (1-\beta)m\mu(w)\}}{2(m+n)+1}\, e. \tag{5.1}$$

Now notice that, by Claim 1 and since $w \in \overline{\mathcal{N}_\beta}$, $\mu(\underline{w}) \geq (1-\beta)\mu(w)$. Hence (5.1) (applied to $\underline{w}$) and (2.8) yield the first inequality in the proposition.

The second inequality is proven by a "dual" argument. Again assume $w \in \mathcal{N}_\beta$ and let $i \in \{1, \ldots, n\}$ and $\epsilon > 0$ be given. Choosing $\overline{x}$ as an appropriate multiple of e_i and applying Proposition 5.2, it is easy to see that there exist v and z such that

$$z_i \geq \rho_D(A) - \epsilon$$

and

$$A^{\mathrm{T}}v + z = 0, \quad z \geq 0, \ \|v\|_\infty \leq 1.$$

Hence we can readily get $\vec{y}$ and $\vec{z}$ such that

$$\mathcal{A}\vec{v} + \vec{z} = \vec{c}, \quad \vec{z} \geq 0, \quad \vec{b}^{\mathrm{T}}\vec{v} \geq \vec{b}^{\mathrm{T}}\vec{y}, \quad \text{and } z_i \geq \max\{-\vec{b}^{\mathrm{T}}\vec{y},\, \rho_D(A) - \epsilon\}.$$

Thus

$$\left\langle \vec{z} - \mu(w)\vec{X}^{-1}e, \frac{\vec{x}}{\mu(w)} \right\rangle = \frac{1}{\mu(w)} \langle \vec{z} - \vec{s}, \vec{x} \rangle = \left\langle -\mathcal{A}^{\mathrm{T}}(\vec{v} - \vec{y}), \vec{x} \right\rangle = \frac{1}{\mu(w)} \left\langle \vec{y} - \vec{v}, \vec{b} \right\rangle \leq 0.$$

Hence Lemma 5.4 implies that $\|\frac{1}{\mu(w)}\vec{X}(\vec{z} - \mu(w)\vec{X}^{-1}e)\| \leq 2(m+n)$. In particular,

$$\left| z_i - \frac{\mu(w)}{x_i} \right| \leq 2(m+n)\frac{\mu(w)}{x_i} \Rightarrow \frac{\mu(w)}{x_i} \geq \frac{z_i}{2(m+n)+1},$$

but $x_i s_i \geq (1-\beta)\mu(w)$, $z_i \geq \max\{-\vec{b}^{\mathrm{T}}\vec{y},\, \rho_D(A) - \epsilon\}$, and $-\vec{b}^{\mathrm{T}}\vec{y} \geq (1-\beta)n\mu(w)$ (by Lemma 5.3), and so

$$s_i \geq \frac{(1-\beta)\max\{-\vec{b}^{\mathrm{T}}\vec{y},\, \rho_D(A) - \epsilon\}}{2(m+n)+1} \geq \frac{(1-\beta)\max\{(1-\beta)n\mu(w),\, \rho_D(A) - \epsilon\}}{2(m+n)+1}.$$

Since $i \in \{1, \ldots, n\}$ and $\epsilon > 0$ are arbitrary, we conclude

$$s \geq \frac{(1-\beta)\max\{\rho_D(A),\, (1-\beta)n\mu(w)\}}{2(m+n)+1}\, e. \tag{5.2}$$

The corresponding statement for $w \in \overline{\mathcal{N}_\beta}$ again follows as before. By Claim 1 and since $w \in \overline{\mathcal{N}_\beta}$, $\mu(\underline{w}) \geq (1-\beta)\mu(w)$. Hence (5.2) (applied to $\underline{w}$) and (2.8) yield the second inequality in the proposition. ∎

A first consequence of Proposition 5.1, formally stated in the corollary below, is that if $\rho_D(A) > 0$, then the algorithm will eventually produce iterates that satisfy $A^{\mathrm{T}}y < 0$.

COROLLARY 5.5. *Let $w = (\vec{x}, \vec{y}, \vec{s}) \in \overline{\mathcal{N}_\beta}$. If $\rho_D(A) \geq 20(m+n)^2\mu(w)$, then*

$$A^{\mathrm{T}}y < -\frac{n\mu(w)}{10(n+m)}. \qquad \blacksquare$$

On the other hand, Proposition 5.1 implies that if $\rho_P(A) > 0$, then x is a forward-approximate solution of $Ax = 0$, $x \geq 0$, when $w \in \overline{\mathcal{N}_\beta}$ with $\mu(w)$ sufficiently small. The next result proves a first step in this direction, showing that x is a forward-approximate solution if the singular values of $X^{1/2}S^{-1/2}A^{\mathrm{T}}$ are large enough. In the next section we prove that the latter is actually the case if $\mu(w)$ is sufficiently small.

PROPOSITION 5.6. *Let $w = (\vec{x}, \vec{y}, \vec{s}) \in \overline{\mathcal{N}_\beta}$ and $\gamma \in (0,1)$. If*

$$\sigma_{\min}(X^{1/2}S^{-1/2}A^{\mathrm{T}}) \geq \frac{2(m+n)\mu(w)^{1/2}}{(1-2\beta)\gamma},$$

then x is a γ-forward solution of $Ax = 0$, $x \geq 0$, and

$$\overline{x} = x - XS^{-1}A^{\mathrm{T}}(AXS^{-1}A^{\mathrm{T}})^{-1}Ax$$

is an associated solution for x.

Proof. The bound on $\sigma_{\min}(X^{1/2}S^{-1/2}A^{\mathrm{T}})$ implies

$$\begin{aligned} \|X^{1/2}S^{-1/2}A^{\mathrm{T}}(AXS^{-1}A^{\mathrm{T}})^{-1}Ax\| &\leq \frac{\|Ax\|}{\sigma_{\min}(X^{1/2}S^{-1/2}A^{\mathrm{T}})} \\ &\leq \frac{2(m+n)\mu(w)}{\sigma_{\min}(X^{1/2}S^{-1/2}A^{\mathrm{T}})} \\ &\leq \gamma(1-2\beta)\mu(w)^{1/2}, \end{aligned}$$

the second inequality by Lemma 5.3 since

$$\|Ax\| = \|x' + x''\| \leq \|x'\|_1 + \|x''\|_1 \leq 2(m+n)\mu(w).$$

On the other hand, since $w \in \overline{\mathcal{N}_\beta}$, it follows easily from Lemma 5.3 that $x_i^{1/2} s_i^{-1/2} \leq \frac{x_i}{(1-2\beta)\mu(w)^{1/2}}$ for $i = 1, \ldots, n$. Thus

$$|(XS^{-1}A^{\mathrm{T}}(AXS^{-1}A^{\mathrm{T}})^{-1}Ax)_i| \leq \gamma x_i < x_i, \quad i = 1, \ldots, n.$$

In consequence, the point

$$\overline{x} = x - XS^{-1}A^{\mathrm{T}}(AXS^{-1}A^{\mathrm{T}})^{-1}Ax$$

satisfies $A\overline{x} = 0$, $\overline{x} > 0$, and, for $i = 1, \ldots, n$, $|x_i - \overline{x}_i| \leq \gamma x_i$. ∎

6. On the singular values of $X^{1/2}S^{-1/2}A^{\mathrm{T}}$. The following two technical results are crucial in our round-off analysis. They are in the same spirit as [15, Theorems 5, 6, 10] and [17, Corollaries 1.3, 1.7].

PROPOSITION 6.1. *Let $w = (\vec{x}, \vec{y}, \vec{s}) \in \overline{\mathcal{N}_\beta}$ and $\gamma \in (0,1)$. If*

$$\rho_P(A) \geq \frac{20(m+n)^{5/2}\mu(w)}{(1-2\beta)^2}\left(1 + \frac{1}{\gamma}\right),$$

then

$$\sigma_{\min}(X^{1/2}S^{-1/2}A^{\mathrm{T}}) \geq \frac{4(m+n)\mu(w)^{1/2}}{(1-2\beta)\gamma}.$$

PROPOSITION 6.2. *Assume $w \in \overline{\mathcal{N}_\beta}$ and let $B = \vec{X}^{1/2}\vec{S}^{-1/2}\mathcal{A}^{\mathrm{T}}$. Then*

$$\sigma_{\min}(\mu(w)^{1/2}B) \geq \frac{\min\{m\mu(w) + \rho_P(A), 1\}}{20(m+n)^{3/2}}$$

and

$$\sigma_{\max}(\mu(w)^{1/2}B) = \mu(w)^{1/2}\|B\| \leq \frac{3\max\{2(m+n)\mu(w), 1\}\sqrt{n}}{2(1-2\beta)}.$$

In particular,

$$\kappa(B) := \frac{\sigma_{\max}(B)}{\sigma_{\min}(B)} \leq 60\sqrt{n}(m+n)^{3/2}\max\left\{\frac{1}{m\mu(w)}, 2(m+n)\mu(w)\right\}.$$

The proof of Proposition 6.2 relies on the following technical lemma, which is a modification of [15, Theorem 12].

LEMMA 6.3. *Assume $w = (\vec{x}, \vec{y}, \vec{s}) \in \overline{\mathcal{N}_\beta}$. Let $\overline{b} \in \mathbb{R}^{2(m+n)}$ be such that for any $\vec{x}' \in \mathbb{R}^{2(m+n)}$ with $\mathcal{A}\vec{x}' = \overline{b}$ one has*

$$\left\| \frac{1}{\mu^{1/2}} \vec{X}^{-1/2}\vec{S}^{1/2}\vec{x}' \right\| \geq 1.$$

If $\alpha > \frac{2(m+n)+2\sqrt{2(m+n)}\beta}{1-2\beta}$, then for either $\Delta b = \alpha\overline{b}$ or $\Delta b = -\alpha\overline{b}$ the optimal value of the perturbed problem

$$\begin{array}{ll} \min & \vec{c}^{\mathrm{T}}\vec{z} \\ \text{s.t.} & \mathcal{A}\vec{z} = \vec{b} + \Delta b, \\ & \vec{z} \geq 0, \end{array} \tag{6.1}$$

is greater than $\vec{c}^{\mathrm{T}}\vec{x} - \frac{\mu(w)}{4}$.

Proof. Assuming $\langle \vec{y}, \bar{b} \rangle \geq 0$, we will prove the lemma with $\Delta b = \alpha \bar{b}$. (If $\langle \vec{y}, \bar{b} \rangle \leq 0$, we would instead prove it for $\Delta b = -\alpha \bar{b}$.)

If (6.1) is infeasible, then its optimal value is ∞ and there is nothing to prove. Hence assume (6.1) is feasible. We proceed by contradiction: suppose that $\vec{z}$ is feasible and

$$\langle \vec{c}, \vec{z} \rangle \leq \langle \vec{c}, \vec{x} \rangle - \frac{\mu(w)}{4}. \tag{6.2}$$

Because $\vec{z}$ is feasible and satisfies (6.2),

$$\langle \vec{z} - \vec{x}, \vec{s} \rangle = \langle \vec{z} - \underline{\vec{x}}, \underline{\vec{s}} \rangle + \langle \underline{\vec{x}} - x, \underline{\vec{s}} \rangle + \langle \vec{z} - \vec{x}, \vec{s} - \underline{\vec{s}} \rangle .$$

Now, since $w \in \overline{\mathcal{N}_\beta}$ and z is feasible for (6.1), it can easily be shown that

$$| \langle \underline{\vec{x}} - x, \underline{\vec{s}} \rangle + \langle \vec{z} - \vec{x}, \vec{s} - \underline{\vec{s}} \rangle | \leq \frac{\mu(w)}{4}.$$

Hence

$$\begin{aligned} \langle \vec{z} - \vec{x}, \vec{s} \rangle &\leq \left\langle \vec{z} - \underline{\vec{x}}, c + \mathcal{A}^{\mathrm{T}} \vec{y} \right\rangle + \frac{\mu(w)}{4} \\ &\leq \left\langle \vec{z} - \underline{\vec{x}}, \mathcal{A}^{\mathrm{T}} \vec{y} \right\rangle = \langle \mathcal{A}(\vec{z} - \underline{\vec{x}}), \vec{y} \rangle \\ &= -\alpha \left\langle \bar{b}, \vec{y} \right\rangle \leq 0. \end{aligned}$$

Thus

$$\left\langle \vec{z} - \mu(w) \vec{S}^{-1} e, \frac{1}{\mu(w)} s \right\rangle = \frac{1}{\mu(w)} \langle \vec{z} - \vec{x}, \vec{s} \rangle \leq 0.$$

Hence Lemma 5.4 implies that

$$\left\| \frac{1}{\mu(w)} \vec{S} \left(\vec{z} - \mu(w) \vec{S}^{-1} e \right) \right\| \leq 2(m+n). \tag{6.3}$$

On the other hand, since $w \in \overline{\mathcal{N}_\beta}$, for each $i = 1, \ldots, 2(m+n)$, $|x_i^{1/2} s_i^{1/2} - \mu(w)^{1/2}| \leq 2\beta \mu(w)^{1/2} \Rightarrow 1 - 2\beta \leq \frac{s/\mu}{x^{-1/2} s^{1/2} / \mu^{1/2}} \leq 1 + 2\beta$,

$$\left\| \frac{1}{\mu(w)^{1/2}} \vec{X}^{-1/2} \vec{S}^{1/2} \left(\vec{z} - \mu(w) \vec{S}^{-1} e \right) \right\| \leq \frac{2(m+n)}{1 - 2\beta}, \tag{6.4}$$

and

$$\left\| \frac{1}{\mu(w)^{1/2}} \vec{X}^{-1/2} \vec{S}^{1/2} \left(\vec{x} - \mu(w) \vec{S}^{-1} e \right) \right\| \leq \frac{2\sqrt{2(m+n)}\beta}{1 - 2\beta}. \tag{6.5}$$

However, since $A(\vec{z} - \vec{x}) = \alpha \bar{b}$, our hypothesis implies that

$$\left\| \frac{1}{\mu(w)^{1/2}} \vec{X}^{-1/2} \vec{S}^{1/2} (\vec{x} - \vec{z}) \right\| \geq \alpha. \tag{6.6}$$

But (6.4), (6.5), and (6.6) are contradictory, assuming $\alpha > \frac{2(m+n) + 2\sqrt{2(m+n)}\beta}{1 - 2\beta}$. ∎

6.1. Proof of Proposition 6.2. From Proposition 5.2, it follows that for any $\Delta b^I \in \mathbb{R}^m$ with $\|\Delta b^I\|_1 < \rho_P(A)$ there exists a z such that

$$A\vec{z} = \Delta b^I, \quad z \geq 0, \quad \|z\|_\infty \leq 1.$$

Hence for any $\Delta b^I \in \mathbb{R}^m$ and $\alpha \geq 0$ with $\|\Delta b^I\|_1 < \alpha + \rho_P(A)$, there exists $\vec{z}$ such that

$$\mathcal{A}\vec{z} = \vec{b} + \begin{bmatrix} \Delta b^I \\ 0 \end{bmatrix}, \quad \vec{z} \geq 0, \text{ and } \vec{c}^{\mathrm{T}}\vec{z} \leq \alpha.$$

Consequently, if $\Delta b \in \mathbb{R}^{(m+n)}$ satisfies $\|\Delta b\|_1 < \min\{\vec{c}^{\mathrm{T}}\vec{x} - \frac{\mu(w)}{4} + \frac{\rho_P(A)}{2}, \frac{1}{2}\}$, then there exists $\vec{z}$ such that

$$\mathcal{A}\vec{z} = \vec{b} + \Delta b, \ \vec{z} \geq 0, \quad \text{and} \quad \vec{c}^{\mathrm{T}}\vec{z} \leq \vec{c}^{\mathrm{T}}\vec{x} - \frac{\mu(w)}{4}.$$

On the other hand, for any given $\epsilon > 0$, Lemma 6.3 implies that there exists a Δb satisfying

$$\|\Delta b\| \leq \frac{2(m+n) + \sqrt{2(m+n)}\beta + \epsilon}{1 - 2\beta}\, \sigma_{\min}(\mu(w)^{1/2}B) \tag{6.7}$$

and such that the optimal value of the following problem exceeds $\vec{c}^{\mathrm{T}}\vec{x} - \frac{\mu(w)}{4}$:

$$\begin{array}{ll} \min & \vec{c}^{\mathrm{T}}\vec{z} \\ \text{s.t.} & \mathcal{A}\vec{z} = \vec{b} + \Delta b, \\ & \vec{z} \geq 0. \end{array}$$

Thus, such a Δb must satisfy

$$\|\Delta b\|_1 \geq \min\left\{\vec{c}^{\mathrm{T}}\vec{x} - \frac{\mu(w)}{4} + \frac{\rho_P(A)}{2}, \frac{1}{2}\right\} \geq \frac{\min\{(1-\beta)m\mu(w) + \rho_P(A), 1\}}{2}, \tag{6.8}$$

the last inequality by Lemma 5.3.

Putting (6.7) and (6.8) together, we get

$$\sigma_{\min}(\mu(w)^{1/2}B) \geq \frac{\min\{(1-\beta)m\mu + \rho_P(A), 1\}(1-2\beta)}{2\sqrt{2(m+n)}(2(m+n) + \sqrt{2(m+n)}\beta + \epsilon)}.$$

Since this holds for any $\epsilon > 0$, we get the bound on $\sigma_{\min}(\mu(w)^{1/2}B)$.

Now the bound on $\sigma_{\max}(\mu(w)^{1/2}B) = \mu(w)^{1/2}\|B\|$:

$$\|\mu(w)^{1/2}B\| = \|\mu(w)^{1/2}\vec{X}^{1/2}\vec{S}^{-1/2}\mathcal{A}\| \leq \mu(w)^{1/2}\,\|\vec{X}^{1/2}\vec{S}^{-1/2}\|\,\|\mathcal{A}\|.$$

Because $w \in \overline{\mathcal{N}_\beta}$, for each $i = 1, \ldots, 2(m+n)$,

$$x_i^{1/2}s_i^{-1/2} \leq \frac{x_i}{(1-2\beta)\mu(w)^{1/2}}.$$

Thus

$$\|\vec{X}^{1/2}\vec{S}^{-1/2}\| = \max\{x_i^{1/2}s_i^{-1/2}\} \leq \frac{1}{(1-2\beta)\mu(w)^{1/2}} \max\{x_i\}.$$

But by Claim 1, $\|\vec{x}\|_\infty \leq \frac{3}{2}\max\{1, 2(m+n)\mu(w)\}$; and by Assumption 1, $\|\mathcal{A}\| \leq \sqrt{n}$. Consequently,

$$\|\mu(w)^{1/2}B\| \leq \mu(w)^{1/2}\|\vec{X}^{1/2}\vec{S}^{-1/2}\|\,\|\mathcal{A}\| \leq \frac{3\max\{2(m+n)\mu(w), 1\}\sqrt{n}}{2(1-2\beta)}. \qquad \square$$

6.2. Proof of Proposition 6.1. We will prove the following bound first:

$$\sigma_{\min}\left(\mu(w)^{1/2}\begin{bmatrix} X^{1/2}S^{-1/2}A^{\mathrm{T}} \\ (X')^{1/2}(S')^{-1/2} \\ (X'')^{1/2}(S'')^{-1/2} \end{bmatrix}\right) \leq \frac{\rho_P(A)}{5(m+n)^{3/2}}. \tag{6.9}$$

To see this, proceed again as in the first part of the proof of Proposition 6.2: by Proposition 5.2, for any $\Delta b^I \in \mathbb{R}^m$ with $\|\Delta b^I\|_1 < \rho_P(A)$ there exists $\vec{z}$ such that

$$\mathcal{A}\vec{z} = \vec{b} + \begin{bmatrix} \Delta b^I \\ 0 \end{bmatrix}, \quad \vec{z} \geq 0, \quad \text{and } \vec{c}^{\mathrm{T}}\vec{z} \leq \vec{c}^{\mathrm{T}}\vec{x} - \frac{\mu(w)}{4}.$$

On the other hand, given $\epsilon > 0$, Lemma 6.3 implies that there exists $\Delta b = \left[\begin{smallmatrix} \Delta b^I \\ 0 \end{smallmatrix}\right]$ satisfying

$$\|\Delta b^I\| \leq \frac{2(m+n) + \sqrt{2(m+n)}\beta + \epsilon}{1-2\beta}\,\sigma_{\min}\left(\mu(w)^{1/2}\begin{bmatrix} X^{1/2}S^{-1/2}A^{\mathrm{T}} \\ (X')^{1/2}(S')^{-1/2} \\ (X'')^{1/2}(S'')^{-1/2} \end{bmatrix}\right) \tag{6.10}$$

and such that the optimal value of the following problem exceeds $\vec{c}^{\mathrm{T}}\vec{x} - \frac{\mu(w)}{4}$:

$$\begin{array}{ll} \min & \vec{c}^{\mathrm{T}}\vec{z} \\ \text{s.t.} & \mathcal{A}\vec{z} = \vec{b} + \Delta b, \\ & \vec{z} \geq 0. \end{array}$$

Thus, such Δb^I must satisfy

$$\|\Delta b^I\|_1 \geq \rho_P(A). \tag{6.11}$$

Putting (6.10) and (6.11) together, we get

$$\sigma_{\min}\left(\mu(w)^{1/2}\begin{bmatrix} X^{1/2}S^{-1/2}A^{\mathrm{T}} \\ (X')^{1/2}(S')^{-1/2} \\ (X'')^{1/2}(S'')^{-1/2} \end{bmatrix}\right) \geq \frac{\rho_P(A)(1-2\beta)}{\sqrt{2(m+n)}(2(m+n)+\sqrt{2(m+n)}\beta+\epsilon)}.$$

Since this holds for any $\epsilon > 0$, we get (6.9).

To finish, notice that

$$\begin{aligned}\sigma_{\min}\left(\begin{bmatrix} X^{1/2}S^{-1/2}A^{\mathrm{T}} \\ (X')^{1/2}(S')^{-1/2} \\ (X'')^{1/2}(S'')^{-1/2} \end{bmatrix}\right) &\leq \sigma_{\min}(X^{1/2}S^{-1/2}A^{\mathrm{T}}) + \|(X')^{1/2}(S')^{-1/2}\| \\ &\quad + \|(X'')^{1/2}(S'')^{-1/2}\|,\end{aligned}$$

and because $w \in \overline{\mathcal{N}_\beta}$,

$$(x'_j)^{1/2}(s'_j)^{-1/2} \leq \frac{x'_j}{(1-2\beta)\mu(w)^{1/2}} \leq \frac{2(m+n)\mu(w)^{1/2}}{1-2\beta},$$

and

$$(x_j'')^{1/2}(s_j'')^{-1/2} \leq \frac{x_j''}{(1-2\beta)\mu(w)^{1/2}} \leq \frac{2(m+n)\mu(w)^{1/2}}{1-2\beta}.$$

Here we used $x_j',\ x_j'' \leq \vec{c}^{\,\mathrm{T}}\vec{x} \leq 2(m+n)\mu(w)$.

Therefore

$$\begin{aligned}\sigma_{\min}(X^{1/2}S^{-1/2}A^{\mathrm{T}}) &\geq \frac{\rho_P(A)(1-2\beta)}{5(m+n)^{3/2}\mu(w)^{1/2}} - \frac{4(m+n)\mu(w)^{1/2}}{1-2\beta} \\ &\geq \frac{4(m+n)\mu(w)^{1/2}}{(1-2\beta)\gamma}.\end{aligned}$$

The second inequality follows from the hypothesis

$$\rho_P(A) \geq \frac{20(m+n)^{5/2}\mu(w)}{(1-2\beta)^2}\left(1+\frac{1}{\gamma}\right). \qquad \square$$

7. Floating-point numbers, floating-point arithmetic. In this section we recall the basics of a floating-point arithmetic which idealizes the usual IEEE standard arithmetic. This system is defined by a set $\mathbb{F} \subset \mathbb{R}$ containing 0 (the *floating-point numbers*), a transformation $r : \mathbb{R} \to \mathbb{F}$ (the *rounding map*), and a constant $u \in \mathbb{R}$ (the *round-off unit*) satisfying $0 < u < 1$. The properties we require for such a system are the following:

(i) For any $x \in \mathbb{F}$, $r(x) = x$. In particular, $r(0) = 0$.

(ii) For any $x \in \mathbb{R}$, $r(x) = x(1+\delta)$ with $|\delta| \leq u$.

We also define on $\mathbb{F}$ arithmetic operations following the classical scheme

$$x \widetilde{\circ} y = r(x \circ y)$$

for any $x, y \in \mathbb{F}$ and $\circ \in \{+, -, \times, /\}$, so that

$$\widetilde{\circ} : \mathbb{F} \times \mathbb{F} \to \mathbb{F}.$$

Fundamental example. The classical floating-point numbers satisfy all these properties (see [23, 10]). Let us recall their definition. Let $\beta, t \in \mathbb{N}$ be given with $\beta \geq 2$ (the base) and $t \geq 1$ (the precision). The floating-point number set $\mathbb{F}$ is given by the numbers with the form

$$y = \pm\beta^e\left(\frac{d_1}{\beta} + \frac{d_2}{\beta^2} + \cdots + \frac{d_t}{\beta^t}\right),$$

with $e \in \mathbb{Z}$, $d_i \in \{0, 1, \ldots, \beta-1\}$, and $d_1 \neq 0$. The rounding map r associates to any $x \in \mathbb{R}$ the element of $\mathbb{F}$ nearest to x (or one of them when x is equidistant from two floating-point numbers). We may take here

$$u = \frac{1}{2}\beta^{1-t}.$$

This is a consequence of the distribution of floating-point numbers: in the interval $[\beta^e, \beta^{e+1}]$ they are equally spaced with space $2\beta^e u$. Thus, for $x \in [\beta^e, \beta^{e+1}]$, the distance between x and $r(x)$ is at most $2\beta^e u/2 \leq |x|u$, and property (ii) above holds.

Remark 7.1. In "real life floating-point arithmetic" a limitation is given on the exponent, $e_{\min} \le e \le e_{\max}$, and consequently there are a smallest and a largest positive floating-point numbers $\min_{\mathbb{F}}$ and $\max_{\mathbb{F}}$, respectively. Associated to these numbers are the concepts of underflow and overflow. To avoid the difficulties associated with under- and overflow we take, as an admissible exponent, any integer $e \in \mathbb{Z}$.

The following is an immediate consequence of property (ii) above.

PROPOSITION 7.1. *For any $x, y \in \mathbb{F}$ we have*

$$x \tilde{\circ} y = (x \circ y)(1+\delta), \qquad |\delta| \le u. \qquad \square$$

When combining many operations in floating-point arithmetic, quantities such as $\prod_{i=1}^{n}(1+\delta_i)^{\rho_i}$ naturally appear. The proof of the following propositions can be found in Chapter 3 of [10]. The notation they introduce, the quantities γ_n and θ_n, and the relations showed therein, will be widely used in our round-off analysis.

PROPOSITION 7.2. *If $|\delta_i| \le u$, $\rho_i \in \{-1, 1\}$, and $nu < 1$, then*

$$\prod_{i=1}^{n}(1+\delta_i)^{\rho_i} = 1 + \theta_n,$$

where

$$|\theta_n| \le \gamma_n = \frac{nu}{1-nu}. \qquad \square$$

PROPOSITION 7.3. *For any positive integer k such that $ku < 1$, let θ_k be any quantity satisfying*

$$|\theta_k| \le \gamma_k = \frac{ku}{1-ku}.$$

The following relations hold.

1. $(1+\theta_k)(1+\theta_j) = 1 + \theta_{k+j}$.
2. $$\frac{1+\theta_k}{1+\theta_j} = \begin{cases} 1+\theta_{k+j} & \text{if } j \le k, \\ 1+\theta_{k+2j} & \text{if } j > k. \end{cases}$$
3. *If $ku, ju \le 1/2$, then $\gamma_k \gamma_j \le \gamma_{\min\{k,j\}}$.*
4. $i\gamma_k \le \gamma_{ik}$.
5. $\gamma_k + u \le \gamma_{k+1}$.
6. $\gamma_k + \gamma_j + \gamma_k\gamma_j \le \gamma_{k+j}$. $\square$

When computing an arithmetic expression q with a round-off algorithm, errors will accumulate and we will obtain another quantity which we will denote by $\mathtt{fl}(q)$. We will also write $\mathrm{Error}(q) = |q - \mathtt{fl}(q)|$.

An example of round-off analysis which will be useful in what follows is given in the next proposition, whose proof can be found in section 3.1 of [10].

PROPOSITION 7.4. *There is a round-off algorithm which, with input $x, y \in \mathbb{R}^n$, computes the dot product of x and y. The computed value $\mathtt{fl}(\langle x, y\rangle)$ satisfies*

$$\mathtt{fl}(\langle x, y\rangle) = \langle x, y\rangle + \theta_{\lceil \log_2 n \rceil + 1}\langle |x|, |y|\rangle,$$

where $|x| = (|x_1|, \ldots, |x_n|)$. In particular, if $x = y$, the algorithm computes $\mathtt{fl}(\|x\|^2)$ satisfying

$$\mathtt{fl}(\|x\|^2) = \|x\|^2(1 + \theta_{\lceil \log_2 n \rceil + 1}). \qquad \square$$

In the next section we will have to deal with square roots. The following result will help us to do so.

LEMMA 7.5. *Let $\theta \in \mathbb{R}$ such that $|\theta| \leq 1/2$. Then, $\sqrt{1+\theta} = 1+\theta'$ with $|\theta'| \leq |\theta|$.*

Proof. By the intermediate value theorem we have that $\sqrt{1+\theta} - 1 = |\theta|(\sqrt{\xi})'$ with $\xi \in (1-|\theta|, 1)$ (if $\theta < 0$; $\xi \in (1, 1+\theta)$ otherwise). But

$$\left|(\sqrt{\xi})'\right| = \left|\frac{1}{2\sqrt{\xi}}\right| \leq \frac{1}{\sqrt{2}},$$

the last since $|\xi| \geq 1/2$. ∎

From Lemma 7.5 it follows that

$$\mathtt{fl}\left(\sqrt{a(1+\theta_k)}\right) = \sqrt{a}(1+\theta_{k+1}).$$

We will use this bound often in the next section.

To avoid the accumulation of cumbersome notation and of uninteresting constants, we will denote by $\operatorname{Lg} n$ any expression of the form $a + \lceil \log_2 n \rceil$, where a is a constant independent of n. Our choice of $u = \phi(\mu(w))$ guarantees that $ku < 1/2$ holds whenever we encounter θ_k, and consequently, $\theta_k \leq 2ku$. We will therefore not bother the reader by repeating this fact each time we use it.

A first application of Proposition 7.4 is in proving Propositions 3.6 and 3.8.

Proof of Proposition 3.6. By our choice of u, $u\theta_{\lceil \log_2 m \rceil + 1} \leq 2u(\lceil \log_2 m \rceil + 1)$. On the other hand, by Proposition 7.4 and Assumption 1, for $i = 1, \ldots, n$,

$$\operatorname{Error}(A_i^{\mathrm{T}} \cdot y) \leq \langle |A_i^{\mathrm{T}}|, |y| \rangle \gamma_{\lceil \log_2 m \rceil + 1} \leq \|y\|_\infty \gamma_{\lceil \log_2 m \rceil + 1} \leq 2u(\lceil \log_2 m \rceil + 1).$$

Hence $\mathtt{fl}(A^{\mathrm{T}} y) < -2u(\lceil \log_2 m \rceil + 1) \Rightarrow A^{\mathrm{T}} y < 0$. ∎

Proof of Proposition 3.8. If $\mu(w) \leq \frac{\rho_D(A)}{20(n+m)^2}$, then Corollary 5.5 yields

$$A^{\mathrm{T}} y < -\frac{n\mu(w)}{10(m+n)} e \leq -4u(\lceil \log_2 m \rceil + 1)e$$

(the last by our choice of u).

Thus $\mathtt{fl}(A^{\mathrm{T}} y) < -2u(\lceil \log_2 m \rceil + 1)e$ because $\operatorname{Error}(A_i^{\mathrm{T}} \cdot y) \leq 2u(\lceil \log_2 m \rceil + 1)$. Hence the algorithm halts at step (iii). ∎

Proof of Proposition 3.5. This proof is an easy verification. ∎

8. Proof of Proposition 4.1.

8.1. A numerically stable method to solve the reduced system. The system of equations (3.2) is equivalent to the least squares problem

$$\min_{v \in \mathbb{R}^m} \|Bv + g\|,$$

where

$$B = \vec{X}^{1/2} \vec{S}^{-1/2} \mathcal{A}^{\mathrm{T}} \text{ and } g = \vec{X}^{-1/2} \vec{S}^{-1/2} (\vec{X}\vec{S}e - \overline{\mu} e). \tag{8.1}$$

Hence we can apply techniques for least squares problems in order to compute a solution to (3.2).

Algorithms for linear least squares problems have been extensively studied in the literature (see, e.g., [2, 7, 9, 11]). For our purposes, we shall use *Golub's method*. This

algorithm is based on computing the QR factorization of the matrix B via Givens rotations. For details see [2, section 2.4.1] or [7, section 3.4].

In order to compute a solution to (3.2), we proceed as follows. First, form B and g as in (8.1). Then apply Golub's method to find a solution $\Delta\vec{y}$ for the least squares problem

$$\min_v \|Bv + g\|.$$

To study the quality of the solution obtained in this fashion, we will rely on the following backward stability property for Golub's method. (See [11, Chapter 16] for details.)

PROPOSITION 8.1. *Let $\mathcal{B} \in \mathbb{R}^{p\times q}$ $(p \geq q)$ have full rank. Let u denote the machine precision. If Golub's method is applied to*

$$\min_{v\in\mathbb{R}^q} \|\mathcal{B}v + f\|,$$

then the computed solution is the exact solution to a problem

$$\min_{v\in\mathbb{R}^q} \|(\mathcal{B} + \delta\mathcal{B})v + (f + \delta f)\|,$$

where

$$\|\delta\mathcal{B}\| \leq cupq^{3/2}\|\mathcal{B}\|, \quad \|\delta f\| \leq cupq\|f\|,$$

and c is a universal constant independent of p, q. ∎

We have already bounded the smallest and largest singular values of B in Proposition 6.2. We will also need the following bound on the vector g.

LEMMA 8.2. *Let g be defined as in* (8.1). *Then*

$$\|g\| \leq \mu(w)^{1/2}.$$

Proof.

$$\|g\| = \|\vec{X}^{-1/2}\vec{S}^{-1/2}\,(\vec{X}\vec{S}e - \overline{\mu}e)\| \leq \|\vec{X}^{-1/2}\vec{S}^{-1/2}\|\|\vec{X}\vec{S}e - \overline{\mu}e\|.$$

But, for $i = 1, \ldots, 2(n+m)$, $\vec{x}_i\vec{s}_i \geq (1-2\beta)\mu(w)$ by Lemma 5.3, so

$$\left\|\vec{X}^{-1/2}\vec{S}^{-1/2}\right\| = \frac{1}{\min\limits_{i\leq 2(n+m)} |\vec{x}_i\vec{s}_i|^{1/2}} \leq \frac{1}{(1-2\beta)^{1/2}\mu(w)^{1/2}}.$$

Also,

$$\|\vec{X}\vec{S}e - \overline{\mu}e\| \leq \|\vec{X}\vec{S}e - \mu(w)e\| + \|\mu(w)e - \overline{\mu}e\| \leq (\sqrt{2}\beta + \xi)\mu(w).$$

The last inequality follows from $w \in \overline{\mathcal{N}_\beta}$ and our definition of $\overline{\mu}$. Multiplying both bounds and recalling that $\beta, \xi \in (0, 1/4]$, the lemma follows. ∎

8.2. Proof of Proposition 4.1. We need to prove that the computed $\Delta\vec{y}$ satisfies

$$\|\mathcal{A}\vec{S}^{-1}\vec{X}\mathcal{A}^{\mathrm{T}}\Delta\vec{y} + \mathcal{A}\vec{S}^{-1}(\vec{X}\vec{S}e - \overline{\mu}(w)e)\| = \|B^{\mathrm{T}}B\Delta\vec{y} + B^{\mathrm{T}}g\| \leq \frac{\xi\,\min\{1/(m+n), \mu(w)\}}{7}.$$

Let $\mathcal{B} = \mathtt{fl}(B)$ and $f = \mathtt{fl}(g)$. By Proposition 8.1, the computed $\Delta\vec{y}$ is the exact solution of

$$\min \|\widetilde{\mathcal{B}}\Delta\vec{y} + \widetilde{f}\|$$

for some $\widetilde{\mathcal{B}}$ and $\widetilde{f}$ satisfying

$$\|\widetilde{\mathcal{B}} - \mathcal{B}\| \le cu(m+n)^{5/2}\|\mathcal{B}\|, \quad \|\widetilde{f} - f\| \le cu(m+n)^2\|f\|,$$

where c is a universal constant. Thus, $\widetilde{\mathcal{B}} = B + \delta B$ with $\delta B = (\mathtt{fl}(B) - B) + (\widetilde{\mathcal{B}} - \mathcal{B})$ and, by the triangle inequality, $\|\delta B\| \le \|\mathtt{fl}(B) - B\| + \|\widetilde{\mathcal{B}} - \mathcal{B}\|$. In this sum, the second term dominates since B is a diagonal matrix, and a straightforward use of Propositions 7.2, 7.3, and Lemma 7.5 yields $\|B - \mathtt{fl}(B)\| \le \|B\|\,\theta_4$. Therefore,

$$\|\delta B\| \le cu(m+n)^{5/2}\|B\| \tag{8.2}$$

for some (slightly larger) universal constant c. A similar argument shows that $\widetilde{f} = g + \delta g$, with

$$\|\delta g\| \le cu(m+n)^2\|g\|, \tag{8.3}$$

for some universal constant c (which we assume to be the same as above).

Thus,

$$\begin{aligned} B^{\mathrm{T}}B\Delta\vec{y} + B^{\mathrm{T}}g &= \widetilde{\mathcal{B}}^{\mathrm{T}}\widetilde{\mathcal{B}}\Delta\vec{y} + \widetilde{\mathcal{B}}^{\mathrm{T}}\widetilde{f} \\ &\quad - (B^{\mathrm{T}}\delta B + \delta B^{\mathrm{T}}B + \delta B^{\mathrm{T}}\delta B)\Delta\vec{y} - B^{\mathrm{T}}\delta g - \delta B^{\mathrm{T}}g \\ &= -(B^{\mathrm{T}}\delta B + \delta B^{\mathrm{T}}B + \delta B^{\mathrm{T}}\delta B)\Delta\vec{y} - B^{\mathrm{T}}\delta g - \delta B^{\mathrm{T}}g. \end{aligned}$$

In the last step we used that, since $\|\widetilde{\mathcal{B}}\Delta\vec{y} + \widetilde{f}\| = \min_v \|\widetilde{\mathcal{B}}v + \widetilde{f}\|$, $\widetilde{\mathcal{B}}^{\mathrm{T}}\widetilde{\mathcal{B}}\Delta\vec{y} + \widetilde{\mathcal{B}}^{\mathrm{T}}\widetilde{f} = 0$.

Therefore, to finish the proof it suffices to show that

$$\big\|(B^{\mathrm{T}}\delta B + \delta B^{\mathrm{T}}B + \delta B^{\mathrm{T}}\delta B)\Delta\vec{y} + B^{\mathrm{T}}\delta g + \delta B^{\mathrm{T}}g\big\| \le \frac{\xi\,\min\{1/(m+n), \mu(w)\}}{7}. \tag{8.4}$$

By (8.2) and (8.3), the left-hand side of (8.4) is bounded above by

$$\begin{aligned} &(2cu(m+n)^{5/2} + c^2u^2(m+n)^5)\|B\|^2\,\|\Delta\vec{y}\| + 2cu(m+n)^{5/2}\|B\|\,\|g\| \\ &\quad \le cu(m+n)^{5/2}\|B\|\big((2 + cu(m+n)^{5/2})\|B\|\,\|\Delta\vec{y}\| + 2\|g\|\big). \end{aligned}$$

Since $u \le \frac{1}{\mathbf{c}(m+n)^{12}}$, by appropriately choosing $\mathbf{c}$ we have $cu(m+n)^{5/2} \le 1$, and therefore the above quantity is no larger than

$$cu(m+n)^{5/2}\|B\|(3\|B\|\,\|\Delta\vec{y}\| + 2\|g\|).$$

Hence we get (8.4) if

$$cu(m+n)^{5/2}\|B\|(3\|B\|\,\|\Delta\vec{y}\| + 2\|g\|) \le \frac{\xi\,\min\{1/(m+n), \mu(w)\}}{7}. \tag{8.5}$$

Since $\|\widetilde{\mathcal{B}}\Delta\vec{y} + \widetilde{f}\| = \min_v \|\widetilde{\mathcal{B}}v + \widetilde{f}\|$,

$$\|\Delta\vec{y}\| \le \frac{\|\widetilde{f}\|}{\sigma_{\min}(\widetilde{\mathcal{B}})}. \tag{8.6}$$

But by Proposition 6.2

$$\kappa(B) \le 60\sqrt{n}(m+n)^{3/2} \max\left\{\frac{1}{m\mu(w)}, 2(m+n)\mu(w)\right\}.$$

Hence our choice of $u = \min\{1, \mu(w)^2\}\frac{1}{\mathbf{c}(m+n)^{12}}$ (with appropriately large $\mathbf{c}$) ensures that

$$cu(m+n)^{5/2}\kappa(B) \le 1/2. \tag{8.7}$$

Thus, using (8.2),

$$\begin{aligned}\sigma_{\min}(\widetilde{\mathcal{B}}) &\ge \sigma_{\min}(B) - \|\delta B\| \\ &\ge \sigma_{\min}(B) - cu(m+n)^{5/2}\|B\| \\ &= \sigma_{\min}(B)(1 - cu(m+n)^{5/2}\kappa(B)) \ge \frac{\sigma_{\min}(B)}{2}.\end{aligned}$$

Now using (8.3), $\|\widetilde{f}\| \le (1 + cu(m+n)^2)\|g\|$. Replacing the bounds for $\sigma_{\min}(\widetilde{B})$ and $\|\widetilde{f}\|$ in (8.6), we obtain

$$\|\Delta\vec{y}\| \le \frac{2(1 + cu(m+n)^2)\|g\|}{\sigma_{\min}(B)}.$$

Consequently,

$$\begin{aligned}&cu(m+n)^{5/2}\|B\|(3\|B\|\,\|\Delta\vec{y}\| + 2\|g\|) \\ &\qquad \le cu(m+n)^{5/2}\|B\|\,\|g\|\,\big(6(1 + cu(m+n)^2)\kappa(B) + 2\big) \\ &\qquad \le cu(m+n)^{5/2}\|\mu(w)^{1/2}B\|\,\big(6(1 + cu(m+n)^2)\kappa(B) + 2\big) \\ &\qquad \le 9cu(m+n)^{5/2}\|\mu(w)^{1/2}B\|\,\kappa(B).\end{aligned}$$

The last two lines follow from Lemma 8.2 and the condition $1 + cu(m+n)^2 \le 7/6$, that holds by our choice of u.

Finally, Proposition 6.2 and our choice of $u = \min\{1, \mu(w)^2\}\frac{1}{\mathbf{c}(m+n)^{12}}$ (with appropriately large $\mathbf{c}$) ensure that this last expression is bounded by

$$\xi\,\min\{1/(m+n), \mu(w)\}/7,$$

thus proving (8.5) as we needed. ∎

9. Proof of Propositions 3.7 and 3.9.

9.1. Proof of Proposition 3.7. Let $D = X^{1/2}S^{-1/2}A^{\mathrm{T}}$. By Proposition 5.6, it suffices to show that if $\mathtt{fl}(\sigma_{\min}(D)) > \frac{3(m+n)\mu^{1/2}}{(1-2\beta)\gamma}$, then $\sigma_{\min}(D) \ge \frac{2(m+n)\mu^{1/2}}{(1-2\beta)\gamma}$.

We first estimate the errors produced in step (iv). A straightforward use of Propositions 7.2, 7.3 and Lemma 7.5 shows that $E' = D - \mathtt{fl}(D)$ satisfies

$$\|E'\| \le \|D\|\theta_4.$$

Now, let $\mathcal{D} = \mathtt{fl}(D)$. Let us assume that we compute $\sigma_{\min}(\mathcal{D})$ using a backward stable algorithm (e.g., QR factorization). Then the computed $\mathtt{fl}(\sigma_{\min}(\mathcal{D}))$ is the exact $\sigma_{\min}(\mathcal{D} + E'')$ for a matrix E'' with

$$\|E''\| \le cn^2u\|\mathcal{D}\|$$

for some universal constant c (see, e.g., [9, 10]). Thus,

$$\mathtt{fl}(\sigma_{\min}(D)) = \mathtt{fl}(\sigma_{\min}(\mathcal{D})) = \sigma_{\min}(\mathcal{D} + E'') = \sigma_{\min}(D + E' + E'')$$

and, letting $E = E' + E''$,

$$\|E\| \leq \|E'\| + \|E''\| \leq \|D\|(\theta_4 + cn^2u(1+\theta_4)).$$

Since $w \in \overline{\mathcal{N}_\beta}$, $\|X^{1/2}S^{-1/2}\| = \max\{x_i^{1/2}s_i^{-1/2}\} \leq \frac{\max\{x_i\}}{(1-2\beta)\mu^{1/2}} \leq \frac{1}{(1-2\beta)\mu^{1/2}}$, and by Assumption 1, $\|A^{\mathrm{T}}\| \leq \sqrt{n}$. Consequently,

$$\|D\| \leq \|X^{1/2}S^{-1/2}\|\,\|A^{\mathrm{T}}\| \leq \frac{\sqrt{n}}{(1-2\beta)\mu^{1/2}}.$$

In particular,

$$\|E\| \leq \|D\|(\theta_4 + cn^2u(1+\theta_4)) \leq \frac{(m+n)\mu^{1/2}}{(1-2\beta)\gamma}.$$

Therefore,

$$\mathtt{fl}(\sigma_{\min}(D)) = \sigma_{\min}(D+E) \leq \sigma_{\min}(D) + \|E\| < \sigma_{\min}(D) + \frac{(m+n)\mu^{1/2}}{(1-2\beta)\gamma}.$$

From here it follows that if $\mathtt{fl}(\sigma_{\min}(D)) > \frac{3(m+n)\mu^{1/2}}{(1-2\beta)\gamma}$, then $\sigma_{\min}(D) \geq \frac{2(m+n)\mu^{1/2}}{(1-2\beta)\gamma}$, as we needed to show. ∎

9.2. Proof of Proposition 3.9. As before,

$$\mathtt{fl}(\sigma_{\min}(D)) = \sigma_{\min}(D+E) \geq \sigma_{\min}(D) - \|E\| > \sigma_{\min}(D) - \frac{(m+n)\mu^{1/2}}{(1-\beta)\gamma}.$$

Therefore, if $\sigma_{\min}(D) \geq \frac{4(m+n)\mu^{1/2}}{(1-\beta)\gamma}$, then $\mathtt{fl}(\sigma_{\min}(D)) > \frac{3(m+n)\mu^{1/2}}{(1-\beta)\gamma}$.

We now apply Proposition 6.1 to deduce that, since

$$\mu \leq \frac{(1-2\beta)^2\rho_P(A)}{20(m+n)^{5/2}}\left(1+\frac{1}{\gamma}\right)^{-1},$$

we indeed have $\sigma_{\min}(D) \geq \frac{4(m+n)\mu^{1/2}}{(1-\beta)\gamma}$, and so $\mathtt{fl}(\sigma_{\min}(D)) > \frac{3(m+n)\mu^{1/2}}{(1-\beta)\gamma}$. ∎

10. Proof of Proposition 3.4. The arguments in this section are modifications of the proofs in [12] and [25]. Although they are a bit laborious, they do not really require any particularly novel insight.

10.1. Some preliminary lemmas. Let $\widetilde{w} = w - \Delta\underline{w} = w - (\underline{w} - \underline{w^+})$. Notice that $\underline{\widetilde{w}} = \underline{w^+}$. We will first prove some lemmas regarding the point $\widetilde{w}$.

LEMMA 10.1. *The point $\widetilde{w}$ defined above satisfies*

$$|\mu(\widetilde{w}) - \overline{\mu}| \leq \frac{\xi\mu(w)}{3\sqrt{2(m+n)}}.$$

Proof. By Theorem 3.3,

$$\begin{aligned} \widetilde{X}\widetilde{S}e &= \vec{X}\vec{S}e - \vec{X}\Delta\underline{\vec{s}} - \vec{S}\Delta\underline{\vec{x}} + \Delta\underline{\vec{X}}\Delta\underline{\vec{S}}e \\ &= \Delta\underline{\vec{X}}\Delta\underline{\vec{S}}e + \overline{\mu}e - r \end{aligned} \tag{10.1}$$

for some r satisfying $\|r\| \le \frac{\xi\mu(w)}{3}$. In addition,

$$\begin{aligned} \mathcal{A}\Delta\underline{\vec{x}} &= 0, \\ \mathcal{A}^{\mathrm{T}}\Delta\vec{y} + \Delta\underline{\vec{s}} &= 0 \end{aligned}$$

implies $e^{\mathrm{T}}\Delta\underline{\vec{X}}\Delta\underline{\vec{S}}e = 0$. Thus $e^{\mathrm{T}}\widetilde{X}\widetilde{S}e = 2(m+n)\overline{\mu} - e^{\mathrm{T}}r$, and therefore

$$|\mu(\widetilde{w}) - \overline{\mu}| = \frac{|e^{\mathrm{T}}r|}{2(m+n)} \le \frac{\|r\|}{\sqrt{2(m+n)}} \le \frac{\xi\mu(w)}{3\sqrt{2(m+n)}}. \qquad \square$$

LEMMA 10.2. *The point $\widetilde{w}$ defined above satisfies*

$$\|\widetilde{X}\widetilde{S}e - \mu(\widetilde{w})e\| \le \left(\left(1 - \frac{3\xi}{2\sqrt{2(m+n)}}\right)\beta - \frac{\xi}{6}\right)\mu(w)$$

and $\widetilde{x}, \widetilde{s} > 0$.

To prove Lemma 10.2 we shall rely on the following technical lemma, which is the same as in [12, Lemma 4.1], further strengthened in [25, Lemma 5.3].

LEMMA 10.3. *Assume that $u, v \in \mathbb{R}^d$ are such that $u^{\mathrm{T}}v \ge 0$. Then*

$$\|UVe\| \le \frac{\|u+v\|^2}{2\sqrt{2}}. \qquad \square$$

Proof of Lemma 10.2. By (10.1) in the proof of Lemma 10.1,

$$\|\widetilde{X}\widetilde{S}e - \mu(\widetilde{w})e\| \le \|\Delta\underline{\vec{X}}\Delta\underline{\vec{S}}e\| + \|r\| + |\mu(\widetilde{w}) - \overline{\mu}|\sqrt{2(m+n)} \le \|\Delta\underline{\vec{X}}\Delta\underline{\vec{S}}e\| + \frac{2\xi\mu(w)}{3}.$$

Thus, for the first inequality, it suffices to show that

$$\left\|\Delta\underline{\vec{X}}\Delta\underline{\vec{S}}e\right\| \le \left(\left(1 - \frac{3\xi}{2\sqrt{2(m+n)}}\right)\beta - \frac{5\xi}{6}\right)\mu(w). \tag{10.2}$$

For this, let $\vec{D} = \vec{X}^{-1/2}\vec{S}^{1/2}$, and apply Lemma 10.3 to

$$u = \vec{D}\Delta\underline{\vec{x}}, \quad v = \vec{D}^{-1}\Delta\underline{\vec{s}}$$

to get

$$\begin{aligned} \|\Delta\underline{\vec{X}}\Delta\underline{\vec{S}}e\| &= \|UVe\| \\ &\le \tfrac{1}{2\sqrt{2}}\|D\Delta\underline{\vec{x}} + D^{-1}\Delta\underline{\vec{s}}\|^2 \\ &= \tfrac{1}{2\sqrt{2}}\|\vec{X}^{-1/2}\vec{S}^{-1/2}(\vec{X}\vec{S}e - \overline{\mu}\,e + r)\|^2 \\ &\le \tfrac{1}{2\sqrt{2}\min\{x_i s_i\}}\|\vec{X}\vec{S}e - \overline{\mu}\,e + r\|^2. \end{aligned}$$

However,

$$\begin{aligned} \|\vec{X}\vec{S}e - \overline{\mu}\,e\|^2 &= \|\vec{X}\vec{S}e - \mu(w)e + (\mu(w) - \overline{\mu})e\|^2 \\ &= \|\vec{X}\vec{S}e - \mu(w)e\|^2 + 2(m+n)|\overline{\mu} - \mu(w)|^2 \\ &\le \|\vec{X}\vec{S}e - \mu(w)e\|^2 + \xi^2\mu(w)^2, \end{aligned}$$

the second line by the definition of $\overline{\mu}$. From Claim 1 it easily follows that $\|\vec{X}\,\vec{S}e - \mu(w)e\|^2 \leq 2\beta^2\mu(w)^2$, and thus

$$\begin{aligned}\|\vec{X}\,\vec{S}e - \overline{\mu}\,e + r\|^2 &\leq \|\vec{X}\,\vec{S}e - \overline{\mu}\,e\|^2 + 2\|\vec{X}\,\vec{S}e - \overline{\mu}\,e\|\,\|r\| + \|r\|^2\\ &\leq 2(\beta+\xi)^2\mu(w)^2.\end{aligned}$$

Therefore, since, by Lemma 5.3, $\min\{x_is_i\} \geq (1-2\beta)\mu(w)$,

$$\begin{aligned}\|\Delta\underline{\vec{X}}\Delta\underline{\vec{S}}e\| &= \frac{\|\vec{X}\,\vec{S}e - \overline{\mu}\,e + r\|^2}{(2\sqrt{2}\min\{x_is_i\})}\\ &\leq \frac{(\beta+\xi)^2}{\sqrt{2}(1-2\beta)}\mu(w).\end{aligned}$$

Applying (2.9) we get (10.2).

To finish, suppose that $\widetilde{x}, \widetilde{s} > 0$ does not hold. Then for some i, we have $\widetilde{x}_i < 0$ or $\widetilde{s}_i < 0$. Since $\|\widetilde{X}\widetilde{S}e - \mu(\widetilde{w})e\| \leq \beta\mu(w)$, $\widetilde{x}_i\widetilde{s}_i \geq \mu(\widetilde{w}) - \beta\mu(w) > 0$. Hence both $\widetilde{x}_i < 0$ and $\widetilde{s}_i < 0$. Therefore $\Delta\underline{x}_i\Delta\underline{s}_i > x_is_i$. This implies that

$$\begin{aligned}(1-2\beta)\mu(w) - \frac{\xi\mu(w)}{3\sqrt{2(m+n)}} - \frac{\xi\mu(w)}{3} &\leq x_is_i + \overline{\mu} - \mu(\widetilde{w}) - \|r\|\\ &\leq \Delta\underline{x}_i\Delta\underline{s}_i + \overline{\mu} - \mu(\widetilde{w}) - r_i\\ &= \widetilde{x}_i\widetilde{s}_i - \mu(\widetilde{w})\\ &\leq \|\widetilde{X}\widetilde{S}e - \mu(\widetilde{w})e\|\\ &\leq \beta\mu(w).\end{aligned}$$

But this yields $1 < 3\beta + \frac{\xi}{3} + \frac{\xi}{3\sqrt{2(m+n)}}$, which contradicts our choice of $\beta, \xi \in (0, 1/4]$. □

LEMMA 10.4. $\underline{w^+} = \underline{\widetilde{w}} \in \mathcal{N}_\beta$.

Proof. We need to show only that

$$\|\underline{\vec{X}^+}\,\underline{\vec{S}^+}e - \mu(\underline{w^+})e\| \leq \beta\mu(\underline{w^+})$$

and $\underline{\vec{x}^+}\,,\ \underline{\vec{s}^+} > 0$.

Notice that

$$\underline{w^+} = \underline{\widetilde{w}} = \widetilde{w} + \delta w,$$

where $\delta w = \underline{w} - w$.

Since $w \in \overline{\mathcal{N}_\beta}$, $\|\delta w\|_\infty \leq \frac{\beta\min\{\mu(w),1\}}{20(m+n)^2}$. Also by Lemma 10.2, $\widetilde{x}$, $\widetilde{s} > 0$. Thus, Claim 1 and Lemma 10.1 yield

$$|\mu(\underline{w^+}) - \mu(\widetilde{w})| \leq \frac{\xi\mu(w)}{12\sqrt{2(m+n)}}.$$

Hence,

$$\begin{aligned}\mu(\underline{w^+}) &\geq \mu(\widetilde{w}) - \frac{\xi\mu(w)}{12\sqrt{2(m+n)}} \geq \overline{\mu} - \frac{\xi\mu(w)}{2\sqrt{2(m+n)}}\\ &\geq \left(1 - \frac{3\xi}{2\sqrt{2(m+n)}}\right)\mu(w), \qquad (10.3)\end{aligned}$$

the second inequality in the first line by Lemma 10.1. Therefore, by Claim 1,

$$\begin{aligned}\|\underline{\vec{X}^+}\ \underline{\vec{S}^+}e-\mu(\underline{w^+})e\| &\le \|\underline{\vec{X}^+}\ \underline{\vec{S}^+}e-\widetilde{X}\widetilde{S}e\|+\|\widetilde{X}\widetilde{S}e-\mu(\widetilde{w})e\|+\|\mu(\widetilde{w})e-\mu(\underline{w^+})e\|\\ &\le \frac{\xi\mu(w)}{12}+\|\widetilde{X}\widetilde{S}e-\mu(\widetilde{w})e\|+\frac{\xi\mu(w)}{12}.\end{aligned}$$

Hence by Lemma 10.2,

$$\begin{aligned}\|\underline{\vec{X}^+}\ \underline{\vec{S}^+}e-\mu(\underline{w^+})e\| &\le \left(\left(1-\frac{3\xi}{2\sqrt{2(m+n)}}\right)\beta-\frac{\xi}{6}\right)\mu(w)+\frac{\xi\mu(w)}{6}\\ &\le \left(1-\frac{3\xi}{2\sqrt{2(m+n)}}\right)\beta\mu(w)\le\beta\mu(\underline{w^+}),\end{aligned}$$

the last inequality by (10.3).

It only remains to show that $\underline{\vec{x}^+}$, $\underline{\vec{s}^+}>0$. Again, $\|\underline{\vec{X}^+}\ \underline{\vec{S}^+}e-\mu(\underline{w^+})e\|\le\beta\mu(w^+)$ implies that, for all i, we have $\underline{x^+}_i\underline{s^+}_i\ge(1-\beta)\mu(w^+)>0$. Furthermore, by Lemma 10.2, $\widetilde{x}_i,\widetilde{s}_i>0$ for all i. Since $|\underline{x^+}_i-\widetilde{x}_i|$, $|\underline{s^+}_i-\widetilde{s}_i|<(1-\beta)\min\{1,\mu(w)\}/2$, we must have $\underline{x^+}_i$, $\underline{s^+}_i>0$. ∎

10.2. Proof of Proposition 3.1. Except for $(x^+)''-(\underline{x^+})''$, the other components in $\vec{x}^+-\underline{\vec{x}^+}$ clearly satisfy the bound. For $j=1,\dots,n$ we have, by Proposition 7.4,

$$|(x^+)''_j-(\underline{x^+})''_j|=\mathrm{Error}(A_j(x^+)+(x^+)'_j)=\big\langle|A_j|,|x^+|\big\rangle\,\theta_{\mathrm{Lg}\,n}.$$

But by Assumption 1, $\|A_j\|_\infty\le n$. Hence the proposition follows if we can show $\|x^+\|_\infty\le 1$. But this readily follows because $x^+=\underline{x^+}$ and in Lemma 10.4 we showed that $\underline{w^+}$ satisfies $\underline{x^+}>0$, which implies $\|\underline{x^+}\|_\infty\le 1$.

The bound on $\|\vec{s}^+-\underline{\vec{s}^+}\|_\infty$ follows from a similar argument. ∎

10.3. Proof of Proposition 3.4(i). Notice that

$$w^+=\widetilde{w}+\delta w,$$

where $\delta w=(w^+-\underline{w^+})+(\underline{w}-w)$. Hence

$$\mu(w^+)-\mu(\widetilde{w})=e^{\mathrm{T}}(\widetilde{X}\delta\vec{S}+\widetilde{S}\delta\vec{X}+\delta\vec{X}\delta\vec{S})e.$$

Now, Proposition 3.1 and $w\in\overline{\mathcal{N}_\beta}$ yield

$$\|\delta w\|_\infty\le\frac{\beta\min\{\mu(w),1\}}{10(m+n)^2}.$$

Applying Claim 1 and (2.10), we obtain

$$|\mu(w^+)-\mu(\widetilde{w})|\le\frac{\xi\mu(w)}{6\sqrt{2(m+n)}}.$$

Thus by Lemma 10.1

$$|\mu(w^+)-\overline{\mu}|\le\frac{\xi\mu(w)}{2\sqrt{2(m+n)}}.$$ ∎

10.4. Proof of Proposition 3.4(ii). By Lemma 10.4, $\underline{w^+} \in \mathcal{N}_\beta$; hence we only need to check that $\|w^+ - \underline{w^+}\|_\infty \leq \frac{\beta \min\{\mu(w^+),1\}}{20(m+n)^2}$. But this follows from Proposition 3.1. ∎

11. Final details and additional remarks.

11.1. In the proof of Theorem 3.3 we ignored round-off error in the computation of $\Delta x, \Delta x'$. Using Propositions 7.2 and 7.3, it is easy to show that

$$\eta := \begin{bmatrix} \Delta x \\ \Delta x' \end{bmatrix} - \mathtt{fl}\begin{bmatrix} \Delta x \\ \Delta x' \end{bmatrix}$$

satisfies

$$\|\eta\| \leq \frac{\xi \min\{1, \mu(w)\}}{60(m+n+1)}.$$

Hence, since $\|\vec{s}\|_\infty \leq 3 + 3(m+n)\mu(w)$ (by Claim 1), we have

$$\left\| \vec{S} \begin{bmatrix} \eta \\ 0 \end{bmatrix} \right\| \leq \frac{\xi \mu(w)}{10}.$$

Thus, as we explained in Remark 4.1, the proof of Theorem 3.3 in section 4 can readily be amended to make it fully rigorous.

11.2. There is an issue, left open in section 2, which still needs to be dealt with: the unit norm assumption for the rows of A^{T}. It is computationally straightforward to modify the matrix A to have this form as long as A does not have a zero column. Let us first assume that this is the case and let $\overline{A}$ be the matrix obtained by scaling each row of A^{T} by a positive number so that $\|\overline{A}_i^{\mathrm{T}}\|_1 = 1$. The following result is easy to prove.

PROPOSITION 11.1. *Assume that the matrix A is such that $\rho(A) > 0$. Then*

$$C(\overline{A}) \leq n\, C(A). \qquad ∎$$

It is immediately clear that any nontrivial solution of $\overline{A}^{\mathrm{T}} y \leq 0$ is also a nontrivial solution of $A^{\mathrm{T}} y \leq 0$. In addition we have the following easy-to-prove result.

PROPOSITION 11.2. *If $\widehat{x}$ is a γ-forward solution of $\overline{A}x = 0$, $x \geq 0$, with associated solution $\overline{x}$, then*

$$x^* = \left(\widehat{x}_1/\|A_1^{\mathrm{T}}\|_1, \ldots, \widehat{x}_n/\|A_n^{\mathrm{T}}\|_1\right)$$

is a γ-forward solution of $Ax = 0$, $x \geq 0$, with associated solution

$$\overline{x}^* = \left(\overline{x}_1/\|A_1^{\mathrm{T}}\|_1, \ldots, \overline{x}_n/\|A_n^{\mathrm{T}}\|_1\right). \qquad ∎$$

Propositions 11.1 and 11.2 allow us to extend Theorem 1.2 to the case of arbitrary matrices without zero columns. One simply modifies Algorithm `FPPD` so that the matrix A is replaced by $\overline{A}$, and a solution for either $\overline{A}x = 0$, $x \geq 0$, or for $\overline{A}^{\mathrm{T}} y \leq 0$ is found. In the first case, one returns the corresponding x^*, and in the second, the computed y.

If A has a zero column, then clearly (1.2) will not have strictly feasible solutions. In fact, in such a case $\rho_D(A) = 0$, and so A is well-posed only when $\rho_P(A) > 0$.

Theorem 1.2 still applies (normalizing the nonzero columns of A) but is of somewhat limited interest, as Algorithm FPPD can only yield solutions to (1.1). More interesting is to consider the natural *reduced* problem. Let A' be the $m \times (n-1)$ matrix obtained by removing that column from A. Consider the pair

$$A'x = 0,\ x \geq 0, \tag{11.1}$$

and

$$(A')^{\mathrm{T}} y \leq 0. \tag{11.2}$$

It is easy to see that

1. (11.1) is strictly feasible if and only if (1.1) is as well;
2. (11.2) is feasible (i.e., has nontrivial solutions) if and only if (1.2) is as well;
3. $\|A'\|_{1,\infty} = \|A\|_{1,\infty}$, $\rho_P(A') = \rho_P(A)$, and $\rho_D(A') \geq \rho_D(A)$. In particular, $C(A') \leq C(A)$, with possible strict inequality when (1.2) is feasible. (As in this case, $\rho(A) = 0$, but it could easily be the case in which $\rho(A') > 0$.)

Theorem 1.2 applied to A' yields a stronger statement than when applied to A. Both yield the same conclusion when $\rho_P(A) > 0$.

11.3. A detail ignored in our exposition is the fact that the successive values of u and $\overline{\mu}$ are computed with round-off themselves.

This is a minor issue. Note that for u, *any* value smaller than $\phi(\mu(w))$ guarantees the correctness of our analysis. Thus, it is enough to approximate $\phi(\mu(w))$ by defect. Notice that since u is likely to be 2^{-e} or 10^{-e} for some positive integer e, the approximation above will be done independently of round-off considerations.

A similar consideration applies to $\overline{\mu}$.

11.4. The issue of the representation of the matrix A deserves some words. When we write $A \in \mathbb{R}^{m \times n}$, we are assuming that the entries of A can be arbitrary real numbers. Such numbers, of course, can not be dealt with by finite-precision machines. So, they are first rounded to floating-point numbers and then given to the machine. This would be the case, for instance, if some of these entries were physical quantities either known to us (some known physical constants) or measured in nature. Such numbers would be calculated up to a certain precision or measured with such precision, respectively. Therefore, there are two instances of rounding in the whole computational process: rounding the input to fit it into the machine and the rounding taking place while operating on floating-point numbers. It is a natural assumption to suppose that both instances are done in the same way, i.e., by the same rounding map r and with the same the round-off unit u (cf. section 7).

In fixed-precision algorithms, the input is read once and its components are rounded to floating-point numbers. In Algorithm FPPD, the matrix A is read at each iteration with the precision set at the beginning of the iteration. Otherwise, if a poorly conditioned matrix A were read only once with the initial machine precision, it would be converted into a matrix A' for which the feasibility status of (1.1), (1.2) might be different. In such a case, the algorithm would yield a wrong output.

11.5. In scientific computation, fixed precision is used more commonly than variable precision. In this case, there is no guarantee that a γ-forward solution for $Ax = 0$, $x \geq 0$, or a solution for $A^{\mathrm{T}} y \leq 0$, can be found. Our development, though, can be used to estimate bounds for the condition of A (as a function of the machine precision) within which Algorithm FPPD yields a solution. We remark, however, that these

bounds are probably too pessimistic, since all of our analysis assumes that round-off errors accumulate in the worst possible way. Error propagation is, in practice, more gentle.

REFERENCES

[1] R. Bartels, *A stabilization of the simplex method*, Numer. Math., 16 (1971), pp. 414–434.
[2] Å. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.
[3] L. Blum, M. Shub, and S. Smale, *On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines*, Bull. Amer. Math. Soc., 21 (1989), pp. 1–46.
[4] D. Cheung and F. Cucker, *A new condition number for linear programming*, Math. Program., to appear.
[5] D. Cheung and F. Cucker, *Probabilistic analysis of condition numbers for linear programming*, J. Optim. Theory Appl., to appear.
[6] R. Clasen, *Techniques for automatic tolerance control in linear programming*, Comm. ACM, 9 (1966), pp. 802–803.
[7] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
[8] R. Freund, F. Jarre, and S. Mizuno, *Convergence of a class of inexact interior-point algorithms for linear programs*, Math. Oper. Res., 24 (1999), pp. 50–71.
[9] G. Golub and C. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 1989.
[10] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.
[11] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, SIAM, Philadelphia, 1995.
[12] R. Monteiro and I. Adler, *Interior path following primal-dual algorithms. Part* I: *Linear programming*, Math Programming, 44 (1989), pp. 27–41.
[13] W. Ogryczak, *The simplex method is not always well behaved*, Linear Algebra Appl., 109 (1988), pp. 41–57.
[14] J. Peña, *Understanding the geometry of infeasible perturbations of a conic linear system*, SIAM J. Optim., 10 (2000), pp. 534–550.
[15] J. Peña and J. Renegar, *Computing approximate solutions for conic systems of constraints*, Math. Program., 87 (2000), pp. 351–383.
[16] J. Renegar, *Linear programming, complexity theory and elementary functional analysis*, Math. Program., 70 (1995), pp. 279–351.
[17] J. Renegar, *Condition numbers, the barrier method, and the conjugate-gradient method*, SIAM J. Optim., 6 (1996), pp. 879–912.
[18] J. Renegar, *A Mathematical View of Interior-Point Methods in Convex Optimization*, SIAM, Philadelphia, 2000.
[19] S. Robinson, *A characterization of stability in linear programming*, Oper. Res., 25 (1977), pp. 435–447.
[20] S. Storoy, *Error control in the simplex-technique*, BIT, 7 (1967), pp. 216–225.
[21] S. Vavasis and Y. Ye, *Condition numbers for polyhedra with real number data*, Oper. Res. Lett., 17 (1995), pp. 209–214.
[22] J. Vera, *On the complexity of linear programming under finite precision arithmetic*, Math. Program., 80 (1998), pp. 91–123.
[23] J. Wilkinson, *Rounding Errors in Algebraic Processes*, Prentice–Hall, Englewood Cliffs, NJ, 1963.
[24] P. Wolfe, *Error in the solution of linear programming problems*, in Error in Digital Computation, L. Ball, ed., John Wiley, 1965, pp. 271–284.
[25] S. J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, 1997.

SIAM J. OPTIM.
Vol. 12, No. 2, pp. 555–573

A CLASS OF GLOBALLY CONVERGENT OPTIMIZATION METHODS BASED ON CONSERVATIVE CONVEX SEPARABLE APPROXIMATIONS*

KRISTER SVANBERG†

Abstract. This paper deals with a certain class of optimization methods, based on conservative convex separable approximations (CCSA), for solving inequality-constrained nonlinear programming problems. Each generated iteration point is a feasible solution with lower objective value than the previous one, and it is proved that the sequence of iteration points converges toward the set of Karush–Kuhn–Tucker points. A major advantage of CCSA methods is that they can be applied to problems with a very large number of variables (say 10^4–10^5) even if the Hessian matrices of the objective and constraint functions are dense.

Key words. nonlinear programming, constrained minimization, convex approximations, method of moving asymptotes,

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

PII. S1052623499362822

1. Introduction.

The purpose of this paper is to present and investigate a new class of optimization methods which we call *conservative convex separable approximation* (CCSA) *methods*. These methods are intended for inequality-constrained nonlinear programming problems, which are assumed to be written as *minimization* problems with *less than or equal to* constraints. There are *outer* and *inner* iterations in the methods. An *outer* iteration starts from the current iterate $x^{(k)}$ and ends up with a new iterate $x^{(k+1)}$. In each *inner* iteration, within a given outer iteration, a convex subproblem is generated and solved. In this subproblem, the original objective and constraint functions are replaced by certain convex separable functions which approximate the original functions around $x^{(k)}$. The optimal solution of the subproblem is either accepted or rejected. If accepted, it becomes $x^{(k+1)}$ and the outer iteration is completed. If rejected, a new inner iteration is made, with a modified subproblem based on somewhat modified approximating functions. These inner iterations are repeated until the approximating objective and constraint functions become greater than or equal to the original functions at the optimal solution of the subproblem. When this happens, we say that the approximating functions are *conservative*. This does not imply that the feasible set of the subproblem is completely contained in the original feasible set, but it does imply that the optimal solution of the subproblem is a feasible solution of the original problem, with lower objective value than the previous iterate. Each new outer iteration requires function values and first order derivatives of the original objective and constraint functions, calculated at the current iterate $x^{(k)}$. Each new inner iteration requires function values, but no derivatives, calculated at the optimal solution of the most recent subproblem.

To use an approach based on solving a sequence of convex subproblems is not a new idea. It is used also in, e.g., *sequential quadratic programming* (SQP) where, at each iteration, a convex quadratic programming (QP) problem is solved and a

*Received by the editors October 20, 1999; accepted for publication (in revised form) December 5, 2000; published electronically January 4, 2002. This research was supported by the Swedish Research Council for the Engineering Sciences (TFR).
http://www.siam.org/journals/siopt/12-2/36282.html
†Optimization and Systems Theory, KTH, SE-10044 Stockholm, Sweden (krille@math.kth.se).

linesearch on a merit function is performed; see, e.g., [5] and [2]. However, the (linear) constraints in the QP subproblems do not in general force the iteration points to be feasible with respect to the original constraints, and thus they are not conservative in the above meaning. In contrast to SQP methods, CCSA methods introduce curvature both in the objective function and in the constraint functions of the subproblem. This curvature is updated in the inner iterations until the approximating functions become conservative, and then there is no need for any linesearch. Another class of methods which generate feasible iteration points is *interior point methods*, see, e.g., [1], [3], and [4]. But in these methods feasibility is typically preserved by adding a logarithmic barrier function to the objective function, and *not* by using conservative approximations of the constraint functions as in CCSA methods.

It should be emphasized that a major benefit of CCSA methods is that they can be successfully applied to problems with a very large number of variables, even if the Hessian matrices of the objective and constraint functions are dense. This property is to a large extent due to the usage of separable approximations.

One of the CCSA methods presented here, namely the *method of moving asymptotes* (MMA), has a background in the structural optimization field, where function and gradient evaluations are very time-consuming (involving huge finite element calculations), and where the users often consider it important that the generated iteration points are feasible. The original version of MMA, presented in [7], usually worked quite well in practice but was not globally convergent and sometimes failed on certain problems. A later version, presented in [8], was globally convergent but turned out to be too slow in practice. The version of MMA presented in this paper apparently outperforms both of these earlier versions, in theory as well as in practice. Moreover, MMA is now just one of several alternative methods within the concept of CCSA methods, which is introduced here and for which a convergent proof has not appeared before.

The paper is organized as follows. In section 2, a convenient formulation of inequality-constrained optimization problems is suggested and shown to have some important properties. In particular, the set of Karush–Kuhn–Tucker (KKT) points is nonempty. In sections 3 and 4, a general description of CCSA methods is given, and then some specific CCSA methods are described in sections 5 and 6. In section 7, it is proved that CCSA methods are globally convergent in the following sense: From any starting point, the sequence of generated iteration points converges towards the set of KKT points. In section 8, finally, numerical results on some large scale problems are presented.

2. Considered problem and some basic properties. Inequality-constrained nonlinear programming problems are often written in the following form, where $x = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$ is the vector of variables, $x_j^{\min}$ and $x_j^{\max}$ are given real numbers, and $f_0, f_1, \ldots, f_m$ are given, typically twice continuously differentiable, real-valued functions:

$$\begin{aligned} &\text{minimize} && f_0(x) && \\ &\text{subject to} && f_i(x) \le 0, && i = 1, \ldots, m, \\ &&& x_j^{\min} \le x_j \le x_j^{\max}, && j = 1, \ldots, n. \end{aligned} \tag{2.1}$$

In this paper, however, any problem of this type is transformed into a closely related problem of the following extended form where, in addition to the variables $x \in \mathbb{R}^n$,

there also appear "artificial" variables $y = (y_1, \ldots, y_m)^T \in \mathbb{R}^m$:

$$
\begin{aligned}
&\text{minimize} && f_0(x) + \sum_{i=1}^{m} c_i y_i && \\
&\text{subject to} && f_i(x) - y_i \le 0, && i = 1, \ldots, m, \\
& && x_j^{\min} \le x_j \le x_j^{\max}, && j = 1, \ldots, n, \\
& && y_i \ge 0, && i = 1, \ldots, m.
\end{aligned} \tag{2.2}
$$

If the constants c_i are chosen as very large numbers, then typically $\hat{y} = 0$ in any optimal solution $(\hat{x}, \hat{y})$ of (2.2), and then the corresponding $\hat{x}$ is an optimal solution of (2.1).

We prefer to work with (2.2) instead of (2.1) for several reasons. First, there always exist feasible solutions of (2.2), and also at least one optimal solution. Further, each optimal solution (local or global) of (2.2) always satisfies the KKT conditions. There is also a reason from a modelling point of view: In many applications, the user should be able to give a rough overestimate (possibly very large) of how much he would require in improved objective value in order to accept a unit increase of the right-hand side of a certain constraint in (2.1). Such an overestimate could then be used as the corresponding coefficient c_i in (2.2), and then problem (2.2) would be at least as relevant as problem (2.1).

As mentioned above, and as will be proved below, there always exists at least one KKT point of (2.2), i.e., a point which satisfies the KKT conditions of the problem. The following relations between KKT points of (2.1) and (2.2) can be readily seen by comparing the KKT conditions for the two problems. First, assume that $\hat{x}$ is a KKT point of (2.1) with Lagrange multipliers λ_i for the constraints $f_i(x) \le 0$, and assume that $c_i \ge \lambda_i$ for all i. Then $(x, y) = (\hat{x}, 0)$ is a KKT point of (2.2) with precisely these values λ_i on the Lagrange multipliers for the constraints $f_i(x) - y_i \le 0$. Next, assume that $(x, y) = (\hat{x}, 0)$ is a KKT point of (2.2) with Lagrange multipliers λ_i for the constraints $f_i(x) - y_i \le 0$ (which will of necessity satisfy $\lambda_i \le c_i$ for all i). Then $\hat{x}$ is a KKT point of (2.1) with precisely these values λ_i on the Lagrange multipliers for the constraints $f_i(x) \le 0$. If there happens to be no KKT point of (2.1), then there is no KKT point of (2.2) with $y = 0$, no matter how large the coefficients c_i are chosen. In this case, however, there is always at least one KKT point of (2.2) with $y \ne 0$.

For the remainder of this paper, we will in fact consider a further extended problem formulation, with one more "artificial" variable $z \in \mathbb{R}$. This formulation contains (2.2) as a special case, but also some other important problem classes such as least squares problems and minimax problems. The (small) price we have to pay for this generality is that the formulation of the problem may look a bit messy at first sight, namely as follows:

$$
\begin{aligned}
&\text{minimize} && f_0(x) + a_0 z + \sum_{i=1}^{m} (c_i y_i + \tfrac{1}{2} d_i y_i^2) && \\
&\text{subject to} && f_i(x) - a_i z - y_i \le 0, && i = 1, \ldots, m, \\
& && x_j^{\min} \le x_j \le x_j^{\max}, && j = 1, \ldots, n, \\
& && z \ge 0 \ \text{ and } \ y_i \ge 0, && i = 1, \ldots, m.
\end{aligned} \tag{2.3}
$$

Here, $f_0, f_1, \ldots, f_m$ are given, twice continuously differentiable, real-valued functions, while a_0, a_i, c_i, and d_i are given real numbers such that $a_0 > 0$, $a_i \ge 0$, $c_i \ge 0$, $d_i \ge 0$,

and $c_i + d_i > 0$ for $i = 1, \ldots, m$. Further, $a_i c_i > a_0$ for all i such that $a_i > 0$. Finally, $x_j^{\min}$ and $x_j^{\max}$ are given real numbers such that $x_j^{\min} < x_j^{\max}$ for $j = 1, \ldots, n$.

Problem (2.2) is obtained as a special case of (2.3) by letting $a_i = d_i = 0$ for $i = 1, \ldots, m$ and $a_0 = 1$, since then $z = 0$ in any optimal solution of (2.3).

As will be shown below, the considered problem (2.3) is equivalent to the following, typically nonsmooth, problem (2.4) in the variables $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$:

$$\begin{aligned} &\text{minimize} && f_0(x) + a_0 \max_{i \in \mathcal{A}_1} \left\{ \frac{f_i^+(x)}{a_i} \right\} + \sum_{i \in \mathcal{A}_0} \left(c_i f_i^+(x) + \tfrac{1}{2} d_i (f_i^+(x))^2 \right) \\ &\text{subject to} && x \in X, \end{aligned} \tag{2.4}$$

where we have used the notation

$$\begin{aligned} X &= \{x \in \mathbb{R}^n \mid x_j^{\min} \le x_j \le x_j^{\max},\ j = 1, \ldots, n\}, \\ \mathcal{A}_1 &= \{\, i \in \{1, \ldots, m\} \mid a_i > 0 \,\}, \\ \mathcal{A}_0 &= \{\, i \in \{1, \ldots, m\} \mid a_i = 0 \,\}, \text{ and} \\ f_i^+(x) &= \max\{0, f_i(x)\}. \end{aligned}$$

This formulation (2.4) will *not* be used for solving problem (2.3), but it shows that least squares problems, minimum 1-norm problems, and minimax problems are all special cases of problem (2.3). It is also used in the proof of Proposition 2.3 below.

PROPOSITION 2.1. *If $x \in X$ is held fixed in problem* (2.3)*, the corresponding optimal values of the variables y and z are unique. These unique optimal values are as follows: If $\mathcal{A}_1 = \emptyset$, then $z = 0$ and $y_i = f_i^+(x)$ for $i \in \{1, \ldots, m\}$. If $\mathcal{A}_1 \neq \emptyset$, then $z = \max_{i \in \mathcal{A}_1} \{\frac{f_i^+(x)}{a_i}\}$, $y_i = 0$ for $i \in \mathcal{A}_1$, and $y_i = f_i^+(x)$ for $i \in \mathcal{A}_0$.*

Proof. If $\mathcal{A}_1 = \emptyset$, the result follows from the assumptions that $a_0 > 0$ and $c_i + d_i > 0$ for all i. If $\mathcal{A}_1 \neq \emptyset$, one also has to use the assumptions that $a_i c_i > a_0$ for all $i \in \mathcal{A}_1$. ∎

This implies that the variables y_i and z can formally be eliminated from problem (2.3). The resulting problem is precisely (2.4). This gives our next proposition.

PROPOSITION 2.2. *The vector $(\hat{x}, \hat{y}, \hat{z})$ is a global optimal solution of problem* (2.3) *if and only if $\hat{x}$ is a global optimal solution of problem* (2.4) *while $\hat{y}$ and $\hat{z}$ are as in Proposition* 2.1*.*

Proof. The proof follows from Proposition 2.1. ∎

PROPOSITION 2.3. *There is at least one global optimal solution of problem* (2.3)*.*

Proof. In problem (2.4), the objective function is continuous on the compact set X. Thus, there is at least one global optimal solution of problem (2.4). But then Proposition 2.2 implies that there is at least one global optimal solution of problem (2.3). ∎

PROPOSITION 2.4. *If $(\hat{x}, \hat{y}, \hat{z})$ is an optimal solution, local or global, of problem* (2.3)*, then there are Lagrange multipliers which together with $(\hat{x}, \hat{y}, \hat{z})$ satisfy the KKT conditions.*

Proof. It is well known (see, e.g., section 9.4 in [6]) that if $\hat{x}$ is an optimal solution of a problem of the form

$$\begin{aligned} &\text{minimize} && h_0(x) \\ &\text{subject to} && h_i(x) \le 0, \quad i = 1, \ldots, m, \\ & && x \in \mathbb{R}^n, \end{aligned} \tag{2.5}$$

and if there is a vector Δx such that $\nabla h_i(\hat{x})\Delta x < 0$ for all $i > 0$ such that $h_i(\hat{x}) = 0$ (i.e., the inner product of Δx and the gradient vector of any active constraint is strictly negative), then there are Lagrange multipliers $\lambda_i,\ i = 1, \ldots, m$, which together with $\hat{x}$ satisfy the KKT conditions, which in this case are

$$\begin{aligned} \frac{\partial h_0}{\partial x_j}(\hat{x}) + \sum_{i=1}^{m} \lambda_i \frac{\partial h_i}{\partial x_j}(\hat{x}) &= 0, \quad j = 1, \ldots, n \quad (\partial L/\partial x_j = 0), \\ h_i(\hat{x}) &\le 0, \quad i = 1, \ldots, m \quad \text{(primal feasibility)}, \\ \lambda_i &\ge 0, \quad i = 1, \ldots, m \quad \text{(dual feasibility)}, \\ \lambda_i h_i(\hat{x}) &= 0, \quad i = 1, \ldots, m \quad \text{(compl slackness)}. \end{aligned}$$

This result shall now be applied to problem (2.3). Assume that $(\hat{x}, \hat{y}, \hat{z})$ is an optimal solution of problem (2.3) and construct a corresponding vector $(\Delta x, \Delta y, \Delta z)$ as follows. For $j = 1, \ldots, n$, let $\Delta x_j = 1$ if $\hat{x}_j = x_j^{\min}$, $\Delta x_j = -1$ if $\hat{x}_j = x_j^{\max}$, $\Delta x_j = 0$, otherwise. For $i = 1, \ldots, m$, let $\Delta y_i = 1 + \sum_{j=1}^{n} |\frac{\partial f_i}{\partial x_j}(\hat{x})|$. Finally, let $\Delta z = 1$.

Then it is easily checked that the inner product of $(\Delta x, \Delta y, \Delta z)$ and the gradient vector, calculated at $(\hat{x}, \hat{y}, \hat{z})$, of any active constraint in problem (2.3) is strictly negative. □

3. General description of a CCSA method. A CCSA method for solving problems of the form (2.3) consists of "outer" and "inner" iterations. The index k is used to denote the outer iteration number, while the index ℓ is used to denote the inner iteration number. Within each outer iteration, there may be zero, one, or several inner iterations. The double index (k, ℓ) is used to denote the ℓth inner iteration within the kth outer iteration.

The first iteration point $(x^{(1)}, y^{(1)}, z^{(1)})$ is obtained by first choosing an $x^{(1)} \in X$, and then calculating $y^{(1)}$ and $z^{(1)}$ in accordance with Proposition 2.1.

An outer iteration, going from the kth iteration point $(x^{(k)}, y^{(k)}, z^{(k)})$ to the $(k+1)$th iteration point $(x^{(k+1)}, y^{(k+1)}, z^{(k+1)})$, can be described as follows.

Given $(x^{(k)}, y^{(k)}, z^{(k)})$, an approximating subproblem is generated and solved. This subproblem is obtained from (2.3) by replacing X with a certain convex subset $X^{(k)}$ and by replacing the functions $f_i(x)$ with certain strictly convex separable functions $g_i^{(k,0)}(x)$ satisfying $g_i^{(k,0)}(x^{(k)}) = f_i(x^{(k)})$. The optimal solution of this subproblem is denoted $(\hat{x}^{(k,0)}, \hat{y}^{(k,0)}, \hat{z}^{(k,0)})$.

If $g_i^{(k,0)}(\hat{x}^{(k,0)}) \ge f_i(\hat{x}^{(k,0)})$ for all $i = 0, 1, \ldots, m$, the next iteration point becomes $(x^{(k+1)}, y^{(k+1)}, z^{(k+1)}) = (\hat{x}^{(k,0)}, \hat{y}^{(k,0)}, \hat{z}^{(k,0)})$, and the outer iteration is completed (without any inner iterations needed).

Otherwise, an inner iteration is made, which means that a new subproblem is generated and solved at $x^{(k)}$, with new approximating functions $g_i^{(k,1)}(x)$, still satisfying $g_i^{(k,1)}(x^{(k)}) = f_i(x^{(k)})$ but more conservative than $g_i^{(k,0)}(x)$ for those indices i for which the above inequality was violated. The optimal solution of this new subproblem is denoted $(\hat{x}^{(k,1)}, \hat{y}^{(k,1)}, \hat{z}^{(k,1)})$.

If $g_i^{(k,1)}(\hat{x}^{(k,1)}) \ge f_i(\hat{x}^{(k,1)})$ for all $i = 0, 1, \ldots, m$, the next iteration point becomes $(x^{(k+1)}, y^{(k+1)}, z^{(k+1)}) = (\hat{x}^{(k,1)}, \hat{y}^{(k,1)}, \hat{z}^{(k,1)})$, and the outer iteration is completed. Otherwise, another inner iteration is made, with new approximating functions $g_i^{(k,2)}(x)$, etc.

These inner iterations are repeated until $g_i^{(k,\ell)}(\hat{x}^{(k,\ell)}) \ge f_i(\hat{x}^{(k,\ell)})$ for all $i = 0, 1, \ldots, m$, which always happens after a finite number of inner iterations. Then the

next iteration point becomes $(x^{(k+1)}, y^{(k+1)}, z^{(k+1)}) = (\hat{x}^{(k,\ell)}, \hat{y}^{(k,\ell)}, \hat{z}^{(k,\ell)})$, and the outer iteration is completed (with ℓ inner iterations needed).

4. Requirements on the approximating functions. The CCSA subproblem looks as follows, for $k \in \{1, 2, 3, \ldots\}$ and $\ell \in \{0, 1, 2, \ldots\}$:

$$\begin{aligned} \text{minimize} \quad & g_0^{(k,\ell)}(x) + a_0 z + \sum_{i=1}^{m} (c_i y_i + \tfrac{1}{2} d_i y_i^2) \\ \text{subject to} \quad & g_i^{(k,\ell)}(x) - a_i z - y_i \le 0, \qquad\qquad i = 1, \ldots, m, \\ & x \in X^{(k)}, \quad y \ge 0, \quad z \ge 0, \end{aligned}$$

where the set $X^{(k)}$ and the approximating functions $g_i^{(k,\ell)}(x)$ will be specified below.

The set $X^{(k)}$ is chosen as $X^{(k)} = X(x^{(k)}, \sigma^{(k)})$, where $\sigma^{(k)} = (\sigma_1^{(k)}, \ldots, \sigma_n^{(k)})^T$ is a vector of strictly positive parameters, and $X(\xi, \sigma)$ is the subset of X defined by

$$X(\xi, \sigma) = \{x \in X \mid x_j \in [\xi_j - 0.9\sigma_j,\ \xi_j + 0.9\sigma_j],\ j = 1, \ldots, n\}.$$

Thus,

$$X^{(k)} = \{x \in X \mid x_j \in [x_j^{(k)} - 0.9\sigma_j^{(k)}, x_j^{(k)} + 0.9\sigma_j^{(k)}],\ j = 1, \ldots, n\}.$$

How to choose values on the parameters $\sigma_j^{(k)}$ will be discussed later. For the moment, it is sufficient to know that each vector $\sigma^{(k)}$ belongs to a given compact set S of the form

$$S = \{\sigma \in \mathbb{R}^n \mid \sigma_j^{\min} \le \sigma_j \le \sigma_j^{\max},\ j = 1, \ldots, n\}, \tag{4.1}$$

where $\sigma_j^{\min}$ and $\sigma_j^{\max}$ are given real numbers such that $0 < \sigma_j^{\min} < \sigma_j^{\max} < \infty$.

The approximating functions $g_i^{(k,\ell)}(x)$ in the CCSA subproblem are chosen as

$$g_i^{(k,\ell)}(x) = v_i(x, x^{(k)}, \sigma^{(k)}) + \rho_i^{(k,\ell)} w_i(x, x^{(k)}, \sigma^{(k)}), \quad i = 0, 1, \ldots, m, \tag{4.2}$$

where $v_i(x, \xi, \sigma)$ and $w_i(x, \xi, \sigma)$ are real-valued functions defined on the set D defined by

$$D = \{(x, \xi, \sigma) \mid \xi \in X,\ \sigma \in S,\ x \in X(\xi, \sigma)\}.$$

In order to ensure that the functions $g_i^{(k,\ell)}(x)$ in (4.2) have suitable properties, the following conditions (4.3a)–(4.3k) must be satisfied for $i = 0, 1, \ldots, m$:

(4.3a) v_i and w_i are continuous functions on the set D,

(4.3b) $\nabla_x v_i = \left(\dfrac{\partial v_i}{\partial x_1}, \ldots, \dfrac{\partial v_i}{\partial x_n}\right)$ exists and is continuous on D,

(4.3c) $\nabla_x w_i = \left(\dfrac{\partial w_i}{\partial x_1}, \ldots, \dfrac{\partial w_i}{\partial x_n}\right)$ exists and is continuous on D,

(4.3d) the $n \times n$ Hessian matrix $\nabla_{xx}^2 v_i$ exists and is continuous on D,

(4.3e) the $n \times n$ Hessian matrix $\nabla_{xx}^2 w_i$ exists and is continuous on D,

(4.3f) $v_i(x, \xi, \sigma) = f_i(x)$ if $x = \xi \in X$,

(4.3g) $w_i(x,\xi,\sigma) = 0$ if $x = \xi \in X$,

(4.3h) $\nabla_x v_i(x,\xi,\sigma) = \nabla f_i(x)$ if $x = \xi \in X$,

(4.3i) $\nabla_x w_i(x,\xi,\sigma) = (0,\dots,0)$ if $x = \xi \in X$,

(4.3j) $\nabla^2_{xx} v_i(x,\xi,\sigma)$ is positive semidefinite for all $(x,\xi,\sigma) \in D$,

(4.3k) $\nabla^2_{xx} w_i(x,\xi,\sigma)$ is positive definite for all $(x,\xi,\sigma) \in D$.

Some choices of appropriate function v_i and w_i will be suggested in section 5. The parameters $\rho_i^{(k,\ell)}$ are strictly positive. The larger the $\rho_i^{(k,\ell)}$, the more conservative the approximation will be. Within a given outer iteration k, the only differences between two inner iterations are the values of these $\rho_i^{(k,\ell)}$. How to choose values on these parameters will be described below.

It follows from the above conditions that the functions $g_i^{(k,\ell)}$ are first order approximations of the original functions f_i at the current iteration point, i.e.,

$$g_i^{(k,\ell)}(x^{(k)}) = f_i(x^{(k)}) \quad \text{and} \quad \nabla g_i^{(k,\ell)}(x^{(k)}) = \nabla f_i(x^{(k)})\,.$$

Further, the approximating functions $g_i^{(k,\ell)}$ are strictly convex since $\rho_i^{(k,\ell)} > 0$. In addition to the above conditions (4.3a)–(4.3k), the approximating functions should be separable, i.e., on the form

$$g_i^{(k,\ell)}(x) = g_{i0}^{(k,\ell)} + \sum_{j=1}^{n} g_{ij}^{(k,\ell)}(x_j).$$

This property is not used in the forthcoming theoretical analysis of global convergence, but it is essential in practice when attacking large scale problems.

5. Four examples of CCSA functions. In this section we give four different examples of CCSA functions v_i and w_i. In each of these four examples, and for each fixed vector $\lambda \geq 0 \in \mathbb{R}^m$, the Lagrange function $L(x,y,z,\lambda)$ corresponding to the CCSA subproblem can easily be minimized analytically with respect to $x \in X^{(k)}$, $y \geq 0$, and $z \geq 0$. If all $d_i > 0$ and a term εz^2 is added to the objective function, this analytical minimization gives a unique point $(\hat{x}(\lambda), \hat{y}(\lambda), \hat{z}(\lambda))$. The concave dual function $\varphi(\lambda) = L(\hat{x}(\lambda), \hat{y}(\lambda), \hat{z}(\lambda), \lambda)$ then becomes an explicit function, and the dual problem of maximizing $\varphi(\lambda)$ subject to the simple bounds $\lambda_i \geq 0,\ i = 1,\dots,m$, can be solved by, e.g., a conjugate gradient or a Newton-type method, combined with an active set strategy to take care of the nonnegativity constraints on the dual variables. If $\hat{\lambda}$ is an optimal solution of this dual problem, then $(x,y,z) = (\hat{x}(\hat{\lambda}), \hat{y}(\hat{\lambda}), \hat{z}(\hat{\lambda}))$ is the unique optimal solution of the CCSA subproblem.

Example 5.1. Linear and separable quadratic approximations:

$$v_i(x,\xi,\sigma) = f_i(\xi) + \nabla f_i(\xi)(x-\xi), \text{ and}$$
$$w_i(x,\xi,\sigma) = \frac{1}{2}\sum_{j=1}^{n}\left(\frac{x_j - \xi_j}{\sigma_j}\right)^2, \text{ so that}$$
$$g_i^{(k,\ell)}(x) = f_i(x^{(k)}) + \nabla f_i(x^{(k)})(x - x^{(k)}) + \frac{\rho_i^{(k,\ell)}}{2}\sum_{j=1}^{n}\left(\frac{x_j - x_j^{(k)}}{\sigma_j^{(k)}}\right)^2.$$

Example 5.2. Linear and separable logarithm approximations:

$$v_i(x,\xi,\sigma) = f_i(\xi) + \nabla f_i(\xi)(x-\xi), \text{ and}$$

$$
\begin{aligned}
w_i(x,\xi,\sigma) &= -\tfrac{1}{2}\sum_{j=1}^{n} \ln\left(1-(x_j-\xi_j)^2/\sigma_j^2\right), \text{ so that}\\
g_i^{(k,\ell)}(x) &= f_i(x^{(k)}) + \nabla f_i(x^{(k)})(x-x^{(k)})\\
&\quad - \frac{\rho_i^{(k,\ell)}}{2}\sum_{j=1}^{n} \ln\left(1-(x_j-x_j^{(k)})^2/(\sigma_j^{(k)})^2\right).
\end{aligned}
$$

Example 5.3. Linear and separable square root approximations:

$$
\begin{aligned}
v_i(x,\xi,\sigma) &= f_i(\xi) + \nabla f_i(\xi)(x-\xi), \text{ and}\\
w_i(x,\xi,\sigma) &= \sum_{j=1}^{n}\left(1-\sqrt{1-(x_j-\xi_j)^2/\sigma_j^2}\right), \text{ so that}\\
g_i^{(k,\ell)}(x) &= f_i(x^{(k)}) + \nabla f_i(x^{(k)})(x-x^{(k)})\\
&\quad + \rho_i^{(k,\ell)}\sum_{j=1}^{n}\left(1-\sqrt{1-(x_j-x_j^{(k)})^2/(\sigma_j^{(k)})^2}\right).
\end{aligned}
$$

Example 5.4. MMA approximations: Here, the approximating functions are chosen as

$$
g_i^{(k,\ell)}(x) = \sum_{j=1}^{n}\left(\frac{p_{ij}^{(k,\ell)}}{u_j^{(k)}-x_j} + \frac{q_{ij}^{(k,\ell)}}{x_j-l_j^{(k)}}\right) + r_i^{(k,\ell)},
$$

where the "moving asymptotes" $l_j^{(k)}$ and $u_j^{(k)}$ are given by

$$
l_j^{(k)} = x_j^{(k)} - \sigma_j^{(k)} \quad \text{and} \quad u_j^{(k)} = x_j^{(k)} + \sigma_j^{(k)},
$$

while the coefficients $p_{ij}^{(k,\ell)}$, $q_{ij}^{(k,\ell)}$, and $r_i^{(k,\ell)}$ are given by

$$
\begin{aligned}
p_{ij}^{(k,\ell)} &= (\sigma_j^{(k)})^2 \max\left\{0\,,\,\frac{\partial f_i}{\partial x_j}(x^{(k)})\right\} + \frac{\rho_i^{(k,\ell)}\sigma_j^{(k)}}{4},\\
q_{ij}^{(k,\ell)} &= (\sigma_j^{(k)})^2 \max\left\{0\,,\,-\frac{\partial f_i}{\partial x_j}(x^{(k)})\right\} + \frac{\rho_i^{(k,\ell)}\sigma_j^{(k)}}{4},\\
r_i^{(k,\ell)} &= f_i(x^{(k)}) - \sum_{j=1}^{n}\frac{p_{ij}^{(k,\ell)}+q_{ij}^{(k,\ell)}}{\sigma_j^{(k)}}.
\end{aligned}
$$

This means that

$$
g_i^{(k,\ell)}(x) = v_i(x,x^{(k)},\sigma^{(k)}) + \rho_i^{(k,\ell)} w_i(x,x^{(k)},\sigma^{(k)}),
$$

where, after some manipulations,

$$
\begin{aligned}
v_i(x,\xi,\sigma) &= f_i(\xi) + \sum_{j=1}^{n}\frac{\sigma_j^2\frac{\partial f_i}{\partial x_j}(\xi)(x_j-\xi_j) + \sigma_j\left|\frac{\partial f_i}{\partial x_j}(\xi)\right|(x_j-\xi_j)^2}{\sigma_j^2-(x_j-\xi_j)^2}, \text{ and}\\
w_i(x,\xi,\sigma) &= \tfrac{1}{2}\sum_{j=1}^{n}\frac{(x_j-\xi_j)^2}{\sigma_j^2-(x_j-\xi_j)^2}.
\end{aligned}
$$

Example 5.4 defines the new globally convergent version of MMA, which is a further development of [8]. The original MMA, [7], can be considered as a special case of the above by letting all $\rho_i^{(k,\ell)} = 0$. Consequently, no inner iterations were performed in the original MMA, and global convergence could not be proved.

6. Rules for updating the parameters $\rho_i^{(k,\ell)}$ and $\sigma_j^{(k)}$. We begin with the parameters $\rho_i^{(k,\ell)}$. For $\ell = 0$, the following values are used, where $\rho_i^{\min}$ is a fixed, strictly positive "small" number, e.g., 10^{-5}:

$$\rho_i^{(1,0)} = 1, \tag{6.1a}$$

$$\rho_i^{(k+1,0)} = \max\{0.1\rho_i^{(k,\hat{\ell}(k))},\ \rho_i^{\min}\}, \tag{6.1b}$$

where $\hat{\ell}(k)$ is the number of inner iterations needed within the kth outer iteration, so that $\rho_i^{(k,\hat{\ell}(k))}$ is the latest value of $\rho_i^{(k,\ell)}$.

In each inner iteration, the updating of $\rho_i^{(k,\ell)}$ is based on the solution of the most recent subproblem. If $g_i^{(k,\ell)}(\hat{x}^{(k,\ell)}) < f_i(\hat{x}^{(k,\ell)})$, it is natural to choose $\rho_i^{(k,\ell+1)}$ so that

$$g_i^{(k,\ell+1)}(\hat{x}^{(k,\ell)}) = f_i(\hat{x}^{(k,\ell)}),$$

which in view of (4.2) gives that $\rho_i^{(k,\ell+1)} = \rho_i^{(k,\ell)} + \delta_i^{(k,\ell)}$, where

$$\delta_i^{(k,\ell)} = \frac{f_i(\hat{x}^{(k,\ell)}) - g_i^{(k,\ell)}(\hat{x}^{(k,\ell)})}{w_i(\hat{x}^{(k,\ell)}, x^{(k)}, \sigma^{(k)})}\ .$$

In order to get a globally convergent method, this natural value is modified as follows:

$$\begin{aligned} \rho_i^{(k,\ell+1)} &= \min\{10\rho_i^{(k,\ell)},\ 1.1(\rho_i^{(k,\ell)} + \delta_i^{(k,\ell)})\} && \text{if } \delta_i^{(k,\ell)} > 0, \\ \rho_i^{(k,\ell+1)} &= \rho_i^{(k,\ell)} && \text{if } \delta_i^{(k,\ell)} \le 0. \end{aligned} \tag{6.2}$$

This means that in the beginning of each new inner iteration, the parameters ρ_i are increased or unaltered but never decreased. Therefore, it is important that they can be decreased again in the beginning of each new outer iteration, as they are in (6.1b), since otherwise the method could be too conservative.

Now to the values of the parameters $\sigma_j^{(k)}$. Updating rules for these parameters depend on the specific functions v_i and w_i. In each of the four examples in the previous section, the $n \times n$ Hessian matrix $\nabla^2_{xx} w_i(x, \xi, \sigma)$ is diagonal with $\frac{\partial^2 w_i}{\partial x_j^2}(x, \xi, \sigma) \ge \frac{1}{\sigma_j^2}$ for all j and every $(x, \xi, \sigma) \in D$, with equality if $x_j = \xi_j$. The curvature of the function w_i in the "x_j-direction" thus increases with decreasing values of σ_j. This makes the following heuristic rule for updating these parameters reasonable. If a certain variable x_j is oscillating, it should be stabilized by a decreased value of the corresponding σ_j. If the variable x_j is monotonically increasing, or monotonically decreasing, it should be released by an increased value of the corresponding σ_j. One possible way of implementing this rule is as follows.

In the first two outer iterations, when $k = 1$ and $k = 2$,

$$\sigma_j^{(k)} = 0.5(x_j^{\max} - x_j^{\min}),$$

while in later outer iterations, when $k \ge 3$,

$$\sigma_j^{(k)} = \gamma_j^{(k)} \sigma_j^{(k-1)},$$

where

$$\gamma_j^{(k)} = \begin{cases} 0.7 & \text{if} \quad (x_j^{(k)} - x_j^{(k-1)})(x_j^{(k-1)} - x_j^{(k-2)}) < 0, \\ 1.2 & \text{if} \quad (x_j^{(k)} - x_j^{(k-1)})(x_j^{(k-1)} - x_j^{(k-2)}) > 0, \\ 1 & \text{if} \quad (x_j^{(k)} - x_j^{(k-1)})(x_j^{(k-1)} - x_j^{(k-2)}) = 0, \end{cases}$$

provided that this leads to values that satisfy

$$0.01(x_j^{\max} - x_j^{\min}) \le \sigma_j^{(k)} \le 10(x_j^{\max} - x_j^{\min}).$$

If any of these bounds is violated, the corresponding $\sigma_j^{(k)}$ is set to the violated bound. Thus, $\sigma_j^{\min} = 0.01(x_j^{\max} - x_j^{\min})$ and $\sigma_j^{\max} = 10(x_j^{\max} - x_j^{\min})$ in the set S defined in (4.1) above.

7. Theoretical analysis of global convergence. A given point $(x, y, z) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}$ is a KKT point of the problem (2.3) if and only if there are Lagrange multipliers which together with (x, y, z) satisfy the KKT conditions of the problem.

Let Ω be the set of KKT points of the original problem (2.3). Ω is nonempty by Propositions 2.3 and 2.4. Then let $\|\Omega - (x^{(k)}, y^{(k)}, z^{(k)})\|$ denote the Euclidean distance from the point $(x^{(k)}, y^{(k)}, z^{(k)})$ to the set Ω, i.e.,

$$\|\Omega - (x^{(k)}, y^{(k)}, z^{(k)})\| \; = \inf_{(x,y,z)\in\Omega} \{ \, \|(x, y, z) - (x^{(k)}, y^{(k)}, z^{(k)})\| \, \}.$$

THEOREM 7.1. *If any of the CCSA methods described above is applied to a problem of the form* (2.3)*, then* $\|\Omega - (x^{(k)}, y^{(k)}, z^{(k)})\| \to 0$ *as* $k \to \infty$.

Before the proof of this main theorem, some preparations are needed.

LEMMA 7.2. *In each outer iteration k, only a finite number ℓ of inner iterations are needed until $g_i^{(k,\ell)}(\hat{x}^{(k,\ell)}) \ge f_i(\hat{x}^{(k,\ell)})$ for all i.*

Proof. A sufficient condition for the inequality $g_i^{(k,\ell)}(\hat{x}^{(k,\ell)}) \ge f_i(\hat{x}^{(k,\ell)})$ to hold is that $\rho_i^{(k,\ell)}\tau_i \ge \kappa_i$, where

$$\kappa_i = \max_{x,h}\{h^T\nabla^2 f_i(x)h \mid x \in X, \; h \in \mathbb{R}^n, \; h^T h = 1\}, \text{ and}$$
$$\tau_i = \min_{x,\xi,\sigma,h}\{h^T\nabla^2_{xx} w_i(x, \xi, \sigma)h \mid (x, \xi, \sigma) \in D, \; h \in \mathbb{R}^n, \; h^T h = 1\}.$$

The number κ_i is finite since the Hessian matrix $\nabla^2 f_i(x)$ is continuous on X. The number τ_i is finite and strictly positive since the Hessian matrix $\nabla^2_{xx} w_i(x, \xi, \sigma)$ is positive definite and continuous in all its arguments. But each time that $g_i^{(k,\ell)}(\hat{x}^{(k,\ell)}) < f_i(\hat{x}^{(k,\ell)})$, the corresponding $\rho_i^{(k,\ell)}$ is increased by at least a factor 1.1; see (6.2). This can be done only a finite number of times, for each i, before $\rho_i^{(k,\ell)}\tau_i \ge \kappa_i$ is satisfied. (Note that, for a fixed k, $\rho_i^{(k,\ell)}$ is nondecreasing in ℓ.) □

As a consequence of this lemma, only outer iterations need to be considered in the analysis of global convergence. Therefore, the following shorter notations will be used:

$\hat{\ell}(k) =$ the number of inner iterations needed within the kth outer iteration,
$\rho_i^{(k)} = \rho_i^{(k,\hat{\ell}(k))}$, and $g_i^{(k)}(x) = g_i^{(k,\hat{\ell}(k))}(x)$.

This means that the subproblem used at the kth (outer) iteration to calculate the next iteration point is the following:

$$\begin{aligned} &\text{minimize} && g_0^{(k)}(x) + a_0 z + \sum_{i=1}^{m} (c_i y_i + \tfrac{1}{2} d_i y_i^2) \\ &\text{subject to} && g_i^{(k)}(x) - a_i z - y_i \le 0, \qquad i = 1, \ldots, m, \\ & && x \in X^{(k)}, \quad y \ge 0, \quad z \ge 0. \end{aligned} \tag{7.1}$$

The optimal solution of (7.1) is the new iteration point $(x^{(k+1)}, y^{(k+1)}, z^{(k+1)})$. Note that $g_i^{(k)}(x^{(k)}) = f_i(x^{(k)})$ and $g_i^{(k)}(x^{(k+1)}) \ge f_i(x^{(k+1)})$ for all $i = 0, 1, \ldots, m$.

LEMMA 7.3. *For each $i = 0, 1, \ldots, m$, there is a finite number $\rho_i^{\max}$ such that $\rho_i^{(k)} \le \rho_i^{\max}$ for all outer iterations k.*

Proof. From the updating rules (6.1b) and (6.2) and the proof of Lemma 7.2, it follows that $\rho_i^{(k)} \le 10(1 + \kappa_i/\tau_i)$ will always hold. □

Let the set Q be defined by

$$Q = \{\rho \in \mathbb{R}^{m+1} \mid \rho_i^{\min} \le \rho_i \le \rho_i^{\max},\ i = 0, 1, \ldots, m\}.$$

Let the functions F_i be defined, for $x \in X$, $y \in \mathbb{R}^m$, and $z \in \mathbb{R}$, by

$$\begin{aligned} F_0(x, y, z) &= f_0(x) + a_0 z + \sum_{i=1}^{m} (c_i y_i + \tfrac{1}{2} d_i y_i^2), \\ F_i(x, y, z) &= f_i(x) - a_i z - y_i, \quad i = 1, \ldots, m. \end{aligned}$$

Then the original problem (2.3) can be written

$$\begin{aligned} &\text{minimize} && F_0(x, y, z) \\ &\text{subject to} && F_i(x, y, z) \le 0, \qquad i = 1, \ldots, m, \\ & && x \in X, \quad y \ge 0, \quad z \ge 0. \end{aligned} \tag{7.2}$$

Let the functions G_i be defined, for $(x, \xi, \sigma) \in D$, $\rho \in Q$, $y \in \mathbb{R}^m$, and $z \in \mathbb{R}$, by

$$\begin{aligned} G_0(x, y, z, \xi, \sigma, \rho) &= v_0(x, \xi, \sigma) + \rho_0 w_0(x, \xi, \sigma) + a_0 z + \sum_{i=1}^{m} (c_i y_i + \tfrac{1}{2} d_i y_i^2), \\ G_i(x, y, z, \xi, \sigma, \rho) &= v_i(x, \xi, \sigma) + \rho_i w_i(x, \xi, \sigma) - a_i z - y_i, \quad i = 1, \ldots, m. \end{aligned}$$

Note that each function G_i is continuous on the set on which it is defined.

Let the problem PSUB(ξ, σ, ρ) be defined, for given $(\xi, \sigma, \rho) \in X \times S \times Q$, as the following problem in the variables (x, y, z):

$$\begin{aligned} &\text{minimize} && G_0(x, y, z, \xi, \sigma, \rho) \\ &\text{subject to} && G_i(x, y, z, \xi, \sigma, \rho) \le 0, \qquad i = 1, \ldots, m, \\ & && x \in X(\xi, \sigma),\ y \ge 0,\ z \ge 0. \end{aligned} \tag{7.3}$$

Then the CCSA subproblem (7.1) is equivalent to the problem PSUB$(x^{(k)}, \sigma^{(k)}, \rho^{(k)})$, i.e., the problem (7.3) with $\xi = x^{(k)}$, $\sigma = \sigma^{(k)}$, and $\rho = \rho^{(k)}$.

LEMMA 7.4. *For each given $\xi \in X$, $\sigma \in S$, and $\rho \in Q$, there is a unique optimal solution of PSUB(ξ, σ, ρ). This solution is also the only KKT point of PSUB(ξ, σ, ρ).*

Proof. The existence of an optimal solution follows by arguments similar to those in the proof of Proposition 2.3. The uniqueness follows from the fact that the problem obtained by eliminating y and z is strictly convex in x. Finally, PSUB(ξ,σ,ρ) is a convex problem for which the Slater's constraint qualifications are fulfilled. Therefore, the KKT conditions are both necessary and sufficient conditions for a global optimum. □

Thus, $(x^{(k+1)},y^{(k+1)},z^{(k+1)})$ is the only KKT point of PSUB$(x^{(k)},\sigma^{(k)},\rho^{(k)})$.

LEMMA 7.5. *For each given $\sigma\in S$ and $\rho\in Q$ the following holds: A given point $(\hat{x},\hat{y},\hat{z})$ is a KKT point of the original problem* (2.3) *if and only if $(\hat{x},\hat{y},\hat{z})$ is a KKT point of the subproblem PSUB$(\hat{x},\sigma,\rho)$.*

Proof. For a given $\hat{x}\in X$, let $B(\hat{x},\varepsilon)=\{x\in\mathbb{R}^n\ ;\ \|x-\hat{x}\|<\varepsilon\}$, and note that there is an $\varepsilon>0$ such that $X\cap B(\hat{x},\varepsilon)=X(\hat{x},\sigma)\cap B(\hat{x},\varepsilon)$. This implies that $(\hat{x},\hat{y},\hat{z})$ is the optimal solution of (the strictly convex problem) PSUB$(\hat{x},\sigma,\rho)$ if and only if $(\hat{x},\hat{y},\hat{z})$ is the optimal solution of PSUB$(\hat{x},\sigma,\rho)$ with the simple bound constraints $x\in X(\hat{x},\sigma)$ replaced by the (looser) simple bound constraints $x\in X$. Further, the following holds for $i=0,1,\ldots,m$:

$$\begin{aligned}
G_i(\hat{x},\hat{y},\hat{z},\hat{x},\sigma,\rho) &= F_i(\hat{x},\hat{y},\hat{z}),\\
\frac{\partial G_i}{\partial x_j}(\hat{x},\hat{y},\hat{z},\hat{x},\sigma,\rho) &= \frac{\partial F_i}{\partial x_j}(\hat{x},\hat{y},\hat{z}),\\
\frac{\partial G_i}{\partial y_j}(\hat{x},\hat{y},\hat{z},\hat{x},\sigma,\rho) &= \frac{\partial F_i}{\partial y_j}(\hat{x},\hat{y},\hat{z}),\\
\frac{\partial G_i}{\partial z}(\hat{x},\hat{y},\hat{z},\hat{x},\sigma,\rho) &= \frac{\partial F_i}{\partial z}(\hat{x},\hat{y},\hat{z}).
\end{aligned}$$

These observations imply that $(\hat{x},\hat{y},\hat{z})$ is a KKT point of the subproblem PSUB$(\hat{x},\sigma,\rho)$ if and only if $(\hat{x},\hat{y},\hat{z})$ is a KKT point of the problem (7.2). □

In particular, if $(x^{(k+1)},y^{(k+1)},z^{(k+1)})=(x^{(k)},y^{(k)},z^{(k)})$, then $(x^{(k)},y^{(k)},z^{(k)})$ is a KKT point of the original problem (2.3), and then the algorithm should be stopped. From now on, it is therefore assumed that $(x^{(k+1)},y^{(k+1)},z^{(k+1)})\neq(x^{(k)},y^{(k)},z^{(k)})$ for all k.

LEMMA 7.6. *Each generated iteration point is a feasible solution of the problem* (7.2)*, i.e., $F_i(x^{(k)},y^{(k)},z^{(k)})\le 0$ for $i\ge 1$ and $k\ge 1$. Further, each generated iteration point has a strictly lower objective value than the previous one, i.e., $F_0(x^{(k+1)},y^{(k+1)},z^{(k+1)})<F_0(x^{(k)},y^{(k)},z^{(k)})$ for $k\ge 1$.*

Proof. The starting point $(x^{(1)},y^{(1)},z^{(1)})$ is feasible by construction. After that, $F_i(x^{(k+1)},y^{(k+1)},z^{(k+1)})\le G_i(x^{(k+1)},y^{(k+1)},z^{(k+1)},x^{(k)},\sigma^{(k)},\rho^{(k)})\le 0$ for $i\ge 1$. Further, $F_0(x^{(k+1)},y^{(k+1)},z^{(k+1)})\le G_0(x^{(k+1)},y^{(k+1)},z^{(k+1)},x^{(k)},\sigma^{(k)},\rho^{(k)})<G_0(x^{(k)},y^{(k)},z^{(k)},x^{(k)},\sigma^{(k)},\rho^{(k)})=F_0(x^{(k)},y^{(k)},z^{(k)})$. □

LEMMA 7.7. *All the iteration points $(x^{(k)},y^{(k)},z^{(k)})$ remain in a compact set.*

Proof. First, $x^{(k)}\in X$, which is a compact set. Next, let the functions $\tilde{g}_i$ be defined, for $(x,\xi,\sigma)\in D$ and $\rho\in Q$, by

$$\tilde{g}_i(x,\xi,\sigma,\rho)=v_i(x,\xi,\sigma)+\rho_i w_i(x,\xi,\sigma).$$

Each function $\tilde{g}_i$ is continuous on the compact set on which it is defined.

By the same arguments as in Proposition 2.1, it follows that $y_i^{(k+1)}\le g_i^{(k)}(x^{(k+1)})$. But since $g_i^{(k)}(x^{(k+1)})=\tilde{g}_i(x^{(k+1)},x^{(k)},\sigma^{(k)},\rho^{(k)})$, it then follows that $y_i^{(k+1)}\le\tilde{g}_i(x^{(k+1)},x^{(k)},\sigma^{(k)},\rho^{(k)})\le\max\{\tilde{g}_i(x,\xi,\sigma,\rho)\mid(x,\xi,\sigma)\in D,\ \rho\in Q\}$.

The existence of an upper bound on $z^{(k)}$ is proved in a similar way. □

As a consequence of Lemma 7.7, the sequence $\{(x^{(k)}, y^{(k)}, z^{(k)})\}_{k=1}^{\infty}$ has at least one convergent subsequence. Thus, there is a point (x^*, y^*, z^*) and an infinite subset $\mathcal{K}$ of the positive integers such that $(x^{(k)}, y^{(k)}, z^{(k)}) \to (x^*, y^*, z^*)$ as $k \in \mathcal{K}$ and $k \to \infty$.

Further, since the sequence $\{(\sigma^{(k)}, \rho^{(k)})\}_{k\in\mathcal{K}}$ (with $\mathcal{K}$ from above) stays in the compact set $S\times Q$, there is a point $(\sigma^*, \rho^*) \in S\times Q$ and an infinite subset $\widetilde{\mathcal{K}} \subseteq \mathcal{K}$ such that $(\sigma^{(k)}, \rho^{(k)}) \to (\sigma^*, \rho^*)$ as $k \in \widetilde{\mathcal{K}}$ and $k \to \infty$.

Next, the sequence $\{(x^{(k+1)}, y^{(k+1)}, z^{(k+1)})\}_{k\in\widetilde{\mathcal{K}}}$ (with $\widetilde{\mathcal{K}}$ from above) also has at least one convergent subsequence. Thus, there is a point $(\bar{x}, \bar{y}, \bar{z})$ and an infinite subset $\overline{\mathcal{K}} \subseteq \widetilde{\mathcal{K}} \subseteq \mathcal{K}$ such that $(x^{(k+1)}, y^{(k+1)}, z^{(k+1)}) \to (\bar{x}, \bar{y}, \bar{z})$ as $k \in \overline{\mathcal{K}}$ and $k \to \infty$.

In the following, (x^*, y^*, z^*), (σ^*, ρ^*), and $(\bar{x}, \bar{y}, \bar{z})$ are these just-described limit points.

LEMMA 7.8. *$F_0(x^{(k)}, y^{(k)}, z^{(k)}) \to F_0(x^*, y^*, z^*)$ as $k \to \infty$ (not only for $k \in \mathcal{K}$).*

Proof. The sequence $\{F_0(x^{(k)}, y^{(k)}, z^{(k)})\}_{k=1}^{\infty}$ is monotonically decreasing and bounded below by the global optimal value of the problem (2.3) (which exists and is finite according to Proposition 2.3). Thus, $F_0(x^{(k)}, y^{(k)}, z^{(k)}) \to F_0^*$ as $k \to \infty$ for some real number F_0^*. But since $F_0(x^{(k)}, y^{(k)}, z^{(k)}) \to F_0(x^*, y^*, z^*)$ as $k \in \mathcal{K}$ and $k \to \infty$, it follows that $F_0^* = F_0(x^*, y^*, z^*)$. □

LEMMA 7.9. *$F_0(\bar{x}, \bar{y}, \bar{z}) = F_0(x^*, y^*, z^*)$.*

Proof. From Lemma 7.8, it follows that $F_0(x^{(k+1)}, y^{(k+1)}, z^{(k+1)}) \to F_0(x^*, y^*, z^*)$ as $k \in \overline{\mathcal{K}}$ and $k \to \infty$. But since $(x^{(k+1)}, y^{(k+1)}, z^{(k+1)}) \to (\bar{x}, \bar{y}, \bar{z})$ as $k \in \overline{\mathcal{K}}$ and $k \to \infty$, it also holds that $F_0(x^{(k+1)}, y^{(k+1)}, z^{(k+1)}) \to F_0(\bar{x}, \bar{y}, \bar{z})$ as $k \in \overline{\mathcal{K}}$ and $k \to \infty$. □

LEMMA 7.10. *$(\bar{x}, \bar{y}, \bar{z})$ is the unique optimal solution of PSUB(x^*, σ^*, ρ^*).*

Proof. Since $x^{(k+1)} \in X(x^{(k)}, \sigma^{(k)})$ and $G_i(x^{(k+1)}, y^{(k+1)}, z^{(k+1)}, x^{(k)}, \sigma^{(k)}, \rho^{(k)}) \le 0$, it follows, by letting $k \in \overline{\mathcal{K}}$ and $k \to \infty$, that $\bar{x} \in X(x^*, \sigma^*)$ and $G_i(\bar{x}, \bar{y}, \bar{z}, x^*, \sigma^*, \rho^*) \le 0$ for $i \ge 1$. Thus, $(\bar{x}, \bar{y}, \bar{z})$ is a feasible solution of PSUB(x^*, σ^*, ρ^*). Let $(\tilde{x}, \tilde{y}, \tilde{z})$ be an arbitrary feasible solution of PSUB(x^*, σ^*, ρ^*), so that $\tilde{x} \in X(x^*, \sigma^*)$ and $G_i(\tilde{x}, \tilde{y}, \tilde{z}, x^*, \sigma^*, \rho^*) \le 0$ for $i \ge 1$. We must show that $G_0(\bar{x}, \bar{y}, \bar{z}, x^*, \sigma^*, \rho^*) \le G_0(\tilde{x}, \tilde{y}, \tilde{z}, x^*, \sigma^*, \rho^*)$.

For $\nu = 1, 2, 3, \ldots$, let $\tilde{x}^{(\nu)} = \tilde{x} + \alpha^{(\nu)}(x^* - \tilde{x})$, $\tilde{y}^{(\nu)} = \tilde{y} + \frac{1}{\nu}(1, \ldots, 1)^T$, and $\tilde{z}^{(\nu)} = \tilde{z} + \frac{1}{\nu}$. If $\alpha^{(\nu)} = 0$, then $G_i(\tilde{x}^{(\nu)}, \tilde{y}^{(\nu)}, \tilde{z}^{(\nu)}, x^*, \sigma^*, \rho^*) \le -\frac{1}{\nu}$ for $i \ge 1$. It is therefore possible to choose the scalar $\alpha^{(\nu)}$ such that $0 < \alpha^{(\nu)} < 1/\nu$ and $G_i(\tilde{x}^{(\nu)}, \tilde{y}^{(\nu)}, \tilde{z}^{(\nu)}, x^*, \sigma^*, \rho^*) \le -\frac{1}{2\nu}$ for $i \ge 1$. Then $(\tilde{x}^{(\nu)}, \tilde{y}^{(\nu)}, \tilde{z}^{(\nu)})$ is in the interior of the feasible set of PSUB(x^*, σ^*, ρ^*). In particular, $\tilde{x}^{(\nu)}$ is in the interior of $X(x^*, \sigma^*)$. This implies that for each ν, there is an integer $K(\nu)$ such that, for all $k \in \overline{\mathcal{K}}$ with $k > K(\nu)$, $\tilde{x}^{(\nu)} \in X(x^{(k)}, \sigma^{(k)})$ and $G_i(\tilde{x}^{(\nu)}, \tilde{y}^{(\nu)}, \tilde{z}^{(\nu)}, x^{(k)}, \sigma^{(k)}, \rho^{(k)}) \le 0$ for $i \ge 1$. For all these $k \in \overline{\mathcal{K}}$ with $k > K(\nu)$ it then holds that $G_0(\tilde{x}^{(\nu)}, \tilde{y}^{(\nu)}, \tilde{z}^{(\nu)}, x^{(k)}, \sigma^{(k)}, \rho^{(k)}) \ge G_0(x^{(k+1)}, y^{(k+1)}, z^{(k+1)}, x^{(k)}, \sigma^{(k)}, \rho^{(k)})$ (because $(x^{(k+1)}, y^{(k+1)}, z^{(k+1)})$ is the optimal solution of PSUB$(x^{(k)}, \sigma^{(k)}, \rho^{(k)})$).

Now, for each ν, let the integer $k(\nu) \in \overline{\mathcal{K}}$ satisfy $k(\nu) > \max\{\nu, K(\nu)\}$, and let $\nu \longrightarrow \infty$. Then $(\tilde{x}^{(\nu)}, \tilde{y}^{(\nu)}, \tilde{z}^{(\nu)}) \longrightarrow (\tilde{x}, \tilde{y}, \tilde{z})$, $(x^{(k(\nu)+1)}, y^{(k(\nu)+1)}, z^{(k(\nu)+1)}) \longrightarrow (\bar{x}, \bar{y}, \bar{z})$, and $(x^{(k(\nu))}, \sigma^{(k(\nu))}, \rho^{(k(\nu))}) \longrightarrow (x^*, \sigma^*, \rho^*)$. Thus, $G_0(\tilde{x}, \tilde{y}, \tilde{z}, x^*, \sigma^*, \rho^*) \ge G_0(\bar{x}, \bar{y}, \bar{z}, x^*, \sigma^*, \rho^*)$. □

LEMMA 7.11. *$(\bar{x}, \bar{y}, \bar{z}) = (x^*, y^*, z^*)$.*

Proof. From $G_i(x^{(k)}, y^{(k)}, z^{(k)}, x^{(k)}, \sigma^{(k)}, \rho^{(k)}) = F_i(x^{(k)}, y^{(k)}, z^{(k)}) \le 0$ for $i \ge 1$, it follows, by letting $k \in \overline{\mathcal{K}}$ and $k \to \infty$, that $G_i(x^*, y^*, z^*, x^*, \sigma^*, \rho^*) \le 0$ for $i \ge 1$. Further, by definition, $x^* \in X(x^*, \sigma^*)$. Thus, (x^*, y^*, z^*) is a feasible solution of PSUB(x^*, σ^*, ρ^*).

From $F_0(x^{(k+1)}, y^{(k+1)}, z^{(k+1)}) \leq G_0(x^{(k+1)}, y^{(k+1)}, z^{(k+1)}, x^{(k)}, \sigma^{(k)}, \rho^{(k)})$, it follows, again by letting $k \in \overline{\mathcal{K}}$ and $k \to \infty$, that $F_0(\bar{x}, \bar{y}, \bar{z}) \leq G_0(\bar{x}, \bar{y}, \bar{z}, x^*, \sigma^*, \rho^*)$.

By definition, $F_0(x^*, y^*, z^*) = G_0(x^*, y^*, z^*, x^*, \sigma^*, \rho^*)$. From Lemma 7.9 it then follows that $G_0(x^*, y^*, z^*, x^*, \sigma^*, \rho^*) \leq G_0(\bar{x}, \bar{y}, \bar{z}, x^*, \sigma^*, \rho^*)$. But since $(\bar{x}, \bar{y}, \bar{z})$ is the unique global optimal solution of PSUB(x^*, σ^*, ρ^*), it then follows that $(x^*, y^*, z^*) = (\bar{x}, \bar{y}, \bar{z})$. □

LEMMA 7.12. *(x^*, y^*, z^*) is a KKT point of the problem* (7.2).

Proof. Follows from Lemmas 7.4, 7.5, 7.10, and 7.11. □

Proof of Theorem 7.1. Assume that the statement in Theorem 7.1 is false. Then there is an $\varepsilon > 0$ and an infinite subset $\mathcal{K}_0$ of the integers such that

$$\|(x, y, z) - (x^{(k)}, y^{(k)}, z^{(k)})\| \geq \varepsilon \quad \text{for all } (x, y, z) \in \Omega \text{ and every } k \in \mathcal{K}_0. \tag{7.4}$$

Then, as a consequence of Lemma 7.7, the sequence $\{(x^{(k)}, y^{(k)}, z^{(k)})\}_{k \in \mathcal{K}_0}$ has at least one convergent subsequence. Thus, there is a point $(\hat{x}, \hat{y}, \hat{z})$ and an infinite subset $\overline{\mathcal{K}}_0 \subseteq \mathcal{K}_0$ such that $(x^{(k)}, y^{(k)}, z^{(k)}) \to (\hat{x}, \hat{y}, \hat{z})$ as $k \in \overline{\mathcal{K}}_0$ and $k \to \infty$.

But then, by letting $(\hat{x}, \hat{y}, \hat{z})$ play the role of (x^*, y^*, z^*) in the above lemmas, in particular Lemma 7.12, it follows that $(\hat{x}, \hat{y}, \hat{z})$ is a KKT point of the problem (7.2), and thus also a KKT point of the original problem (2.3). Thus, $(\hat{x}, \hat{y}, \hat{z}) \in \Omega$. By letting $(x, y, z) = (\hat{x}, \hat{y}, \hat{z})$ in (7.4), a contradiction has then been established. Therefore, the statement in Theorem 7.1 can not be false, but must be true. □

8. Test problems and numerical results. As mentioned in the introduction, a major benefit of CCSA methods is that they can be successfully applied to problems with a very large number of variables, even if the Hessian matrices of the objective and constraint functions are dense. Such problems often appear in, e.g., structural optimization, in particular in the subfield dealing with topology optimization. To illustrate this, we present two problems which are parameterized by the integer n = the number of variables x_j. The general structure of these problems resembles the corresponding structure of topology optimization problems (nonconvex problems with a large number of variables, upper and lower bounds on all variables, and a relatively small number of general inequality constraints); but in order to facilitate the reader's making her own numerical tests, the problems are not genuine structural optimization problems (which would require a finite element package) but are instead explicitly stated "academic" problems.

8.1. Three matrices which are used in the test problems. Let n be a given positive integer > 1 and let S, P, and Q be symmetric $n \times n$ matrices with elements given by

$$s_{ij} = \frac{2 + \sin(4\pi\alpha_{ij})}{(1 + |i - j|) \ln n}, \quad p_{ij} = \frac{1 + 2\alpha_{ij}}{(1 + |i - j|) \ln n}, \quad q_{ij} = \frac{3 - 2\alpha_{ij}}{(1 + |i - j|) \ln n},$$

where $\alpha_{ij} = \frac{i+j-2}{2n-2} \in [0, 1]$ for all i and j.

The matrices S, P, and Q are positive definite, and for $n = 9$ they look as follows:

$$S = \frac{1}{\ln 9}\begin{pmatrix}
2.0000 & 1.3536 & 1.0000 & 0.6768 & 0.4000 & 0.2155 & 0.1429 & 0.1616 & 0.2222 \\
1.3536 & 3.0000 & 1.3536 & 0.6667 & 0.3232 & 0.2000 & 0.2155 & 0.2857 & 0.3384 \\
1.0000 & 1.3536 & 2.0000 & 0.6464 & 0.3333 & 0.3232 & 0.4000 & 0.4512 & 0.4286 \\
0.6768 & 0.6667 & 0.6464 & 1.0000 & 0.6464 & 0.6667 & 0.6768 & 0.6000 & 0.4512 \\
0.4000 & 0.3232 & 0.3333 & 0.6464 & 2.0000 & 1.3536 & 1.0000 & 0.6768 & 0.4000 \\
0.2155 & 0.2000 & 0.3232 & 0.6667 & 1.3536 & 3.0000 & 1.3536 & 0.6667 & 0.3232 \\
0.1429 & 0.2155 & 0.4000 & 0.6768 & 1.0000 & 1.3536 & 2.0000 & 0.6464 & 0.3333 \\
0.1616 & 0.2857 & 0.4512 & 0.6000 & 0.6768 & 0.6667 & 0.6464 & 1.0000 & 0.6464 \\
0.2222 & 0.3384 & 0.4286 & 0.4512 & 0.4000 & 0.3232 & 0.3333 & 0.6464 & 2.0000
\end{pmatrix},$$

$$P = \frac{1}{\ln 9}\begin{pmatrix}
1.0000 & 0.5625 & 0.4167 & 0.3438 & 0.3000 & 0.2708 & 0.2500 & 0.2344 & 0.2222 \\
0.5625 & 1.2500 & 0.6875 & 0.5000 & 0.4062 & 0.3500 & 0.3125 & 0.2857 & 0.2656 \\
0.4167 & 0.6875 & 1.5000 & 0.8125 & 0.5833 & 0.4688 & 0.4000 & 0.3542 & 0.3214 \\
0.3438 & 0.5000 & 0.8125 & 1.7500 & 0.9375 & 0.6667 & 0.5312 & 0.4500 & 0.3958 \\
0.3000 & 0.4062 & 0.5833 & 0.9375 & 2.0000 & 1.0625 & 0.7500 & 0.5938 & 0.5000 \\
0.2708 & 0.3500 & 0.4688 & 0.6667 & 1.0625 & 2.2500 & 1.1875 & 0.8333 & 0.6562 \\
0.2500 & 0.3125 & 0.4000 & 0.5312 & 0.7500 & 1.1875 & 2.5000 & 1.3125 & 0.9167 \\
0.2344 & 0.2857 & 0.3542 & 0.4500 & 0.5938 & 0.8333 & 1.3125 & 2.7500 & 1.4375 \\
0.2222 & 0.2656 & 0.3214 & 0.3958 & 0.5000 & 0.6562 & 0.9167 & 1.4375 & 3.0000
\end{pmatrix},$$

$$Q = \frac{1}{\ln 9}\begin{pmatrix}
3.0000 & 1.4375 & 0.9167 & 0.6562 & 0.5000 & 0.3958 & 0.3214 & 0.2656 & 0.2222 \\
1.4375 & 2.7500 & 1.3125 & 0.8333 & 0.5938 & 0.4500 & 0.3542 & 0.2857 & 0.2344 \\
0.9167 & 1.3125 & 2.5000 & 1.1875 & 0.7500 & 0.5312 & 0.4000 & 0.3125 & 0.2500 \\
0.6562 & 0.8333 & 1.1875 & 2.2500 & 1.0625 & 0.6667 & 0.4688 & 0.3500 & 0.2708 \\
0.5000 & 0.5938 & 0.7500 & 1.0625 & 2.0000 & 0.9375 & 0.5833 & 0.4062 & 0.3000 \\
0.3958 & 0.4500 & 0.5312 & 0.6667 & 0.9375 & 1.7500 & 0.8125 & 0.5000 & 0.3438 \\
0.3214 & 0.3542 & 0.4000 & 0.4688 & 0.5833 & 0.8125 & 1.5000 & 0.6875 & 0.4167 \\
0.2656 & 0.2857 & 0.3125 & 0.3500 & 0.4062 & 0.5000 & 0.6875 & 1.2500 & 0.5625 \\
0.2222 & 0.2344 & 0.2500 & 0.2708 & 0.3000 & 0.3438 & 0.4167 & 0.5625 & 1.0000
\end{pmatrix}.$$

8.2. Problem 1. In the first considered problem, called Problem 1, the objective function is strictly convex, but the nonlinear constraint functions are strictly concave so the set of feasible solutions is nonconvex. The formulation of the problem is as follows:

$$\begin{aligned}
&\text{minimize} && f_0(x) = x^T S x \\
&\text{subject to} && f_1(x) = \frac{n}{2} - x^T P x \le 0, \\
&&& f_2(x) = \frac{n}{2} - x^T Q x \le 0, \\
&&& -1 \le x_j \le 1, \quad j = 1, \dots, n,
\end{aligned} \tag{8.1}$$

with starting point $x^{(0)} = (0.5,\ 0.5, \dots,\ 0.5)^T$.

8.3. Problem 2. In the second considered problem, called Problem 2, the nonlinear constraint functions are strictly convex, but the objective function is strictly concave and thus nonconvex. The formulation of the problem is as follows.

$$\begin{aligned} \text{minimize} \quad & f_0(x) = -x^T S x \\ \text{subject to} \quad & f_1(x) = x^T P x - \frac{n}{2} \le 0, \\ & f_2(x) = x^T Q x - \frac{n}{2} \le 0, \\ & -1 \le x_j \le 1, \quad j = 1, \ldots, n, \end{aligned} \tag{8.2}$$

with starting point $x^{(0)} = (0.25,\ 0.25, \ldots,\ 0.25)^T$.

8.4. Numerical results. We have used the CCSA method based on MMA approximations (see Example 5.4 in section 5) to solve the above two problems with $n = 1000$, 2000, 5000, 10000, and 20000. Both problems are of the form (2.1), and they were first transformed to the form (2.3) with $a_0 = 1$, $a_1 = a_2 = 0$, $d_1 = d_2 = 1$, and $c_1 = c_2 = 1000$. It then turned out that $y = 0$ and $z = 0$ in the optimal solution of each generated CCSA subproblem.

Concerning the termination criterion that we used, first note that the KKT conditions of the considered problems (8.1) and (8.2) can be written as follows, using the notations $a^+ = \max\{0, a\}$ and $a^- = \max\{0, -a\}$:

$$(1 + x_j)\left(\frac{\partial f_0}{\partial x_j} + \lambda_1 \frac{\partial f_1}{\partial x_j} + \lambda_2 \frac{\partial f_2}{\partial x_j}\right)^+ = 0, \quad j = 1, \ldots, n, \tag{8.3a}$$

$$(1 - x_j)\left(\frac{\partial f_0}{\partial x_j} + \lambda_1 \frac{\partial f_1}{\partial x_j} + \lambda_2 \frac{\partial f_2}{\partial x_j}\right)^- = 0, \quad j = 1, \ldots, n, \tag{8.3b}$$

$$f_i(x)^+ = 0, \quad i = 1, 2, \tag{8.3c}$$

$$\lambda_i f_i(x)^- = 0, \quad i = 1, 2, \tag{8.3d}$$

$$\lambda_i \ge 0, \quad i = 1, 2, \tag{8.3e}$$

$$-1 \le x_j \le 1, \quad j = 1, \ldots, n. \tag{8.3f}$$

Equations (8.3a)–(8.3d) can be written more concisely as $r_k(x, \lambda) = 0$, $k = 1, \ldots, 2n + 4$. The inequalities (8.3e) and (8.3f) are always satisfied by the primal variables x_j and the dual variables λ_i obtained from the solution of the CCSA subproblem. The outer iterations were terminated when these x and λ also satisfied

$$\frac{1}{n} \sum_{k=1}^{2n+4} (r_k(x, \lambda))^2 \le 10^{-10}. \tag{8.4}$$

A similar, but harder, termination criterion was used when solving the CCSA subproblems; a subproblem was considered as solved when a condition corresponding to (8.4) was satisfied with the right-hand side equal to 10^{-16}.

The optimal solutions obtained for the case $n = 1000$ are plotted in Figures 8.1 and 8.2, with the index j on the horizontal axis and x_j on the vertical axis.

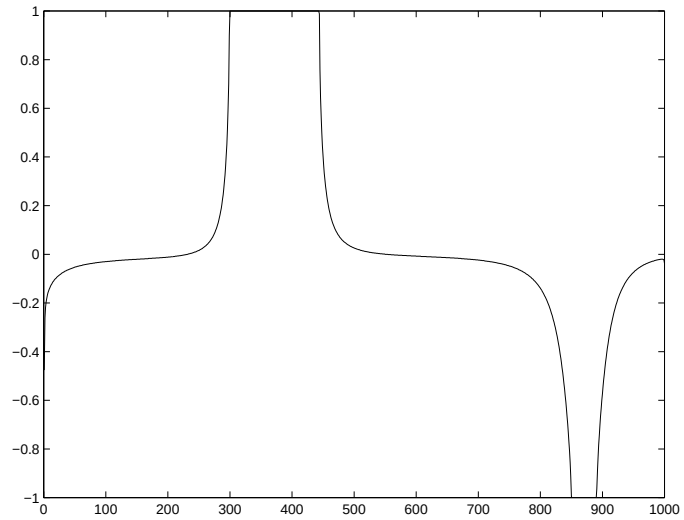


FIG. 8.1. *Obtained x_j for Problem* 1 *with* $n = 1000$.

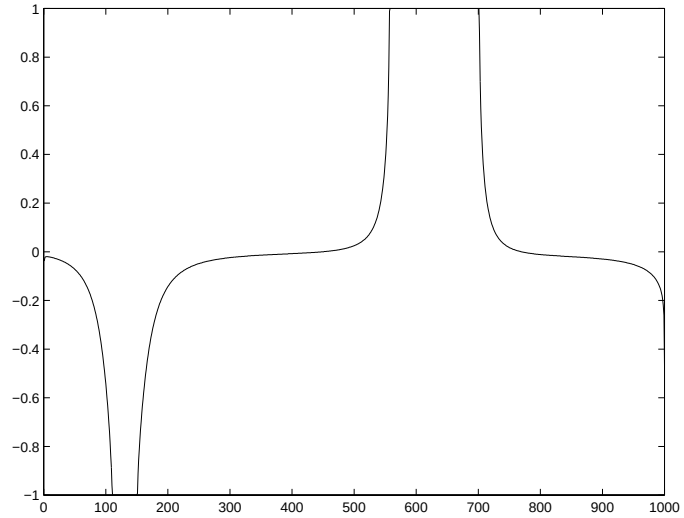


FIG. 8.2. *Obtained x_j for Problem* 2 *with* $n = 1000$.

Additional results are presented in Tables 8.1 and 8.2, where for each problem we present the number of variables (n), the objective value of the obtained optimal solution, the number of variables which are at the upper or lower bound in the obtained optimal solution, the obtained values of the two Lagrange multipliers λ_1 and λ_2, the total number of required outer iterations, and the total number of additionally required inner iterations.

The method was implemented in Fortran 77 on a Sun Enterprise 4000 (using only one of the four processors). The total required CPU-time was approximately $2n^2/10^6$ CPU-minutes for Problem 1 and approximately $5n^2/10^6$ CPU-minutes for Problem 2. Most of this CPU-time was spent calculating function values and gradients of $f_0(x)$,

TABLE 8.1
Results for Problem 1.

Number of variables	Objective value	Variables at bounds	λ_1	λ_2	Total number of outer iter.	Total number of inner iter.
1000	260.85	184	0.138	0.451	177	209
2000	523.51	353	0.147	0.442	190	224
5000	1312.05	840	0.156	0.431	221	263
10000	2626.76	1629	0.161	0.425	251	296
20000	5256.56	3184	0.165	0.420	286	316

TABLE 8.2
Results for Problem 2.

Number of variables	Objective value	Variables at bounds	λ_1	λ_2	Total number of outer iter.	Total number of inner iter.
1000	-739.15	184	0.549	0.862	436	415
2000	-1476.49	353	0.558	0.853	465	471
5000	-3687.95	840	0.569	0.844	584	606
10000	-7373.24	1629	0.575	0.839	682	704
20000	-14743.44	3182	0.580	0.835	793	816

$f_1(x)$, and $f_2(x)$, while only a minor part was spent solving the CCSA subproblems. It should be noted that the matrices S, P, and Q are never stored. Instead, the elements s_{ij}, p_{ij}, and q_{ij} are generated as needed when calculating function values and gradients of $f_0(x)$, $f_1(x)$, and $f_2(x)$ at a given iteration point.

It could finally be mentioned that it is virtually impossible to solve the considered problems by, e.g., an SQP method. The approximate Hessian matrix (of the Lagrange function) simply becomes too big.

9. Conclusions. A class of optimization methods based on the concept of conservative convex separable approximations has been presented. Global convergence has been theoretically proved, and it has been demonstrated that the methods work numerically.

We do not claim that a CCSA method is always the natural choice, but for certain problems it is certainly a competitive alternative. This is typically the case for problems with a very large number of variables and a relatively small number of general inequality constraints, in particular if it is desirable that the iteration points remain feasible.

Finally, it could be noted that if the considered problem also contains some linear constraints, these can simply be included as (exactly the same) linear constraints in the CCSA subproblems. Since exact approximations are conservative approximations, the global convergence properties of the methods will not be altered.

Acknowledgments. I am most grateful to my colleague Anders Forsgren for numerous valuable discussions. I also want to thank the two anonymous referees for their constructive criticism of an early version of the paper and for valuable suggestions on how to improve the presentation.

REFERENCES

[1] A. V. FIACCO AND G. P. MCCORMICK, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley, New York, 1968.

[2] R. Fletcher, *Practical Methods of Optimization*, 2nd ed., John Wiley, Chichester, England, 1987.
[3] A. Forsgren and P. E. Gill, *Primal-dual interior methods for nonconvex nonlinear programming*, SIAM J. Optim., 8 (1998), pp. 1132–1152.
[4] D. Gay, M. Overton, and M. Wright, *A primal-dual interior method for nonconvex nonlinearly constrained optimization*, in Advances in Nonlinear Programming, Y. Yuan, ed., Kluwer, Dordrecht, The Netherlands, 1998, pp. 31–56.
[5] P. Gill, W. Murray, M. Saunders, and M. Wright, *Constrained nonlinear programming*, in Optimization, Handbooks Oper. Res. Management Sci. 1, G. Nemhauser, A. R. Kan, and M. Todd, eds., North–Holland, Amsterdam, 1989, pp. 171–210.
[6] D. G. Luenberger, *Optimization by Vector Space Methods*, John Wiley, New York, 1968.
[7] K. Svanberg, *The method of moving asymptotes—a new method for structural optimization*, Internat. J. Numer. Methods Engrg., 24 (1987), pp. 359–373.
[8] K. Svanberg, *A globally convergent version of MMA without linesearch*, in Proceedings of the First World Congress of Structural and Multidisciplinary Optimization, N. Olhoff and G. Rozvany, eds., Pergamon Press, Elmsford, NY, 1995, pp. 9–16.

SIAM J. OPTIM.
Vol. 12, No. 3, pp. 575–582

INTEGRATION OF FENCHEL SUBDIFFERENTIALS OF EPI-POINTED FUNCTIONS*

JOËL BENOIST† AND ARIS DANIILIDIS‡

Abstract. It is shown that in finite dimensions Rockafellar's technique of integrating cyclically monotone operators, applied to the Fenchel subdifferential of an epi-pointed function, yields the closed convex hull of that function.

Key words. Fenchel subdifferential, cyclically monotone operator, integration, epi-pointed function

AMS subject classifications. Primary, 52A41, 47H05; Secondary, 26E25

PII. S0152623400381279

1. Introduction.

By the term *integration of a multivalued operator* $T : \mathbb{R}^d \rightrightarrows \mathbb{R}^d$, we mean the problem of finding a lower semicontinuous (lsc) function f such that $T \subseteq \partial f$, where ∂f corresponds to some notion of subdifferential for the function f. This problem has recently attracted researchers' interest; see, for instance, [3], [5], [6], [9], and references therein.

If we impose the further restriction that ∂f is the Fenchel subdifferential (defined below), then a complete answer (even in infinite dimensions) to the aforementioned problem has been established by Rockafellar [7], with the introduction of the class of cyclically monotone operators. Indeed, as shown in [7] (see also [4]), every such operator T is included in the subdifferential ∂f of an lsc convex function f. In particular, T coincides with ∂f if and only if it is maximal, and in such a case f is unique up to a constant.

In dealing with the above problem, Rockafellar used a technique consisting of a formal construction of an lsc convex function f_T started from a given cyclically monotone operator T. The function f_T is further called the *convex integral* of T. Let us recall that Fenchel subdifferentials are particular cases of cyclically monotone operators. Consequently, for every lsc function f with $\operatorname{dom} \partial f \neq \emptyset$, the convex integral $f_{\partial f}$ (also denoted $\widehat{f}$ in this paper) of its subdifferential ∂f naturally defines an lsc convex function minorizing f. If in particular f is convex, then the convex integral $\widehat{f}$ is equal to f up to a constant [7]. In the general case, a natural question arises:

($\mathcal{Q}$) Given an lsc function f, is $\widehat{f}$ equal to the closed convex hull $\overline{\operatorname{co}}\, f$ of f?

This question was first considered in [1, Proposition 2.6], where the authors provided a positive answer (in finite dimensions) for the class of *strongly coercive functions*, that is, functions satisfying

$$\lim_{\|x\| \to \infty} \frac{f(x)}{\|x\|} = +\infty. \tag{1.1}$$

In this paper we improve the above result by establishing the same conclusion for the larger class of *epi-pointed functions* introduced in [2] (see definition below).

*Received by the editors November 16, 2000; accepted for publication (in revised form) March 21, 2001; published electronically January 9, 2002.
http://www.siam.org/journals/siopt/12-3/38127.html

†Faculté des Sciences, LACO, URA CNRS 1586, Université de Limoges, 123 avenue Albert Thomas, 87060 Limoges, Cedex, France (benoist@unilim.fr).

‡Laboratoire de Mathématiques Appliquées, CNRS ERS 2055, Université de Pau et des Pays de l'Adour, avenue de l'Université, 64000 Pau, France (aris.daniilidis@univ-pau.fr).

Moreover, we shall give an easy example of a non-epi-pointed function for which $(\mathcal{Q})$ is no longer valid. However, for the one-dimensional case ($d = 1$), we shall show that $(\mathcal{Q})$ holds true for every lsc function defined on $\mathbb{R}$.

The paper is organized as follows. In the next section, we fix our notation and give some preliminaries concerning Fenchel duality and convex integration of the (Fenchel) subdifferential of a nonconvex function. The result of [1] for the class of strongly coercive functions is recalled, and an example where the convex integration does not yield the closed convex hull of the function is illustrated. Finally, in section 3 we state and prove the main result of this article, concerning the class of epi-pointed functions.

2. Convex integration. Throughout this paper we consider the Euclidean space $\mathbb{R}^d$ equipped with the usual scalar product $\langle\cdot,\cdot\rangle$. In what follows, we denote by $f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ a lsc function which is proper, that is $\operatorname{dom} f := \{x \in \mathbb{R}^d : f(x) \in \mathbb{R}\}$ is nonempty. We also denote by $\operatorname{epi} f$ the epigraph of f, that is the set $\{(x,t) \in \mathbb{R}^d \times \mathbb{R} : f(x) \le t\}$. We recall that the second conjugate $\overline{\operatorname{co}} f$ (also denoted by f^{**}) of f is given by

$$\overline{\operatorname{co}} f(x) = \sup_{x^* \in \mathbb{R}^d} \{\langle x^*, x\rangle - f^*(x^*)\}, \tag{2.1}$$

where

$$f^*(x^*) = \sup_{x \in \mathbb{R}^d} \{\langle x^*, x\rangle - f(x)\}. \tag{2.2}$$

It is known that $\overline{\operatorname{co}} f$ is the greatest lsc convex function majorized by f, and that its epigraph coincides with the closed convex hull of the epigraph of f. By the term subdifferential we shall always mean the Fenchel subdifferential ∂f, defined for every $x \in \operatorname{dom} f$ as follows

$$\partial f(x) = \{x^* \in \mathbb{R}^d : f(y) \ge f(x) + \langle x^*, y - x\rangle \ \ \forall y \in \mathbb{R}^d\}. \tag{2.3}$$

If $x \in \mathbb{R}^d \setminus \operatorname{dom} f$, we set $\partial f(x) = \emptyset$. Throughout this paper, the set

$$\operatorname{dom} \partial f := \{x \in \mathbb{R}^d : \partial f(x) \neq \emptyset\}$$

is assumed to be nonempty. Further, let x_0 denote an arbitrary point of $\operatorname{dom} \partial f$. We call convex integral of ∂f the lsc convex function $\widehat{f} : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ defined for all $x \in \mathbb{R}^d$ by the formula

$$\widehat{f}(x) := \ f(x_0) + \ \sup \left\{ \sum_{i=0}^{n-1} \langle x_i^*, x_{i+1} - x_i\rangle \ + \ \langle x_n^*, x - x_n\rangle \right\}, \tag{2.4}$$

where the supremum is taken for all $n \ge 1$, all $x_1, x_2, \dots, x_n$ in $\operatorname{dom} \partial f$, and all $x_0^* \in \partial f(x_0), x_1^* \in \partial f(x_1), \dots, x_n^* \in \partial f(x_n)$. According to (2.3), we can easily check that $\widehat{f} \le f$, and consequently f is proper and

$$\widehat{f} \le \overline{\operatorname{co}} f. \tag{2.5}$$

Rockafellar [8] has shown that if f is in particular convex, then the convex integral $\widehat{f}$ of ∂f is equal to f, that is

$$\widehat{f} = f. \tag{2.6}$$

In [1, Proposition 2.6] the authors generalized (2.6) to the nonconvex case by showing that if f is strongly coercive (that is f satisfies (1.1)), then (2.5) becomes

$$\widehat{f} = \overline{\text{co}}\, f.$$

However, the exact relation between $\widehat{f}$ and $\overline{\text{co}}\, f$ for a function not satisfying (1.1) remains to be discovered. In particular, while in one-dimensional spaces we always have $\widehat{f} = \overline{\text{co}}\, f$ (see Corollary 3.7), the following simple counterexample shows that this is not the case in general.

Example 2.1. Let $f : \mathbb{R}^2 \to \mathbb{R}$ be defined as follows:

$$f(a,b) = \begin{cases} \exp(-a^2) + \frac{1}{2}b^2 & \text{if } (a,b) \neq (0,0), \\ 0 & \text{if } (a,b) = (0,0). \end{cases}$$

We can easily check that

$$f^*(a,b) = \begin{cases} \frac{1}{2}b^2 & \text{if } a = 0, \\ +\infty & \text{if } a \neq 0 \end{cases}$$

and that

$$\overline{\text{co}}\, f(a,b) = \frac{1}{2}b^2.$$

On the other hand, since

$$\partial f(a,b) = \begin{cases} \{0\} & \text{if } (a,b) = (0,0), \\ \emptyset & \text{if } (a,b) \neq (0,0), \end{cases}$$

formula (2.4) yields (for $x_0 = (0,0)$) that $\widehat{f}(x) = 0$ for all $x \in \mathbb{R}^2$. Hence $\widehat{f} \neq \overline{\text{co}}\, f$.

Remark. Appropriately modifying the function f around the origin, we can obtain a continuous function $g : \mathbb{R}^2 \to \mathbb{R}$ such that $\widehat{g} \neq \overline{\text{co}}\, g$.

Let us also remark that in the previous example we have

$$\text{int}\,(\text{dom}\, f^*) = \emptyset. \tag{2.7}$$

It will follow from the main theorem of section 3 that (2.7) is in fact a necessary condition for obtaining such examples.

3. Epi-pointed functions. The aim of this section is to establish the equality between the convex integral $\widehat{f}$ of ∂f and the closed convex hull $\overline{\text{co}} f$ of f for the class of proper, lsc, and epi-pointed functions defined in $\mathbb{R}^d$.

Let us recall the following definition [2].

DEFINITION 3.1. *The function f is called epi-pointed if* $\text{int}\,(\text{dom}\, f^*) \neq \emptyset$.

It follows easily (see [2, Proposition 4.5 (iv)]) that every strongly coercive function is epi-pointed. Note also that for every $\overline{x}^* \in \text{int}\,(\text{dom}\, f^*)$ we can always find $\overline{x} \in \mathbb{R}^d$ such that $f^*(\overline{x}^*) = \langle \overline{x}^*, \overline{x} \rangle - f(\overline{x})$ (that is the "sup" in (2.2) is attained). This obviously yields that $\overline{x}^* \in \partial f(x) \cap \text{int}\,(\text{dom}\, f^*)$. In particular, if f is epi-pointed the set $\text{dom}\, \partial f$ is nonempty. If now x_0 is any point of $\text{dom}\, \partial f$, we can consider the lsc convex function $\tilde{f}$ defined for all $x \in \mathbb{R}^d$ by

$$\tilde{f}(x) = f(x_0) + \sup \left\{ \sum_{i=0}^{n-1} \langle x_i^*, x_{i+1} - x_i \rangle \; + \; \langle x_n^*, x - x_n \rangle \right\}, \tag{3.1}$$

where the supremum is taken for all $n \geq 1$, all $x_1, x_2, \ldots, x_n$ in $\mathbb{R}^d$, all $x_0^* \in \partial f(x_0)$, and all

$$x_i^* \in \partial f(x_i) \cap \operatorname{int}(\operatorname{dom} f^*),$$

where $i \in \{1, \ldots, n\}$. Note that whenever f is epi-pointed, the set

$$\{x \in \mathbb{R}^d : \partial f(x) \cap \operatorname{int}(\operatorname{dom} f^*) \neq \emptyset\}$$

is nonempty, so that $\tilde{f}$ is proper. Comparing formulas (2.4) and (3.1) we immediately conclude that

$$\tilde{f} \leq \widehat{f}.$$

We shall show that if the function f is convex and epi-pointed, then f is equal to $\tilde{f}$ and so, in view of (2.6), the previous inequality becomes an equality. This is the context of Proposition 3.3 below.

We shall first need the following lemma.

LEMMA 3.2. *Suppose that f is lsc convex and epi-pointed. Then we have the inclusion*

$$\partial f^*(x^*) \subseteq \partial \tilde{f}^*(x^*) \quad \textit{on} \ \operatorname{int}(\operatorname{dom} f^*).$$

Proof. A classic result (see [8]) states that for the lsc convex function f and all $x, x^* \in \mathbb{R}^d$ we have

$$x \in \partial f^*(x^*) \quad \text{if and only if} \quad x^* \in \partial f(x).$$

Similarly, for the lsc convex function $\tilde{f}$,

$$x \in \partial \tilde{f}^*(x^*) \quad \text{if and only if} \quad x^* \in \partial \tilde{f}(x).$$

Let $x^* \in \operatorname{int}(\operatorname{dom} f^*)$ and $x \in \partial f^*(x^*)$. We shall show that $x \in \partial \tilde{f}^*(x^*)$. It follows that

$$x^* \in \partial f(x) \cap \operatorname{int}(\operatorname{dom} f^*). \tag{3.2}$$

For any $t < \tilde{f}(x)$, using formula (3.1), we may choose $x_1, \ldots, x_n$ in $\mathbb{R}^d$, $x_0^* \in \partial f(x_0)$, and $x_1^* \in \partial f(x_1) \cap \operatorname{int}(\operatorname{dom} f^*), \ldots, x_n^* \in \partial f(x_n) \cap \operatorname{int}(\operatorname{dom} f^*)$ such that

$$t < f(x_0) + \sum_{i=0}^{n-1} \langle x_i^*, x_{i+1} - x_i \rangle + \langle x_n^*, x - x_n \rangle. \tag{3.3}$$

For any $y \in \mathbb{R}^d$, adding to both sides of (3.3) the quantity $\langle x^*, y - x \rangle$, we obtain

$$t + \langle x^*, y - x \rangle < f(x_0) + \sum_{i=0}^{n-1} \langle x_i^*, x_{i+1} - x_i \rangle + \langle x_n^*, x - x_n \rangle + \langle x^*, y - x \rangle. \tag{3.4}$$

In view of (3.1), the right part of (3.4) is always less than or equal to $\tilde{f}(y)$. Letting $t \to \tilde{f}(x)$, we infer

$$\tilde{f}(x) + \langle x^*, y - x \rangle \leq \tilde{f}(y),$$

which yields $x^* \in \partial \tilde{f}(x)$, or, equivalently, $x \in \partial \tilde{f}^*(x^*)$. ∎

PROPOSITION 3.3. *If f is lsc convex and epi-pointed, then $\tilde{f} = f$.*

Proof. Since the functions f^* and $\tilde{f}^*$ are proper, lsc, and convex, we deduce from [8] and Lemma 3.2 that

$$f^* = \tilde{f}^* + k \quad \text{on } \operatorname{int}(\operatorname{dom} f^*) \tag{3.5}$$

for some constant $k \in \mathbb{R}$.

Let us now prove that the equality in (3.5) can be extended to all $\mathbb{R}^d$. According to [7, Corollary 7.3.4], it suffices to prove that the relative interiors of the convex sets $\operatorname{dom} f^*$ and $\operatorname{dom} \tilde{f}^*$ are equal or, equivalently (since $\operatorname{int}(\operatorname{dom} f^*)$ is nonempty), that

$$\operatorname{int}(\operatorname{dom} f^*) = \operatorname{int}(\operatorname{dom} \tilde{f}^*). \tag{3.6}$$

Let us now prove this last equality. Taking conjugates in both sides of the inequality $\tilde{f} \le f$ we obtain $f^* \le \tilde{f}^*$; hence in particular

$$\operatorname{dom} \tilde{f}^* \subseteq \operatorname{dom} f^*,$$

and so

$$\operatorname{int}(\operatorname{dom} \tilde{f}^*) \subseteq \operatorname{int}(\operatorname{dom} f^*). \tag{3.7}$$

Conversely, let $x^* \in \operatorname{int}(\operatorname{dom} f^*)$. Since f^* is convex, we have $\partial f^*(x^*) \neq \emptyset$. By Lemma 3.2 we get $\partial \tilde{f}^*(x^*) \neq \emptyset$, yielding that $x^* \in \operatorname{dom} \partial \tilde{f}^*$. It follows that

$$\operatorname{int}(\operatorname{dom} f^*) \subseteq \operatorname{dom} \tilde{f}^*. \tag{3.8}$$

Combining (3.7) with (3.8), we conclude that equality (3.6) holds as desired. Hence we obtain

$$f^* = \tilde{f}^* + k.$$

Taking conjugates, this last equality yields $f = \tilde{f} - k$. Since $f(x_0) = \tilde{f}(x_0)$, we conclude that $k = 0$ and thus $f = \tilde{f}$. ∎

We shall finally need the following lemma.

LEMMA 3.4. *Suppose that f is lsc and epi-pointed, and set $g = \overline{\operatorname{co}}\, f$. Then for any $x \in \operatorname{dom} \partial f$ and $x^* \in \partial g(x) \cap \operatorname{int}(\operatorname{dom} f^*)$ there exist $y_1, \dots, y_p$ in $\mathbb{R}^d$ such that $x \in \operatorname{co}\{y_1, y_2, \dots, y_p\}$ and*

$$x^* \in \bigcap_{i=1}^{p} \partial f(y_i).$$

Proof. From [2, Theorem 4.6] we conclude that for any $x^* \in \partial g(x)$ there exist $y_1, \dots, y_p$ in $\mathbb{R}^d$ and $w_1, \dots, w_q$ in $\mathbb{R}^d \setminus \{0\}$ such that

$$x - \sum_{j=1}^{q} w_j \ \in \ \operatorname{co}\{y_1, y_2, \dots, y_p\}$$

and

$$x^* \in \left[\bigcap_{i=1}^{p} \partial f(y_i)\right] \cap \left[\bigcap_{j=1}^{q} \partial f_\infty(w_j)\right], \tag{3.9}$$

where f_∞ is defined via the relation $\operatorname{epi}(f_\infty) = (\operatorname{epi} f)_\infty$, where

$$(\operatorname{epi} f)_\infty := \left\{ d \in X : \exists \{x_n\}_{n\geq 1} \text{ in } \operatorname{epi} f, \exists \{t_n\} \searrow 0^+ \text{ with } d = \lim_{n\to+\infty} t_n x_n \right\}.$$

It suffices to show that for $x^* \in \operatorname{int}(\operatorname{dom} f^*)$, (3.9) yields $q = 0$. In order to find a contradiction, suppose that $q \neq 0$. Since the function f_∞ is sublinear positively homogeneous and $f_\infty(0) = 0$ (e.g., [2]), it follows easily that for any $w_j \neq 0$ and any $x^* \in \partial f_\infty(w_j)$ we have $\langle x^*, w_j\rangle = f_\infty(w_j)$. Since $x^* \in \operatorname{int}(\operatorname{dom} f^*)$, we may find some $z^* \in \mathbb{R}^d$ (near x^*) such that $z^* \in \operatorname{int}(\operatorname{dom} f^*)$ and $\langle z^*, w_j\rangle > f_\infty(w_j)$. The latter yields easily that

$$z^* \notin \partial f_\infty(0). \tag{3.10}$$

On the other hand, since $z^* \in \operatorname{int}(\operatorname{dom} f^*) \subseteq \operatorname{dom} \partial f^*$, we conclude the existence of x in $\mathbb{R}^d$ such that $x \in \partial f^*(z^*)$, or, equivalently,

$$z^* \in \partial g(x). \tag{3.11}$$

Since $\partial g(x) \subseteq \partial f_\infty(0)$ [2, Theorem 4.6], relations (3.10) and (3.11) give the contradiction. □

We are now ready to establish the main result of this section.

THEOREM 3.5. *If f is lsc and epi-pointed, then $\widehat{f} = \overline{\operatorname{co}}\, f$.*

Proof. Set $g = \overline{\operatorname{co}}(f)$. Then g is lsc convex and $\operatorname{int}(\operatorname{dom} g^*) = \operatorname{int}(\operatorname{dom} f^*)$. In particular, g is epi-pointed. Using Proposition 3.3 we conclude that

$$g(x) = g(x_0) + \sup\left\{ \sum_{i=0}^{n-1} \langle x_i^*, x_{i+1} - x_i\rangle \ + \ \langle x_n^*, x - x_n\rangle \right\},$$

where the supremum is taken over all $n \geq 1$, all $x_1, \ldots, x_n$ in $\mathbb{R}^d$, all $x_0^* \in \partial g(x_0)$, and all

$$x_i^* \in \partial g(x_i) \cap \operatorname{int}(\operatorname{dom} f^*),$$

where $i \in \{1, \ldots, n\}$. Take any $x \in \mathbb{R}^d$ and any $t < g(x)$. Then there exist $x_1, \ldots, x_n$ in $\mathbb{R}^d$, $x_0^* \in \partial g(x_0)$, and $x_i^* \in \partial g(x_i) \cap \operatorname{int}(\operatorname{dom} f^*)$ (for $i = 1$ to n) such that

$$t < g(x_0) + \sum_{i=0}^{n-1} \langle x_i^*, x_{i+1} - x_i\rangle \ + \ \langle x_n^*, x - x_n\rangle. \tag{3.12}$$

Recalling that $x_0 \in \operatorname{dom} \partial f$, we easily check that $g(x_0) = f(x_0)$ and $\partial g(x_0) = \partial f(x_0)$. On the other hand, for all $i \in \{1, \ldots, n\}$ Lemma 3.4 guarantees the existence of points $y_i^1, \ldots, y_i^{p_i}$ in $\mathbb{R}^d$ such that $x_i \in \operatorname{co}\{y_i^1, y_i^2, \ldots, y_i^{p_i}\}$ and

$$x_i^* \in \bigcap_{j=1}^{p_i} \partial f\left(y_i^j\right).$$

We claim that, for $i = 1$, there exists an index j_1 in $\{1, 2, \ldots, p_1\}$ such that

$$\langle x_0^*, x_1 - x_0\rangle + \langle x_1^*, x_2 - x_1\rangle \leq \langle x_0^*, y_1^{j_1} - x_0\rangle + \langle x_1^*, x_2 - y_1^{j_1}\rangle.$$

Indeed, if this were not the case, then for every j we would have

$$\langle x_0^*, x_1 - x_0\rangle + \langle x_1^*, x_2 - x_1\rangle > \langle x_0^*, y_1^j - x_0\rangle + \langle x_1^*, x_2 - y_1^j\rangle.$$

This yields a contradiction, since $x_1 \in \operatorname{co}\{y_1^1, \dots, y_1^{p_1}\}$.

Proceeding like this for $i \geq 1$, we inductively replace all x_i's in (3.12) by $y_i^{j_i}$'s in a way that $x_i^* \in \partial f(y_i^{j_i})$, thus obtaining the formula

$$t < f(x_0) + \langle x_0^*, y_1^{j_1} - x_0\rangle + \langle x_1^*, y_2^{j_2} - y_1^{j_1}\rangle + \cdots + \langle x_n^*, x - y_n^{j_n}\rangle.$$

Comparing with (2.4), we obtain $t < \widehat{f}(x)$. Letting $t \to g(x)$ we infer $g(x) = \overline{\operatorname{co}}\, f(x) \leq \widehat{f}(x)$, which finishes the proof in view of (2.5). ∎

COROLLARY 3.6. *Suppose that f, h are proper lsc and epi-pointed functions. If $\partial f = \partial h$, then $\overline{\operatorname{co}}\, f$ and $\overline{\operatorname{co}}\, h$ are equal up to a constant.*

Proof. For $x_0 \in \operatorname{dom}\partial f$ and $c = g(x_0) - f(x_0)$ we obviously have $\widehat{f} = \widehat{h} + c$, which, in view of Theorem 3.5, yields $\overline{\operatorname{co}}\, f = \overline{\operatorname{co}}\, h + c$. ∎

The class of proper, lsc, and epi-pointed functions is not minimal, in order to ensure the conclusion of Theorem 3.5. For example, every constant function f satisfies $\widehat{f} = \overline{\operatorname{co}}\, f = f$, and obviously $\operatorname{dom} f^* = \{0\}$. (In fact, one can consider any lsc convex function f which is not epi-pointed.) Furthermore, the example of the function $f(x) = \min\{\|x\|, 1\}$ shows that the conclusion $\widehat{f} = \overline{\operatorname{co}}\, f$ might be true even in cases where f is nonconvex and non-epi-pointed at the same time. In particular, in one-dimensional spaces the following result is true.

COROLLARY 3.7. *If $d = 1$ (that is $f : \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$) and $\operatorname{dom}\partial f \neq \emptyset$, then $\widehat{f} = \overline{\operatorname{co}}\, f$.*

Proof. In view of Theorem 3.5, it suffices to consider only the case $\operatorname{int}(\operatorname{dom} f^*) = \emptyset$. Since f^* is convex (and $\operatorname{dom}\partial f \neq \emptyset$) it follows that $\operatorname{dom} f^* = \{\alpha\}$ for some $\alpha \in \mathbb{R}$. We easily conclude from (2.1) that

$$\overline{\operatorname{co}}\, f(x) = \alpha x - f^*(\alpha) \tag{3.13}$$

for all $x \in \mathbb{R}$. On the other hand, for any $x_0 \in \operatorname{dom}\partial f$ we have $\partial f(x_0) = \{\alpha\}$, which yields, in view of (2.2) and (2.3), that

$$f^*(\alpha) = \alpha x_0 - f(x_0). \tag{3.14}$$

Finally, it follows easily from relation (2.4) that

$$\widehat{f}(x) = f(x_0) + \alpha(x - x_0). \tag{3.15}$$

Relations (3.13), (3.14), and (3.15) directly yield $\widehat{f} = \overline{\operatorname{co}}\, f$. ∎

REFERENCES

[1] M. BACHIR, A. DANIILIDIS, AND J.-P. PENOT, *Lower subdifferentiability and integration*, Set-Valued Anal., to appear.

[2] J. BENOIST AND J.-B. HIRIART-URRUTY, *What is the subdifferential of the closed convex hull of a function?*, SIAM J. Math. Anal., 27 (1996), pp. 1661–1679.

[3] J. BORWEIN, W. MOORS, AND Y. SHAO, *Subgradient representation of multifunctions*, J. Austral. Math. Soc. Ser. B, 40 (1998), pp. 1–13.

[4] R. PHELPS, *Convex Functions, Monotone Operators and Differentiability*, 2nd ed., Springer-Verlag, Berlin, 1991.

[5] R. Poliquin, *Integration of subdifferentials of nonconvex functions*, Nonlinear Anal., 17 (1991), pp. 385–398.
[6] L. Qi, *The maximal normal operator space and integration of subdifferentials of nonconvex functions*, Nonlinear Anal., 13 (1989), pp. 1003–1011.
[7] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.
[8] R. T. Rockafellar, *On the maximal monotonicity of subdifferential mappings*, Pacific J. Math., 33 (1970), pp. 209–216.
[9] L. Thibault and D. Zagrodny, *Integration of subdifferentials of lower semi-continuous functions on Banach spaces*, J. Math. Anal. Appl., 189 (1995), pp. 33–58.

SIAM J. OPTIM.
Vol. 12, No. 3, pp. 583–605

A HOMOTOPY-BASED ALGORITHM FOR MIXED COMPLEMENTARITY PROBLEMS*

STEPHEN C. BILLUPS†

Abstract. This paper develops an algorithm for solving mixed complementarity problems that is based upon probability-one homotopy methods. After the complementarity problem is reformulated as a system of nonsmooth equations, a homotopy method is used to solve a sequence of smooth approximations to this system of equations. The global convergence properties of this approach are qualitatively different from those of other recent methods, which rely upon decrease of a merit function. This enables the algorithm to reliably solve certain classes of problems that prove troublesome for other methods. To improve efficiency, the homotopy algorithm is embedded in a generalized Newton method.

Key words. complementarity problems, homotopy methods, smoothing

AMS subject classifications. 90C33, 65H20, 65D10

PII. S1052623498337431

1. Introduction.

This paper discusses a robust method for solving mixed complementarity problems (MCPs), which is based upon the probability-one homotopy methods of [13, 31, 33]. The idea is to reformulate the MCP as a system of equations and then solve smooth approximations of this system with a homotopy method. While extremely robust, the homotopy methods we have considered tend to be slower than Newton-based methods. We therefore propose to embed the homotopy method inside a Newton-based method. A similar approach was successfully applied in the proximal perturbation strategy described in [4, 5, 7]. The idea is to invoke the homotopy technique only when the Newton-based method fails. The homotopy method is used to construct an improved starting point, from which the Newton method can be restarted.

The idea of applying homotopy methods to complementarity problems is not new; Watson [32] proposed such a method to solve the nonlinear complementarity problem (NCP). Watson's method involved reformulating the NCP as a system of smooth (C^2) equations and applying a homotopy method to solve this system. In the context of Newton-based methods, such smooth reformulations of complementarity problems are inferior to nonsmooth reformulations due to slow local convergence for degenerate solutions. In contrast, nonsmooth reformulations allow much faster (superlinear or quadratic) convergence to degenerate solutions. As such, we are interested in applying the homotopy method in the context of nonsmooth reformulations of the MCP. One such approach was developed by Sellami [25] and Sellami and Robinson [27, 26], based on the theoretical framework for piecewise smooth continuation methods presented in [1, 2, 3]. This approach was complicated by the fact that a special procedure was needed to make the transition from one smooth segment of the homotopy zero curve to another.

*Received by the editors April 15, 1998; accepted for publication (in revised form) May 21, 2001; published electronically January 9, 2002. This research was partially supported by NSF grant DMS–9973321.

http://www.siam.org/journals/siopt/12-3/33743.html

†Department of Mathematics, University of Colorado at Denver, Denver, CO 80217–3364 (sbillups@carbon.cudenver.edu).

In this paper, we consider a different approach; rather than applying the homotopy method to the original nonsmooth equations, we instead apply it to a smooth approximation of these equations. The solution of this smooth approximation can then be shown to be nearly a zero of the original function. This solution then gives the improved starting point from which to restart Newton's method. The overall strategy is as follows: First, apply a nonsmooth Newton method using a linesearch to ensure a reduction of a merit function at each iteration. If the Newton method stalls (for example, at a local minimum of the merit function), then apply the homotopy method to a smooth approximation of the equations. If the smooth approximation is properly chosen, the solution generated by the homotopy method will provide a reduction in the merit function of the original equations. It is then possible to return to the damped Newton method with no risk of returning to the region where the method stalled.

In the remainder of this paper, we describe this approach in more detail. Section 2 provides essential background material, including reformulations of MCPs, smoothing functions, and homotopy methods. Section 3 describes the algorithm in general and proves global convergence results. Section 4 discusses a particular implementation of the approach along with some numerical experimentation. Finally, section 5 gives conclusions.

2. Background.

2.1. Notation. The following notational conventions are used throughout the paper. Iteration numbers appear as superscripts on vectors and matrices and as subscripts on scalars. Subscripts on a vector (or matrix) are used to represent components or subvectors (or submatrices). For example, V_{ij} represents the component in the ith row and jth column of V, whereas $V_{i\cdot}$ and $V_{\cdot j}$ represent, respectively, the ith row and jth column of V. The Euclidean norm is represented by $\|\cdot\|$, whereas the ∞- and 1-norms are represented by $\|\cdot\|_\infty$ and $\|\cdot\|_1$, respectively.

The componentwise median function mid : $\mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ is defined by $\text{mid}_i(a, b, c) = \text{median}\{a_i, b_i, c_i\}$. The sign function sign is defined by

$$\text{sign}(x) = \begin{cases} 1, & x > 0, \\ 0, & x = 0, \\ -1, & x < 0. \end{cases}$$

The notation O is used as follows: Given a sequence $\{u^k\}$, we use the expression $\{O(u^k)\}$ to represent any sequence $\{v^k\}$ satisfying

$$\limsup_{k\to\infty} \frac{v^k}{u^k} < \infty.$$

2.2. MCPs. Given a rectangular region $\mathbb{B} = \prod_{i=1}^n [l_i, u_i]$ (where for each i, $-\infty \le l_i < u_i \le \infty$) and a function $F : \mathbb{B} \to \mathbb{R}^n$, the problem MCP$(F, \mathbb{B})$ is to find $x \in \mathbb{B}$ such that for each $i \in \{1, \ldots, n\}$ either

1. $x_i = l_i$ and $F_i(x) \ge 0$, or
2. $F_i(x) = 0$, or
3. $x_i = u_i$ and $F_i(x) \le 0$.

A more concise way of stating these conditions is that $\text{mid}(x - l, x - u, F(x)) = 0$, where mid is the componentwise median function.

In the above definition, if $l_i = 0, u_i = \infty$ for all $i = 1, 2, \ldots, n$, then MCP$(F, \mathbb{B})$ reduces to the standard form NCP(F), which is to find $x \geq 0$ such that

$$\min(x, F(x)) = 0.$$

In discussing algorithms for solving these problems, it is normal to assume that F is a C^1 function on an open set $\Omega \supset \mathbb{B}$. *For our homotopy approach, we shall make the stronger assumption that F is C^2 on Ω. Furthermore, for simplicity of discussion, we will assume that $\Omega = \mathbb{R}^n$.*

2.3. MCP reformulations. A common approach to solving the MCP is to define a function $H : \mathbb{R}^n \to \mathbb{R}^n$ such that the zeros of H correspond to solutions of the complementarity problem. To discuss such reformulations, we need to state several definitions, which are equivalent to the NCP function and the BVIP function defined in [23].

DEFINITION 2.1. *A function $\phi : \mathbb{R}^2 \to \mathbb{R}$ is called an* NCP function*, provided $\phi(a, b) = 0$, if and only if $\min(a, b) = 0$.*

DEFINITION 2.2. *A function $\psi : \mathbb{R}\bigcup\{-\infty\} \times \mathbb{R}\bigcup\{\infty\} \times \mathbb{R}^2 \to \mathbb{R}$ is called an* MCP function*, provided $\psi(l, u, a, b) = 0$, if and only if $\operatorname{mid}(a - l, a - u, b) = 0$.*

It is useful to further distinguish NCP and MCP functions according to their orientations.

DEFINITION 2.3. *An NCP function ϕ is called* positively oriented *if, for all $a, b \in \mathbb{R}$,*

$$\operatorname{sign}(\phi(a, b)) = \operatorname{sign}(\min(a, b)).$$

An MCP function ψ is called positively oriented *if, for all $l \in \mathbb{R}\bigcup\{-\infty\}, u \in \mathbb{R}\bigcup\{\infty\}$, and $a, b \in \mathbb{R}$,*

$$\operatorname{sign}(\psi(l, u, a, b)) = \operatorname{sign}(\operatorname{mid}(a - l, a - u, b)).$$

Finally, we can further classify NCP and MCP functions with the following definition.

DEFINITION 2.4. *A positively oriented NCP function ϕ is said to be* median-bounded *if there exist positive constants m and M such that, for all $a, b \in \mathbb{R}$,*

$$m|\min(a, b)| \leq |\phi(a, b)| \leq M|\min(a, b)|.$$

A positively oriented MCP function ψ is said to be median-bounded *if there exist positive constants m and M such that, for all $l \in \mathbb{R}\bigcup\{-\infty\}, u \in \mathbb{R}\bigcup\{\infty\}$, and $a, b \in \mathbb{R}$,*

$$m|\operatorname{mid}(a - l, a - u, b)| \leq |\psi(l, u, a, b)| \leq M|\operatorname{mid}(a - l, a - u, b)|.$$

Two popular NCP functions are the min function and the Fischer–Burmeister function [17, 18] defined by

$$\phi^{FB}(a, b) = a + b - \sqrt{a^2 + b^2}. \tag{2.1}$$

ϕ^{FB} is continuously differentiable everywhere except at the origin, and furthermore, it has the nice property that $(\phi^{FB})^2$ is continuously differentiable. (Note: This version of the Fischer–Burmeister function is actually the negative of the function presented in [17, 18]. This change of sign makes ϕ^{FB} a positively oriented NCP function.)

Billups [4, 5] showed how either the min function or the Fischer–Burmeister function can be used to construct an MCP function using the formula

$$\psi(l,u,a,b) := \phi(a-l, -\phi(u-a,-b)). \tag{2.2}$$

In the case in which ϕ is the min function, this formula simplifies to $\psi(l,u,a,b) = \operatorname{mid}(a-l, a-u, b)$. In the case in which ϕ is the Fischer–Burmeister function ϕ^{FB}, the resulting MCP function, which we denote by ψ^{FB}, is semismooth (see Definition 2.6 and [4, Proposition 3.2.7, Theorem 3.2.8]). This approach was generalized by Qi [23], who showed that if ϕ is any regular pseudo-smooth (see [23, Definition 2.1]) NCP function, then ψ defined by (2.2) is a regular pseudo-smooth MCP function.

PROPOSITION 2.5. *The functions ϕ^{FB}, defined by* (2.1)*, and ψ^{FB}, defined by* (2.2) *with ϕ replaced by ϕ^{FB}, are median-bounded NCP and MCP functions, respectively.*

Proof. By [29, Lemma 3.1], for any $\alpha, \beta \in \mathbb{R}$,

$$\hat{m}|\min(\alpha,\beta)| \le |\phi^{FB}(\alpha,\beta)| \le \hat{M}|\min(\alpha,\beta)|, \tag{2.3}$$

where $\hat{m} := 2-\sqrt{2}$ and $\hat{M} := 2+\sqrt{2}$. Thus, ϕ^{FB} is median-bounded. Let $c := -\phi^{FB}(u-a,-b)$. Then

$$\begin{aligned}
\hat{m}^2|\operatorname{mid}(a-l,a-u,b)| &= \hat{m}^2|\min(a-l,-\min(u-a,-b))| \le \hat{m}|\min(a-l,c)| \\
&\le |\phi^{FB}(a-l,c)| \qquad \big(= \psi^{FB}(l,u,a,b)\big) \\
&\le \hat{M}|\min(a-l,c)| \le \hat{M}^2|\min(a-l,-\min(u-a,-b))| \\
&= \hat{M}^2|\operatorname{mid}(a-l,a-u,b)|.
\end{aligned}$$

Thus, Definition 2.4 is satisfied for ψ^{FB} with $m = \hat{m}^2$ and $M = \hat{M}^2$. ∎

It follows from the definitions that if we define $H : \mathbb{R}^n \to \mathbb{R}^n$ by

$$H_i(x) := \begin{cases} \psi(l_i,u_i,x_i,F_i(x)) & \text{for } l_i, u_i \text{ both finite,} \\ \phi(x_i-l_i,F_i(x)) & \text{for } u_i=\infty, l_i \text{ finite,} \\ -\phi(u_i-x_i,-F_i(x)) & \text{for } l_i=-\infty, u_i \text{ finite,} \\ F_i(x) & \text{for } l_i=-\infty, u_i=\infty, \end{cases} \tag{2.4}$$

where ϕ and ψ are NCP and MCP functions, respectively, then a point x is a solution of $\mathrm{MCP}(F,\mathbb{B})$ if and only if $H(x)=0$. Thus, the problem of solving the MCP reduces to finding a zero of the function H. Given such a function H, it is usual to define the *natural* merit function

$$\theta(\cdot) := \frac{1}{2}\|H(\cdot)\|^2,$$

which is useful for linesearch strategies.

2.4. Generalized Newton algorithms. Since the function H defined by (2.4) is not necessarily smooth, Newton's method cannot be applied directly to solve $H(x)=0$; however, a generalization can be stated using the idea of the B-subdifferential.

By Rademacher's theorem, if $H : \mathbb{R}^n \to \mathbb{R}^m$ is locally Lipschitzian, it is differentiable almost everywhere. Let D_H be the set where H is differentiable. Define the B-subdifferential by

$$\partial_B H(x) := \left\{ V \;\middle|\; \exists\{x^k\} \to x, x^k \in D_H, \quad \text{with} \quad V = \lim_{k\to\infty} \nabla H(x^k) \right\}.$$

Step 1 [Initialization]. Select linesearch parameters $\alpha, \sigma \in (0,1)$, a positive integer m_{max}, a starting point $x^0 \in \mathbb{R}^n$, and a stopping tolerance *tol*. Set $k = 0$.

Step 2 [Direction generation]. Choose $V^k \in \partial_B H(x^k)$. If V^k is singular, stop, returning the point x^k along with a failure message. Otherwise choose the direction

$$d^k = -(V^k)^{-1} H(x^k). \tag{2.6}$$

Step 3 [Steplength determination]. Let m_k be the smallest nonnegative integer $m \leq m_{max}$ such that

$$\theta(x^k + \alpha^m d^k) - \theta(x^k) \leq -2\sigma\alpha^m \theta(x^k). \tag{2.7}$$

If no such m_k exists, stop, returning the point x^k along with a failure message. Otherwise set $x^{k+1} = x^k + \alpha^{m_k} d^k$.

Step 4 [Termination check]. If $\theta(x^{k+1}) < tol$, stop, returning the point x^{k+1}. Otherwise, return to Step 2, with k replaced by $k+1$.

FIG. 2.1. *Generalized damped Newton method.*

The Clarke subdifferential $\partial H(x)$ is the convex hull of $\partial_B H(x)$.

DEFINITION 2.6. *We say that H is* semismooth *at x if*

$$\lim_{\substack{V \in \partial H(x+th') \\ h' \to h,\, t \downarrow 0}} \{Vh'\}$$

exists for any $h \in \mathbb{R}^n$. We say that H is strongly semismooth *at x if for any sequence $\{d^k\} \subset \mathbb{R}^n$ converging to 0, and for $V^k \in \partial H(x + d^k)$,*

$$V^k d^k - H'(x; d^k) = O\left(\left\|d^k\right\|^2\right). \tag{2.5}$$

DEFINITION 2.7. *We say that a semismooth function H is* BD-regular *at x if all elements in $\partial_B H(x)$ are nonsingular.*

DEFINITION 2.8. *Suppose that $H : \mathbb{R}^n \to \mathbb{R}^m$ is B-differentiable in a neighborhood of x. We say that the directional derivative $H'(\cdot;\cdot)$ is* semicontinuous *at x if, for every $\epsilon > 0$, there exists a neighborhood N of x such that, for all $x + h \in N$,*

$$\|H'(x+h;h) - H'(x;h)\| \leq \epsilon \|h\|.$$

We say that $H'(\cdot;\cdot)$ is semicontinuous of degree 2 *at x if there exist a constant L and a neighborhood N of x such that, for all $x + h \in N$,*

$$\|H'(x+h;h) - H'(x;h)\| \leq L \left\|h^2\right\|.$$

A nonsmooth version of a damped Newton method, which is discussed in [5], is shown in Figure 2.1.

The algorithm has three features that make it attractive for use in our context:

1. The calculation of the search direction at each iteration only requires solving a single linear equation (namely, (2.6)), instead of a more complicated subproblem, such as a linear complementarity problem or quadratic program.
2. The algorithm either fails in a finite number of steps or produces a sequence of iterates $\{x^k\}$ such that the corresponding merit function values $\{\theta(x^k)\}$ are strictly decreasing and converge to zero. This property, which is an obvious consequence of the upper bound m_{max} placed on m_k for the steplength determination step, guarantees finite termination if $tol > 0$ and is essential for our purposes. When the algorithm fails, we intend to employ a homotopy method to construct an improved starting point $\tilde{x}$ for which $\theta(\tilde{x})$ is smaller than any merit function values evaluated thus far. It will then be possible to restart the Newton method from $\tilde{x}$ with the guarantee that the iterates will not return to the region where the algorithm failed previously.
3. The algorithm has fast local convergence behavior near a solution, which is summarized in the following theorem from Qi [22].

THEOREM 2.9. *Suppose that x^* is a solution of $H(x) = 0$, and that H is semismooth and BD-regular at x^*. Then the iteration method defined by $x^{k+1} = x^k + d^k$, where d^k is given by (2.6), is well defined and convergent to x^* superlinearly in a neighborhood of x^*. In addition, if $H(x^k) \neq 0$ for all k, then*

$$\lim_{k\to\infty} \frac{\|H(x^{k+1})\|}{\|H(x^k)\|} = 0.$$

If, in addition, H is directionally differentiable at a neighborhood of x^* and $H'(\cdot;\cdot)$ is semicontinuous of degree 2 at x^*, then the convergence of the iteration method is quadratic.

One consequence of this local convergence theorem is that within a neighborhood of a BD-regular solution x^*, the linesearch criteria (2.7) will be satisfied by $m_k = 0$. Thus, the inner algorithm will take full Newton steps and achieve the fast local convergence rates specified by the theorem.

2.5. Homotopy methods. The probability-one homotopy methods we consider in this paper are based on the following proposition from [13, 30, 31].

PROPOSITION 2.10. *Let $H : \mathbb{R}^n \to \mathbb{R}^n$ be a C^2 function and suppose there exists a C^2 map*

$$\rho : \mathbb{R}^m \times [0, 1) \times \mathbb{R}^n \to \mathbb{R}^n$$

such that

1. *the $n \times (m + 1 + n)$ Jacobian matrix $\nabla\rho(a, \lambda, x)$ has rank n on the set*

$$\rho^{-1}(0) = \{(a, \lambda, x) \mid a \in \mathbb{R}^m, 0 \leq \lambda < 1, x \in \mathbb{R}^n, \rho(a, \lambda, x) = 0\}$$

and, for any fixed $a \in \mathbb{R}^m$ defining $\rho_a(\lambda, x) := \rho(a, \lambda, x)$, the following also hold:

2. *$\rho_a(0, x) = 0$ has a unique solution x_0,*
3. *$\rho_a(1, x) = H(x)$,*
4. *$\rho_a^{-1}(0)$ is bounded.*

Then for almost all $a \in \mathbb{R}^m$ (in the sense of Lebesgue measure) there exists a zero curve γ of ρ_a, along which the Jacobian matrix $\nabla\rho_a$ has rank n, emanating from $(0, x_0)$ and reaching a zero $\bar{x}$ of H at $\lambda = 1$. Moreover, γ does not intersect itself and is disjoint from any other zeros of ρ_a.

The expression "reaching a zero" requires some clarification. This expression means that there exists a sequence of points $\{(\lambda_k, x^k)\}$ in γ accumulating at $(1, \bar{x})$.

The full rank conclusion of $\nabla\rho_a$ on $\rho^{-1}(0)$ allows us to parameterize γ by arc length. Thus, we denote by $\gamma(s)$ the point on γ of arclength s along γ from $(0, x_0)$.

Given such a homotopy mapping ρ, a globally convergent algorithm can be constructed, which picks $a \in \mathbb{R}^m$ (uniquely determining x_0) and then tracks the homotopy zero curve γ. Perhaps the simplest choice of homotopy mapping is given by $\rho : \mathbb{R}^n \times [0,1) \times \mathbb{R}^n \to \mathbb{R}^n$ defined by

$$\rho(a, \lambda, x) := \lambda H(x) + (1-\lambda)(x-a). \tag{2.8}$$

When H is a C^2 map, this choice of ρ satisfies properties 1–3 but not necessarily 4. However, there are fairly general sufficient conditions on $H(x)$ guaranteeing that it does satisfy property 4. One such sufficient condition is particularly relevant in our context and gives us the following theorem from [32].

THEOREM 2.11. *Let $H : \mathbb{R}^n \to \mathbb{R}^n$ be a C^2 map such that*

$$\text{for some } r > 0, \quad x^T H(x) \geq 0 \ \text{ whenever } \ \|x\| = r. \tag{2.9}$$

Then H has a zero in the ball $\{x \in \mathbb{R}^n \mid \|x\| \leq r\}$, and for almost all a in the interior of this ball there is a zero curve γ of

$$\rho_a(\lambda, x) := \lambda H(x) + (1-\lambda)(x-a),$$

along which the Jacobian matrix $\nabla\rho_a(\lambda, x)$ has full rank, emanating from $(0, a)$ and reaching a zero $\bar{x}$ of H at $\lambda = 1$. Furthermore, γ has finite arc length if $\nabla H(\bar{x})$ is nonsingular.

Conceptually, the homotopy method is very simple: Construct the homotopy mapping ρ_a and follow the zero curve γ from the point x^0 to the solution. However, implementing this idea in an efficient computer algorithm is very difficult. Clearly, it is impractical to trace the zero curve exactly. Instead we must generate a sequence of points $\{\lambda^k, x^k\}$ that loosely follow the zero curve (within some prescribed tolerances) and that make reliable progress along its arclength. These points should not be too close together, since this requires more function evaluations than are really necessary. However, if these points are spaced too loosely, one can end up tracing a different component of the zero set, or reversing direction on the zero curve γ, thereby never reaching the desired solution.

Obviously it is not possible to ensure "perfect" curve tracking; however, much research has been devoted to this problem and reliable codes have been developed. One such code, which we use in our implementation, is HOMPACK [33].

2.6. Smoothing functions. Since the function H defined in (2.4) is not C^2, we cannot apply a homotopy algorithm to it directly. Instead we must form a smooth approximation of H. In recent years, numerous techniques have emerged for solving the nonsmooth equation $H(x) = 0$ which are based on the notion of smoothing (see, for example, [8] and the references therein).

The basic idea of these techniques is to approximate the function H by a family of smooth approximations H_μ with *smoothing parameter* μ. Under suitable assumptions, the solutions to the perturbed systems $H_\mu = 0$ form a smooth trajectory, leading to a solution of the original problem. The smoothing methods generate a sequence of iterates that follow this trajectory. However, these methods decrease μ monotonically, and so they may fail for highly nonlinear functions.

DEFINITION 2.12. *Given a nonsmooth function $\varphi : \mathbb{R}^p \to \mathbb{R}$, a* smoother for φ *is a continuous function $\tilde{\varphi} : \mathbb{R}^p \times \mathbb{R}_+ \to \mathbb{R}$ with the following properties:*

1. $\tilde{\varphi}(x, 0) = \varphi(x)$;
2. $\tilde{\varphi}(\cdot, \mu)$ *is continuously differentiable for all* $\mu > 0$.

If $\tilde{\varphi}(\cdot, \mu)$ is twice continuously differentiable for all $\mu > 0$, then $\tilde{\varphi}$ is said to be a C^2 smoother *for* φ.

We shall find it convenient to make the following assumption on the smoother.

Assumption 2.13. There exists a function $\xi : \mathbb{R}_+ \to \mathbb{R}_+$ with $\lim_{\mu \downarrow 0} \xi(\mu) = 0$ such that

$$|\tilde{\varphi}(x, \mu) - \tilde{\varphi}(x, 0)| \leq \xi(\mu)$$

for all $x \in \mathbb{R}^p$ and $\mu \in \mathbb{R}_+$.

Numerous smoothers have been proposed in the literature [9, 10, 11, 12, 19, 20, 24, 28, 34]. Many of the early smoothers were unified by the family of smoothing functions described by Chen and Mangasarian [11]. More recently, Gabriel and Moré [19] introduced a more general family of smoothers for the MCP, which includes the Chen–Mangasarian family. Unfortunately, many of these smoothers are not C^2 and so are not appropriate for our homotopy framework.

A smoother that is C^2 is Kanzow's smoother for the Fischer–Burmeister function (2.1) [20]:

$$\tilde{\phi}^K(a, b, \mu) := a + b - \sqrt{a^2 + b^2 + 2\mu}. \tag{2.10}$$

Using $\tilde{\phi}^K$ in the formula (2.2) yields the following smoother for ψ^{FB}:

$$\tilde{\psi}^{FB}(l, u, a, b, \mu) := \tilde{\phi}^K(a - l, -\tilde{\phi}^K(u - a, -b, \mu), \mu). \tag{2.11}$$

The following proposition establishes that Assumption 2.13 is satisfied for this smoother.

PROPOSITION 2.14. *The smoother $\tilde{\psi}^{FB}$ defined by* (2.11) *satisfies*

$$\left|\tilde{\psi}^{FB}(l, u, a, b, \mu) - \psi^{FB}(l, u, a, b)\right| \leq 3\sqrt{2\mu}.$$

Proof. For simplicity, define $\phi_\mu := \tilde{\phi}^K(\cdot, \cdot, \mu)$. It is easy to show that for all $a, b, c \in \mathbb{R}$, $|\phi_\mu(a, b) - \phi_\mu(a, c)| \leq 2|b - c|$ for all $\mu \in \mathbb{R}_+$. It is also easy to show that $|\phi_\mu(a, b) - \phi(a, b)| \leq \sqrt{2\mu}$. Thus,

$$\begin{aligned}
&\left|\tilde{\psi}^{FB}(l, u, a, b, \mu) - \psi^{FB}(l, u, a, b)\right| \\
&\quad = |\phi_\mu(a - l, -\phi_\mu(u - a, -b)) - \phi(a - l, -\phi(u - a, -b))| \\
&\quad = |\phi_\mu(a - l, -\phi_\mu(u - a, -b)) - \phi_\mu(a - l, -\phi(u - a, -b)) \\
&\qquad + \phi_\mu(a - l, -\phi(u - a, -b)) - \phi(a - l, -\phi(u - a, -b))| \\
&\quad \leq 2\,|-\phi_\mu(u - a, -b) + \phi(u - a, -b)| + \sqrt{2\mu} \\
&\quad \leq 2\sqrt{2\mu} + \sqrt{2\mu} = 3\sqrt{2\mu}. \qquad \square
\end{aligned}$$

3. Algorithmic framework. The basic idea behind our algorithm is to employ the damped Newton method from Figure 2.1 until it fails. Such failure may, for example, be a result of the iterates converging to a local minimum of the merit

Step 1 [Initialization]. Given a starting vector $x^0 \in \mathbb{R}^n$, a parameter $0 < \beta < 1$, and a convergence tolerance $\epsilon > 0$, choose $0 < \beta < 1$, and set $k = 0$.

Step 2 [Attempt descent algorithm]. Run the generalized damped Newton algorithm from Figure 2.1 with starting point x^k and with $tol = \epsilon$. This generates a point $\tilde{x}^k$.

Step 3 [Termination check]. If $\theta(\tilde{x}^k) := \frac{1}{2}\left\|H(\tilde{x}^k)\right\|^2 < \epsilon$, stop, returning the solution $\tilde{x} := \tilde{x}^k$; otherwise continue with Step 4.

Step 4 [Generate better starting point]. Determine a smoothing parameter $\mu > 0$ such that $\xi(\mu) \le (\beta/2\sqrt{n})\left\|H(\tilde{x}^k)\right\|$. Run the homotopy algorithm to solve the smooth equation $H_\mu(x) = 0$ to a tolerance of $\frac{\beta}{2}\left\|H(\tilde{x}^k)\right\|$. If the homotopy algorithm fails, stop. Otherwise, set x^{k+1} equal to the solution.

Step 5. Return to Step 2 with k replaced by $k + 1$.

FIG. 3.1. *Algorithmic framework.*

function θ. When the Newton method fails, we then apply a homotopy method to solve a smooth approximation $H_\mu(x) = 0$, where

$$(H_\mu)_i(x) := \begin{cases} \tilde{\psi}(l_i, u_i, x_i, F_i(x), \mu) & \text{for } l_i, u_i \text{ both finite,} \\ \tilde{\phi}(x_i - l_i, F_i(x), \mu) & \text{for } u_i = \infty, l_i \text{ finite,} \\ -\tilde{\phi}(u_i - x_i, -F_i(x), \mu) & \text{for } l_i = -\infty, u_i \text{ finite,} \\ F_i(x) & \text{for } l_i = -\infty, u_i = \infty. \end{cases} \tag{3.1}$$

It is not necessary to solve this smooth equation exactly; we are only interested in generating a point $\tilde{x}$ for which θ is decreased. Under mild assumptions, the homotopy method will find such a point, provided that (1) the smoothing parameter is not too large and (2) the stopping tolerance for the homotopy method is sufficiently small. The general algorithm is given in Figure 3.1.

The global convergence behavior for this algorithm is established by the following proposition.

PROPOSITION 3.1. *Let ψ be an MCP function, let $\tilde{\psi}$ be a C^2-smoother for ψ satisfying Assumption* 2.13*, and let H be defined by* (2.4)*. Choose $\mu > 0$, and let H_μ be defined by* (3.1)*. The algorithm in Figure* 3.1 *either terminates in Step* 3 *(at an approximate solution $\tilde{x}$ satisfying $\theta(\tilde{x}) < \epsilon$) or fails in Step* 4 *(during the homotopy algorithm).*

Proof. Assume that Step 4 of the algorithm is always successful and that the test in Step 3 of the algorithm always fails. Then since the damped Newton method always terminates in a finite number of iterations, the algorithm will generate an infinite sequence of points $\{x^k\}$. Because of the linesearch criteria in the damped Newton method, $\theta(\tilde{x}^k) \le \theta(x^k)$. Now,

$$\begin{aligned} \left\|H(x^{k+1})\right\| &\le \left\|H_\mu(x^{k+1})\right\| + \left\|H(x^{k+1}) - H_\mu(x^{k+1})\right\| \\ &\le \tfrac{\beta}{2}\left\|H(\tilde{x}^k)\right\| + \sqrt{n}\xi(\mu) \\ &\le \tfrac{\beta}{2}\left\|H(\tilde{x}^k)\right\| + \tfrac{\beta}{2}\left\|H(\tilde{x}^k)\right\| \\ &\le \beta\left\|H(\tilde{x}^k)\right\|. \end{aligned}$$

Thus, $\theta(x^{k+1}) \leq \beta^2\theta(\tilde{x}^k) \leq \beta^2\theta(x^k) \leq \beta^{2(k+1)}\theta(x^0)$. Thus, since $\beta < 1$, then for some finite value of k, $\theta(x^k) < \epsilon$, contradicting the assumption that the test in Step 3 always fails. □

It should be noted that the stopping criterion $\theta(\tilde{x}) < \epsilon$ does not ensure that $\tilde{x}$ is near a solution to the MCP, no matter how small ϵ is. But the proposition does ensure that if we set $\epsilon = 0$ and assume that the homotopy algorithm in Step 4 never fails, then any accumulation point of the iterates $\{\tilde{x}^k\}$ will be a solution. Thus, for example, if the level sets of θ are bounded, then $\tilde{x}$ can be made arbitrarily close to a solution by choosing ϵ sufficiently small. In this case, the success of the algorithm relies entirely upon the success of the homotopy method in Step 4. This in turn depends on two questions: (1) Does the homotopy zero curve lead to a solution (or at least an approximate zero of H_μ)? and (2) Can the homotopy method successfully track this zero curve? Since we cannot guarantee successful curve tracking, the second question represents a theoretical stumbling block. However, as previously discussed, sophisticated codes, such as HOMPACK [33], are available that perform this curve tracking fairly reliably. We therefore focus our attention on the first question.

Theorem 2.11 provides sufficient conditions under which a homotopy zero curve exists that leads to a solution in finite length. We now prove several results that are more specific to the complementarity framework.

LEMMA 3.2. *Let ψ be a positively oriented median-bounded MCP function, and let H be defined by* (2.4). *If $\mathbb{B}$ is bounded, then*

$$\lim_{\|x\|\to\infty} \frac{x^T H(x)}{\|x\|} = +\infty. \tag{3.2}$$

Proof. For a given $x \in \mathbb{R}^n$, suppose $x_i < 0$ and $H_i(x) > 0$. Then by positive orientation, $\operatorname{mid}(x_i - l_i, x_i - u_i, F_i(x)) > 0$, which implies that $x_i > l_i$ and $\operatorname{mid}(x_i - l_i, x_i - u_i, F_i(x)) \leq x_i - l_i < |l_i|$. Thus,

$$\begin{aligned} x_i H_i(x) &\geq -|l_i|\,|\psi(l_i, u_i, x_i, F_i(x))| \\ &\geq -|l_i|M|\operatorname{mid}(x_i - l_i, x_i - u_i, F_i(x))| \\ &\geq -Ml_i^2, \end{aligned}$$

where M is the constant guaranteed by the median-bounded property (see Definition 2.4). Similarly, if $x_i > 0$ and $H_i(x) < 0$, we can show that $x_i H_i(x) \geq -Mu_i^2$. Since these are the only two cases in which $x_i H_i(x)$ can be negative, we have that

$$x_i H_i(x) \geq -Mb_i^2, \tag{3.3}$$

where $b_i := \max\{|l_i|, |u_i|, 1\}$. Let $b_{max} := \max_i b_i$, $b_{min} := \min_i b_i$, and $d := \|b\|$.

For a given x, let $\kappa_x := \|x\|/d$. Then if $\kappa_x > 1$, there exists an index j such that $|x_j| \geq \kappa_x b_j$. If x_j is positive, then $\operatorname{mid}(x_j - l_j, x_j - u_j, F_j(x)) \geq (\kappa_x - 1)b_j$, so by median-boundedness,

$$x_j H_j(x) = x_j\psi(l_j, u_j, x_j, F_j(x)) \geq (\kappa_x b_j)(m(\kappa_x - 1)b_j) > \kappa_x(\kappa_x - 1)mb_{min}^2. \tag{3.4}$$

In similar fashion, we can show that this inequality holds if x_j is negative. Thus,

$$\begin{aligned} x^T H(x) &= \sum_{i\neq j} x_i H_i(x) + x_j H_j(x) \\ &> -N_1 + \kappa_x(\kappa_x - 1)mb_{min}^2 \qquad \text{(by (3.3) and (3.4))}, \end{aligned}$$

where $N_1 := nMb_{max}^2$. It follows that

$$\begin{aligned}\lim_{\|x\|\to\infty} \frac{x^T H(x)}{\|x\|} &> \lim_{\|x\|\to\infty} -\frac{N_1}{\|x\|} + \frac{\kappa_x(\kappa_x - 1)mb_{min}^2}{\|x\|} \\ &= \lim_{\kappa_x\to\infty} \frac{(\kappa_x - 1)mb_{min}^2}{d} \\ &= +\infty. \qquad \square\end{aligned}$$

THEOREM 3.3. *Let ψ be a positively oriented median-bounded MCP function, and let $\tilde{\psi}$ be a smoother for ψ satisfying Assumption* 2.13. *Choose $\mu > 0$, and let H_μ be defined by* (3.1), *where $\mathbb{B} = \prod_{i=1}^n [l_i, u_i]$ is bounded. Then, H_μ satisfies condition* (2.9) *for all r sufficiently large, and therefore the conclusions of Theorem* 2.11 *hold.*

Proof.

$$\begin{aligned}x^T H_\mu(x) &= x^T H(x) + x^T(H_\mu(x) - H(x)) \\ &\geq x^T H(x) - \|x\| \, \|H_\mu(x) - H(x)\| \\ &\geq x^T H(x) - \|x\| \, \sqrt{n}\xi(\mu).\end{aligned}$$

Dividing both sides by $\|x\|$ and taking the limit as $\|x\| \to \infty$, we have

$$\begin{aligned}\lim_{\|x\|\to\infty} \frac{x^T H_\mu(x)}{\|x\|} &\geq \lim_{\|x\|\to\infty} \frac{x^T H(x)}{\|x\|} - \sqrt{n}\xi(\mu) \\ &= +\infty \quad \text{by Lemma 3.2.}\end{aligned}$$

Thus, for r sufficiently large, we have $x^T H_\mu(x) > 0$ whenever $\|x\| = r$, so (2.9) holds. $\square$

It is not at all difficult to find MCP functions and corresponding smoothers that satisfy the assumptions of this theorem. With these in hand, the theorem gives a strong result: *If $\mathbb{B}$ is bounded, then the homotopy zero curve being tracked in Step* 4 *of Figure* 3.1 *leads to a solution $\bar{x}$ of $H_\mu(x) = 0$. Furthermore, if $\nabla H_\mu(\bar{x})$ is nonsingular, then this zero curve has finite arclength. Thus, the algorithm will not fail in Step* 4 *as long as the curve tracking is performed reliably.*

In the case in which $\mathbb{B}$ is unbounded, the analysis is a bit more difficult. To appreciate the difficulties, note that even if F is a Lipschitz continuous, strongly monotone function, there is no guarantee that H_μ will satisfy condition (2.9). Fortunately, this is not necessary. To guarantee the boundedness of the zero curve, it is sufficient that F satisfy the following assumption.

Assumption 3.4.

$$\lim_{\|x\|\to\infty} \max_k \min\{|x_k|, |F_k(x)|, x_k F_k(x)\} = \infty.$$

It is easy to show (see, for example, the proof of [29, Theorem 2.3]) that if F is strongly monotone and Lipschitz continuous, it will satisfy Assumption 3.4.

THEOREM 3.5. *Suppose that $F : \mathbb{R}^n \to \mathbb{R}^n$ satisfies Assumption* 3.4. *Let ψ be a positively oriented median-bounded MCP function, let $\tilde{\psi}$ be a C^2 smoother for ψ satisfying Assumption* 2.13, *and let H_μ be defined by* (3.1). *Choose $a \in \mathbb{R}^n$ and define $\rho_a : [0, 1) \times \mathbb{R}^n \to \mathbb{R}^n$ by*

$$\rho_a(\lambda, x) := \lambda H_\mu(x) + (1 - \lambda)(x - a).$$

Then the zero set $\rho^{-1}(0)$ is bounded, and therefore the conclusions of Proposition 2.10 *hold.*

Proof. Choose $\delta > \max\{\|a\|_\infty, \max_{l_i > -\infty} |l_i|, \max_{u_i < \infty} |u_i|\} + \xi(\mu)/m$, where ξ is as defined in Assumption 2.13 and m is as defined in Definition 2.4. By Assumption 3.4, there exists some $\bar{r} \geq 0$ such that $\|x\| \geq \bar{r}$ implies there is some index k such that

$$|x_k| \geq \delta, \quad |F_k(x)| \geq \delta, \quad \text{and} \quad x_k F_k(x) > 0. \tag{3.5}$$

Let $(\lambda, x) \in \rho_a^{-1}(0)$ and suppose, towards a contradiction, that $\|x\| \geq \bar{r}$. Let k be the index that satisfies (3.5). Define $R_k := \text{mid}(x_k - l_k, x_k - u_k, F_k(x))$, $S_k := H_k(x)$, and $T_k := (H_\mu)_k(x)$.

By the choice of δ, if l_k is finite, then $|x_k| > |l_k|$, so $x_k - l_k$ will have the same sign as x_k, and $|x_k - l_k| \geq \delta - |l_k| > \xi(\mu)/m$. Similarly, if u_k is finite, then $x_k - u_k$ will have the same sign as x_k, and $|x_k - u_k| > \xi(\mu)/m$. Finally, by (3.5), $F_k(x)$ has the same sign as x_k, and $|F_k(x)| > \xi(\mu)/m$. Since $\text{mid}(x_k - l_k, x_k - u_k, F_k(x))$ is either $F_k(x), x_k - l_k$ (if l_k is finite) or $x_k - u_k$ (if u_k is finite), it follows that R_k has the same sign as x_k and $|R_k| > \xi(\mu)/m$.

Now, by positive orientation and median-boundedness, S_k has the same sign as R_k (therefore the same sign as x_k), and $|S_k| \geq m|R_k| > \xi(\mu)$. Finally, by Assumption 2.13, $|T_k - S_k| \leq \xi(\mu)$, so T_k must have the same sign as S_k, and therefore the same sign as x_k.

On the other hand, since $\rho_a(\lambda, x) = 0$,

$$(\rho_a(\lambda, x))_k = \lambda T_k + (1 - \lambda)(x_k - a_k) = 0.$$

Since $\delta > a_k$, $x_k - a_k$ has the same sign as x_k. Thus, since $\lambda \in [0, 1)$, T_k must have the opposite sign of x_k, which is a contradiction. Therefore, $\|x\|$ must be less than $\bar{r}$. ∎

A deficiency of the above result is that Assumption 3.4 is not necessarily satisfied for monotone functions. We therefore present some additional results based on the following assumption.

Assumption 3.6 (global monotonicity). There exists $r > 0$ such that for any $x, y \in \mathbb{R}^n$ with $\|x - y\| \geq r$

$$(x - y)^T (F(x) - F(y)) \geq 0.$$

Observe that this assumption is satisfied trivially if F is monotone. However, the assumption is weaker than monotonicity, since, for example, $\|F\|$ may have many local minima.

We shall also assume the existence of a strictly feasible point for the MCP, which is defined as follows.

DEFINITION 3.7. *A point $\bar{x} \in \mathbb{R}^n$ is said to be a* strictly feasible point *for the MCP($F, \mathbb{B}$) if, for $i = 1, \ldots, n$,*

1. $l_i = -\infty, u_i = \infty \Longrightarrow F_i(\bar{x}) = 0$,
2. $l_i = -\infty, u_i < \infty \Longrightarrow F_i(\bar{x}) < 0$,
3. $l_i > -\infty, u_i = \infty \Longrightarrow F_i(\bar{x}) > 0$.

Finally, we shall need some stronger assumptions on the smoothers for the NCP and MCP functions.

Assumption 3.8. For all $\mu \geq 0$, $l, u \in \mathbb{R}$ there exists $c > 0$ such that, for all $a, b \in \mathbb{R}$,

1. $\lim_{a_k\to\infty,b_k\to b}\tilde{\phi}(a_k,b_k,\mu)\geq cb$,
2. $\lim_{a_k\to a,b_k\to\infty}\tilde{\phi}(a_k,b_k,\mu)\geq ca$,
3. $\lim_{a_k\to a,b_k\to\infty}\tilde{\psi}(l,u,a_k,b_k,\mu)\geq c(a-l)$,
4. $\lim_{a_k\to a,b_k\to-\infty}\tilde{\psi}(l,u,a_k,b_k,\mu)\leq c(a-u)$.

PROPOSITION 3.9. *The smoothers $\tilde{\phi}^K$ and $\tilde{\psi}^{FB}$ for ϕ^{FB} and ψ^{FB}, respectively, satisfy Assumption* 3.8.

Proof. The following equations can easily be shown:

$$\lim_{a_k\to\infty,b_k\to b}\tilde{\phi}^K(a_k,b_k,\mu)=b, \tag{3.6}$$

$$\lim_{a_k\to a,b_k\to\infty}\tilde{\phi}^K(a_k,b_k,\mu)=a, \tag{3.7}$$

$$\lim_{a_k\to-\infty}\tilde{\phi}^K(a_k,b_k,\mu)=\lim_{b_k\to-\infty}\tilde{\phi}^K(a_k,b_k,\mu)=-\infty. \tag{3.8}$$

Given sequences $\{a_k\}$ and $\{b_k\}$, define $d_k:=-\phi^K(u-a_k,-b_k,\mu)$. Then, using (3.8) and (3.7), respectively,

$$\begin{aligned}\lim_{a_k\to a,b_k\to\infty}\tilde{\psi}^{FB}(l,u,a_k,b_k,\mu)&=\lim_{a_k\to a,b_k\to\infty}\tilde{\phi}^K(a_k-l,-\phi^K(u-a,-b,\mu),\mu)\\&=\lim_{a_k\to a,d_k\to\infty}\tilde{\phi}^K(a_k-l,d_k,\mu)=a-l.\end{aligned}$$

Similarly, using (3.7), [29, Lemma 3.1], and the fact that $\tilde{\phi}^K(a,b,\mu)\leq\phi^{FB}(a,b)$,

$$\begin{aligned}\lim_{a_k\to a,b_k\to-\infty}\tilde{\psi}^{FB}(l,u,a_k,b_k,\mu)&=\lim_{a_k\to a,d_k\to a-u}\tilde{\phi}^K(a_k-l,d_k,\mu)\\&=\tilde{\phi}^K(a-l,a-u,\mu)\\&\leq\phi^{FB}(a-l,a-u)\leq(2-\sqrt{2})(a-u).\end{aligned}$$ ∎

LEMMA 3.10. *Suppose that $F:\mathbb{R}^n\to\mathbb{R}^n$ satisfies Assumption* 3.6 *and there is a strictly feasible point $\bar{x}$ for MCP(F,*$\mathbb{B}$*). Let ϕ and ψ be median-bounded NCP and MCP functions, respectively, and let $\tilde{\phi}$ and $\tilde{\psi}$ be smoothers for ϕ and ψ, respectively, satisfying Assumptions* 2.13 *and* 3.8. *Let H_μ be defined by* (3.1)*, choose $a\in\operatorname{int}\mathbb{B}$, and define $\rho_a:[0,1)\times\mathbb{R}^n\to\mathbb{R}^n$ by*

$$\rho_a(\lambda,x):=\lambda H_\mu(x)+(1-\lambda)(x-a). \tag{3.9}$$

Then for any unbounded sequence of points in $\rho_a^{-1}(0)$, there is an unbounded subsequence $\{(\lambda_k,x^k)\}\subset\rho_a^{-1}(0)$ such that $\lambda_k\to1$, and $H_\mu(x^k)\to0$.

Proof. Let $\{(\lambda_k,x^k)\}$ be an unbounded sequence of points in $\rho_a^{-1}(0)$. After going to a subsequence, we may assume that $x^k\to x^*$, $\lambda_k\to\lambda_*$, $F(x^k)\to f^*$, and $H_\mu(x^k)\to h^*$ for some $x^*,f^*,h^*\in(\mathbb{R}\bigcup\{-\infty,\infty\})^n$ with $\|x^*\|=\infty$ and $\lambda_*\in[0,1]$.

Define $P:=\{i\mid x_i^*=\infty\}$, $N:=\{i\mid x_i^*=-\infty\}$, and $B:=\{i\mid x_i^*\text{ finite}\}$. Suppose u_i is finite for some $i\in P$; then $\operatorname{mid}(x_i^*-l_i,x_i^*-u_i,F_i(x^*))\geq x_i^*-u_i=\infty$. By median-boundedness and Assumption 2.13, $h_i^*=\infty$. But this yields a contradiction, since by (3.9), $(\rho_a(\lambda_k,x^k))_i$ would be positive for all sufficiently large k. Thus $u_i=\infty$ for all $i\in P$. A similar argument establishes that $l_j=-\infty$ for all $j\in N$.

Now, suppose $\lambda_*<1$. Then for any $i\in P$, by (3.9), $h_i^*=-\infty$. It follows that $f_i^*=-\infty$. Similarly, for any $j\in N$, $f_j^*=\infty$. Thus,

$$(x_i^*-a_i)^T(f_i^*-F_i(a))=-\infty\text{ for all }i\in P\bigcup N. \tag{3.10}$$

Since $P \bigcup N \neq \emptyset$, then by global monotonicity there exists j such that $\limsup(x_j^k - a_j)(F_j(x^k) - F_j(a)) = \infty$. By (3.10), $j \in B$, so $|f_j^*| = \infty$. Without loss of generality (going to a subsequence if necessary), we may assume $f_j^* = \infty$ and $x_j^k > a_j$ for all k. By (3.9), $\lambda_k > 0$ and $(H_\mu(x^k))_j < 0$ for all k, so $h_j^* \leq 0$. But this yields a contradiction since, by Assumption 3.8, $h_j^* \geq c(x_i^* - l_i) \geq c(a_i - l_i) > 0$. It follows that $\lambda_* = 1$. By (3.9), $h_i^* = 0$ for all $i \in B$, $h_i^* \leq 0$ for all $i \in P$, and $h_i^* \geq 0$ for all $i \in N$.

Now, suppose $h_i^* < 0$ for some $i \in P$. Since $u_i = \infty$, by Assumption 3.8, $f_i^* < 0$, and since $\bar{x}$ is strictly feasible, $F_i(\bar{x}) \geq 0$. Thus, $(x_i^* - \bar{x}_i)(f_i^* - F_i(\bar{x})) = -\infty$. By global monotonicity (Assumption 3.6), there exists j such that, going to a subsequence if necessary, $\lim(x_j^k - \bar{x}_j)(F_j(x^k) - F_j(\bar{x})) = \infty$. Without loss of generality (going to a subsequence if necessary), we may assume that $x_j^k > \bar{x}_j$ and $F_j(x^k) - F_j(\bar{x}) > 0$ for all k. Thus, either $x_j^* = \infty$ or $f_j^* = \infty$. Then by Assumption 3.8, $h_j^* \geq c(x_j^* - l_j) \geq c(\bar{x}_j - l_j) > 0$. Since $h_j^* = 0$ for all $j \in B$, it follows that $j \in P$ and $u_j = \infty$. If $l_j > -\infty$, then by Assumption 3.8, $h^* \geq cf_j^* \geq cF_j(\bar{x}) > 0$ (by strict feasibility), contradicting the fact that $h_j^* \leq 0$ for all $j \in P$. If instead $l_j = -\infty$, then $(H_\mu(x^k))_j = F_j(x^k) > F_j(\bar{x}) = 0$ for all k. But $(H_\mu(x^k))_j$ would then have the same sign as $(x_j^k - a_j)$ for k sufficiently large. This yields a contradiction since, by (3.9), $(\rho_a(\lambda_k, x^k))_i$ would be positive for all k sufficiently large.

It follows that $h_i^* = 0$ for all $i \in P$. A similar argument yields $h_i^* = 0$ for all $i \in N$. Thus, $h^* = 0$. □

THEOREM 3.11. *Suppose that $F : \mathbb{R}^n \to \mathbb{R}^n$ satisfies Assumption* 3.6 *and there is a strictly feasible point $\bar{x}$ for MCP(F, $\mathbb{B}$). Let H_μ and ρ_a be as defined in Lemma* 3.10*, let γ_a be the connected component of $\rho_a^{-1}(0)$ containing $(0, a)$, and let $(\lambda(s), x(s))$ represent the point on γ_a of arclength s along γ_a from $(0, a)$. Then, for almost all a in the interior of $\mathbb{B}$, either γ_a reaches a zero $\bar{x}$ of H_μ in finite arclength or $\lim_{s\to\infty} \lambda(s) = 1$ and $\lim_{s\to\infty} H_\mu(x(s)) = 0$.*

Proof. Since $\nabla\rho(a, \lambda, x)$ has rank n for all $(\lambda, x) \in \rho^{-1}(0)$, then by the parameterized Sard's theorem [13, Theorem 2.1] for almost all a (in the sense of Lebesgue measure) $\nabla\rho_a$ has full rank on $\rho_a^{-1}(0) \bigcap((0, 1) \times \mathbb{R}^n)$. It follows that $\gamma_a \bigcap((0, 1) \times \mathbb{R}^n)$ is a smooth curve and is therefore diffeomorphic either to a circle or to an interval. Because $\nabla_x \rho_a(0, a)$ has rank n, the implicit function theorem gives x as a function of λ in a neighborhood of $(0, a)$. Thus, γ_a cannot be diffeomorphic to a circle and must therefore be diffeomorphic to an interval with $(0, a)$ corresponding to one end of it. By continuity of ρ_a, all limit points of γ_a must lie in $\rho_a^{-1}(0)$ (with ρ_a's extended domain $[0, 1] \times \mathbb{R}^n$) so the only limit point of $\rho_a^{-1}(0)$ in $\{0\} \times \mathbb{R}^n$ is $(0, a)$. Since $\nabla\rho_a$ has full rank on γ_a, by the implicit function theorem, γ_a cannot have an end point in $(0, 1) \times \mathbb{R}^n$. Thus, if γ_a has finite arclength, it must reach a point $(\bar{\lambda}, \bar{x})$ with $\bar{\lambda} = 1$, in which case $\bar{x}$ is a zero of H_μ.

If γ_a does not have finite arclength, then let $(\lambda_k, x^k) := (\lambda(s_k), x(s_k))$ for some increasing unbounded sequence $\{s_k\}$. Suppose $(\bar{\lambda}, \bar{x})$ is an accumulation point of $\{\lambda_k, x^k\}$. By continuity, $\rho(\bar{\lambda}, \bar{x}) = 0$. By the above paragraph, $\bar{\lambda} > 0$. Suppose $0 < \bar{\lambda} < 1$. Since $\nabla\rho_a(\bar{\lambda}, \bar{x})$ has rank n, then by the implicit function theorem there is a neighborhood N of $(\bar{\lambda}, \bar{x})$ such that $N \bigcap \rho_a^{-1}(0)$ is diffeomorphic to an open interval and has finite arclength. Thus, for s_k sufficiently large, $(\lambda(s_k), x(s_k)) \notin N$, contradicting the assumption that $(\bar{\lambda}, \bar{x})$ is an accumlation point of $\{(\lambda_k, x^k)\}$. Thus, every accumulation point $(\bar{\lambda}, \bar{x})$ satisfies $\bar{\lambda} = 1$.

Now define $\hat{\lambda} = \liminf \lambda_k$. There exists a subsequence $\{(\lambda_j, x^j)\}$ such that $\lambda_j \to$

$\hat{\lambda}$. If $\{(\lambda_j, x^j)\}$ is bounded, then it has an accumulation point $(\hat{\lambda}, \bar{x})$ and, from the above paragraph, $\hat{\lambda} = 1$. If instead $\{(\lambda_k, x^k)\}$ is unbounded, then by Lemma 3.10, $\hat{\lambda} = 1$.

Finally, suppose $H(x^k) \not\to 0$. Then there is a subsequence $\{(\lambda_j, x^j)\}$ such that $\|H(x^j)\|$ is bounded away from 0. By Lemma 3.10, $\{(\lambda_j, x^j)\}$ must be bounded. But this yields a contradiction, since by (3.9), $H(x^j) \to 0$ (since $\lambda_j \to 1$, and $\{x^j\}$ is bounded). □

4. Implementation. We implemented the algorithm in Figure 3.1 with $H = H^{FB}$ and $H_\mu = H_\mu^{FB}$ defined as follows:

$$H_i^{FB}(x) := \psi^{FB}(l_i, u_i, x_i, F_i(x)),$$
$$(H_\mu^{FB})_i(x) := \tilde{\psi}^{FB}(l_i, u_i, x_i, F_i(x), \mu),$$

where obvious limits are used to define the function when either bound is infinite; thus, if $l_i = -\infty$, then $H_i^{FB}(x) := -\phi^{FB}(u_i - x_i, -F_i(x))$ and $(H_\mu^{FB}(x))_i := -\phi_\mu^{FB}(u_i - x_i, -F_i(x), \mu)$; if $u_i = \infty$, then $H_i^{FB}(x) := \phi^{FB}(x_i - l_i, F_i(x))$, $(H_\mu^{FB}(x))_i := \tilde{\phi}^K(x_i - l_i, F_i(x), \mu)$; and if $l_i = -\infty$ and $u_i = \infty$, then $H_i^{FB}(x) := F_i(x)$ and $H_\mu^{FB}(x) := F_i(x)$. To track the homotopy zero curves, we used the FIXPDF algorithm from HOMPACK.

To use the generalized Newton method from Figure 2.1 to find a zero of H^{FB}, we need to establish that H^{FB} is semismooth. The following theorem was proved in [4].

THEOREM 4.1. *If F is continuously differentiable on $\mathbb{R}^n$, then the following hold:*

1. *H^{FB} is semismooth on $\mathbb{R}^n$.*
2. *If for each i, F_i is twice continuously differentiable with Lipschitz continuous Hessian, then H^{FB} is strongly semismooth everywhere.*
3. *The natural merit function $\theta := \frac{1}{2}(H^{FB}(\cdot))^T H^{FB}(\cdot)$ is continuously differentiable, with gradient given by $\nabla\theta(x) = V^T H^{FB}(x)$, where V is any element of $\partial H^{FB}(x)$.*

Observe that Step 2 of the generalized Newton algorithm requires choosing an element of $\partial_B H^{FB}(x^k)$ or $\partial_B H_\mu^{FB}(x^k)$. We now address the question of how to calculate such an element. To do this, we shall need the following lemma, which generalizes [16, Proposition 3.1].

LEMMA 4.2.

$$\partial H_\mu^{FB}(x) \subset \{D_a(x) + D_b(x)\nabla F(x)\}. \tag{4.1}$$

Here $D_a(x)$ and $D_b(x)$ are $n \times n$ diagonal matrices whose ith diagonal elements are given by

$$(D_a)_{ii}(x) := a_i(x) + b_i(x)c_i(x), \quad (D_b)_{ii}(x) := b_i(x)d_i(x), \tag{4.2}$$

where

$$\begin{aligned} a_i(x) &= 1 - \frac{x_i - l_i}{\sqrt{(x_i - l_i)^2 + \tilde{\phi}^K(u_i - x_i, -F_i(x), \mu)^2 + 2\mu}}, \\ b_i(x) &= 1 + \frac{\phi(u_i - x_i, -F_i(x))}{\sqrt{(x_i - l_i)^2 + \tilde{\phi}^K(u_i - x_i, -F_i(x), \mu)^2 + 2\mu}} \end{aligned} \tag{4.3}$$

if $(x_i - l_i, F_i(x), \mu) \neq (0, 0, 0)$, or

$$(a_i(x), b_i(x)) \in \left\{(1 - \xi, 1 - \rho) \in \mathbb{R}^2 \mid \|(\xi, \rho)\| \leq 1\right\} \tag{4.4}$$

if $(x_i - l_i, F_i(x), \mu) = (0,0,0)$, *and*

$$\begin{aligned} c_i(x) &= 1 + \frac{x_i - u_i}{\sqrt{(x_i - u_i)^2 + F_i(x)^2 + 2\mu}}, \\ d_i(x)) &= 1 + \frac{F_i(x)}{\sqrt{(x_i - u_i)^2 + F_i(x)^2 + 2\mu}} \end{aligned} \tag{4.5}$$

if $(x_i - u_i, F_i(x), \mu) \neq (0,0,0)$, *or*

$$(c_i(x), d_i(x)) \in \left\{ (1+\xi, 1+\rho) \in \mathbb{R}^2 \mid \|(\xi,\rho)\| \leq 1 \right\} \tag{4.6}$$

if $(x_i - u_i, F_i(x), \mu) = (0,0,0)$.

Note that in (4.3) and (4.5), if either l_i or u_i is infinite, then the obvious limits are used to define the fractions. Thus, if $l_i = -\infty$, then $(a_i(x), b_i(x)) = (0,1)$, and if $u_i = \infty$, then $(c_i(x), d_i(x)) = (0,1)$.

Proof. For simplicity of notation, let $H_\mu := H_\mu^{FB}$ and let $\phi_\mu := \tilde{\phi}^K(\cdot,\cdot,\mu)$. By [14, Proposition 2.6.2(e)],

$$\partial H_\mu(x) \subset (\partial(H_\mu)_1(x) \times \cdots \times \partial(H_\mu)_n(x)).$$

Thus, it suffices to prove that, for each i,

$$\partial(H_\mu)_i(x) \subset \{(a_i(x) + b_i(x)c_i(x))e^{i^T} + b_i(x)d_i(x)\nabla F_i(x)\}, \tag{4.7}$$

where $a_i(x), b_i(x), c_i(x), d_i(x)$ satisfy (4.3)–(4.6).

To prove this result, let $g_i : \mathbb{R}^n \to \mathbb{R}$ be defined by $g_i(x) := -\phi_\mu(u_i - x_i, -F_i(x))$, and let $h_i : \mathbb{R}^n \to \mathbb{R}^2$ be defined by $h_i(x) := (x_i - l_i, g_i(x))$. We then have that $(H_\mu)_i(x) = \phi_\mu(h_i(x))$. Our first step is to show that $\partial(H_\mu)_i(x) = \partial\phi_\mu(h_i(x))\partial h_i(x)$.

We consider two cases. In the first case, suppose that $\mu \neq 0$ or $h_i(x) \neq (0,0)$. It follows that ϕ_μ is continuously differentiable at $h_i(x)$. Furthermore, since F is continuously differentiable and ϕ_μ is Lipschitz, h_i is locally Lipschitz at x. Thus, by [14, Theorem 2.6.6], $\partial(H_\mu)_i(x) = \partial\phi_\mu(h_i(x))\partial h_i(x)$.

In the second case, suppose that $\mu = 0$ and $h_i(x) = (0,0)$. It then follows that $u_i - x_i = u_i - l_i > 0$, so ϕ_μ is continuously differentiable at $(u_i - x_i, -F_i(x))$, and therefore h_i is continuously differentiable at x. By the corollary to [14, Proposition 2.2.1], h_i is strictly differentiable at x. Furthermore, since ϕ_μ is Lipschitz and convex [17], then by [14, Proposition 2.3.6(b)], ϕ_μ is regular everywhere. Thus, by [14, Theorem 2.3.9(iii)], $\partial(H_\mu)_i(x) = \partial\phi_\mu(h_i(x))\partial h_i(x)$.

We now look at the terms $\partial\phi_\mu(h_i(x))$ and $\partial h_i(x)$. It is easily shown that

$$\partial\phi_\mu(a,b) = \begin{cases} \left\{ \left(1 - \dfrac{a}{\sqrt{a^2+b^2+2\mu}}, 1 - \dfrac{b}{\sqrt{a^2+b^2+2\mu}} \right) \right\}, & (a,b,\mu) \neq 0, \\ \left\{ (1-\xi, 1-\rho) \in \mathbb{R}^2 \mid \|(\xi,\rho)\| \leq 1 \right\}, & (a,b,\mu) = 0. \end{cases}$$

Also,

$$\partial h_i(x)^T = \left\{ (e^i, \sigma^i) \,\middle|\, \sigma^i \in \partial g_i(x) \right\},$$

where e^i is the ith column of the identity matrix. Thus,

$$\partial H_i(x) = \left\{ a_i(x)e^{i^T} + b_i(x)\sigma^i \,\middle|\, \sigma^i \in \partial g_i(x), a_i(x), b_i(x) \text{ satisfy (4.3) and (4.4)} \right\}.$$

Step 1. Set $\beta_l := \{i \mid x_i - l_i = 0 = F_i(x)\}$, $\beta_u := \{i \mid u_i - x_i = 0 = F_i(x)\}$.
Step 2. Choose $z \in \mathbb{R}^n$ such that $z_i \neq 0$ for all $i \in \beta_l \bigcup \beta_u$.
Step 3. For each i, if $i \notin \beta_u$ or $\mu \neq 0$, set

$$c_i(x) := 1 + \frac{x_i - u_i}{\sqrt{(x_i - u_i)^2 + F_i(x)^2 + 2\mu}},$$
$$d_i(x) := 1 + \frac{F_i(x)}{\sqrt{(x_i - u_i)^2 + F_i(x)^2 + 2\mu}};$$

else if $\mu = 0$ and $i \in \beta_u$, set

$$c_i(x) := 1 + \frac{z_i}{\|(z_i, \nabla F_i(x)z)\|},$$
$$d_i(x) := 1 + \frac{\nabla F_i(x)z}{\|(z_i, \nabla F_i(x)z)\|}.$$

Step 4. For each i, if $i \notin \beta_l$ or $\mu \neq 0$, set

$$a_i(x) := 1 - \frac{x_i - l_i}{\sqrt{(x_i - l_i)^2 + \tilde{\phi}^K(u_i - x_i, -F_i(x), \mu)^2 + 2\mu}},$$
$$b_i(x) := 1 + \frac{\phi^{FB}(u_i - x_i, -F_i(x))}{\sqrt{(x_i - l_i)^2 + \tilde{\phi}^K(u_i - x_i, -F_i(x), \mu)^2 + 2\mu}};$$

else if $\mu = 0$ and $i \in \beta_l$, set

$$a_i(x) := 1 - \frac{z_i}{\|(z_i, c_i(x)z_i + d_i(x)\nabla F_i(x)z)\|},$$
$$b_i(x) := 1 + \frac{c_i(x)z_i + d_i(x)\nabla F_i(x)z}{\|(z_i, c_i(x)z_i + d_i(x)\nabla F_i(x)z)\|}.$$

Step 5. For each i, set

$$V_{i\cdot} := (a_i(x) + b_i(x)c_i(x))e^{i^T} + b_i(x)d_i(x)\nabla F_i(x).$$

FIG. 4.1. *Procedure to evaluate an element of* $\partial_B H_\mu^{FB}(x)$.

By similar arguments, we get

$$\partial g_i(x) = \left\{ c_i(x)e^{i^T} + d_i(x)\nabla F_i(x) \mid c_i(x), d_i(x) \text{ satisfy (4.5) and (4.6)} \right\}.$$

Combining these last two relations, we see that (4.7) is satisfied as an equality. ∎

Notice that if H_μ^{FB} is differentiable, then the right-hand side of (4.1) is a singleton, so (4.1) is satisfied as an equality. Figure 4.1 describes a simple procedure for calculating an element of $\partial_B H_\mu^{FB}(x)$.

THEOREM 4.3. *The matrix V calculated by the procedure given in Figure* 4.1 *is an element of* $\partial_B H_\mu^{FB}(x)$.

Proof. For simplicity of notation, let $H_\mu := H_\mu^{FB}$ and $\phi_\mu := \tilde{\phi}^K(\cdot, \cdot, \mu)$. If $\mu \neq 0$,

then H_μ is differentiable, and by Lemma 4.2, the Jacobian of H_μ is as calculated in Figure 4.1.

We now consider the case in which $\mu = 0$. In similar fashion to the proof of [15, Theorem 7.1], we build a sequence of points $\{y^k\}$, where $H_\mu(y^k)$ is differentiable and such that $\nabla H_\mu(y^k)$ tends to V. The theorem then follows by the definition of the B-subdifferential.

Let $y^k := x + \epsilon_k z$, where z is the vector of Step 2 of Figure 4.1 and $\{\epsilon_k\}$ is a sequence of positive numbers converging to 0. For $i \notin \beta_l \bigcup \beta_u$, either $x_i \neq l_i$ and $x_i \neq u_i$, or $F_i(x) \neq 0$, and for $i \in \beta_l \bigcup \beta_u$, $z_i \neq 0$. Thus, if ϵ_k is small enough, either $y_i^k \neq l_i$ and $y_i^k \neq u_i$, or $F_i(y^k) \neq 0$. In either case, H is differentiable at y^k.

We now show that for each i, $\lim_{k\to\infty} \nabla(H_\mu)_i(y^k) = V_{i\cdot}$. If either l_i or u_i is infinite, the result is given by [15, Theorem 7.1] by a simple change of variables. Thus, without loss of generality, we assume that l_i and u_i are both finite.

By Lemma 4.2, $\nabla(H_\mu)_i(y^k)$ is given by

$$(a_i(y^k) + b_i(y^k)c_i(y^k))e^i + b_i(y^k)d_i(y^k)\nabla F_i(y^k),$$

where a_i, b_i, c_i, d_i are defined by (4.3) and (4.5).

We now consider three cases.

Case 1. $i \notin \beta_l \bigcup \beta_u$. In this case, by continuity, $\lim_{k\to\infty} \nabla(H_\mu)_i(y^k) = V_{i\cdot}$.

Case 2. $i \in \beta_u$. In this case, $x_i = u_i$, so $y_i^k - u_i = \epsilon_k z_i$, so

$$\begin{aligned} c_i(y^k) &= 1 + \frac{\epsilon_k z_i}{\|(\epsilon_k z_{,i}\, F_i(y^k))\|}, \\ d_i(y^k) &= 1 + \frac{F_i(y^k)}{\|(\epsilon_k z_i, F_i(y^k))\|}. \end{aligned} \tag{4.8}$$

Since F is continuously differentiable and $F_i(x) = 0$, we can use a Taylor series expansion to get

$$F_i(y^k) = F_i(x) + \epsilon_k \nabla F_i(\zeta^k) z = \epsilon_k \nabla F_i(\zeta^k) z \quad \text{with } \zeta^k \in [x, y^k].$$

Substituting this expression into (4.8), we see that

$$\begin{aligned} \lim_{k\to\infty} c_i(y^k) &= 1 + \frac{z_i}{\|(z_i, \nabla F_i(x) z)\|}, \\ \lim_{k\to\infty} d_i(y^k) &= 1 + \frac{\nabla F_i(x) z}{\|(z_i, \nabla F_i(x) z)\|}. \end{aligned}$$

Thus, $\lim_{k\to\infty} \nabla(H_\mu)_i(y^k) = V_{i\cdot}$.

Case 3. $i \in \beta_l$. In this case, $x_i = l_i$ and $F_i(x) = 0$. Clearly, $x_i \neq u_i$, so ϕ is continuously differentiable in a neighborhood of $(u_i - x_i, -F_i(x))$. Thus, using an argument similar to that above, we get

$$\lim_{k\to\infty} a_i(y^k) = 1 - \frac{z_i}{\|(z_i, \nabla\phi(u_i - x_i, -F_i(x)) z)\|}, \tag{4.9}$$

$$\lim_{k\to\infty} b_i(y^k) = 1 + \frac{\nabla\phi(u_i - x_i, -F_i(x)) z}{\|(z_i, \nabla\phi(u_i - x_i, -F_i(x)) z)\|}. \tag{4.10}$$

Finally, $\nabla\phi(u_i - x_i, -F_i(x)) = c_i(x)e^i + d_i(x)\nabla F_i(x)$, where $c_i(x)$ and $d_i(x)$ are given by (4.5). Substituting this expression into (4.9) and (4.10), we see that $\lim_{k\to\infty} \nabla(H_\mu)_i(y^k) = V_{i\cdot}$. □

4.1. Tracking the homotopy zero curve. The above discussion describes how to use the Fischer–Burmeister MCP function and the Kanzow MCP smoother within our algorithmic framework. It remains to discuss how to track the homotopy zero curve of H_μ. To do this, we used the FIXPDF routine from HOMPACK. FIXPDF tracks the zero curve using an ODE-based algorithm. There are two user-defined parameters, which govern how accurately the zero curve is tracked: *arctol* specifies the local error allowed the ODE solver when following the zero curve, and *eps* specifies the local error allowed the ODE solver when very near the solution. We used choices of $arctol = 10^{-4}$ and $eps = 10^{-6}$. However, if the algorithm failed, we restarted with $arctol = 10^{-5}$. It should be noted that HOMPACK includes other curve tracking routines, which are faster than FIXPDF. We chose FIXPDF because it is believed to be the most robust algorithm.

We terminated the homotopy curve tracking whenever a point was discovered with a sufficiently improved merit function. That is, rather than following the zero curve all the way to the solution, we stopped as soon as a point $\hat{x}^k$ was generated with $\theta(\hat{x}^k) \leq \zeta\theta(\tilde{x}^k)$, where $\zeta \in (0, 1)$. For our testing we chose $\zeta = 0.1$.

4.2. Scaling. One potential difficulty with the homotopy algorithm is that if the Jacobian matrix is poorly conditioned at the solution, it can be very difficult to track the zero curve. To address this difficulty, we incorporated the following scaling method, which is based on the fact that $\mathrm{MCP}(F, \mathbb{B})$ is equivalent to $\mathrm{MCP}(DF, \mathbb{B})$, where D is a diagonal matrix with strictly positive entries. At the beginning of each major iteration (Step 2 in Figure 3.1), the algorithm calculates the 1-norm of each row of $\nabla F(x^k)$. If $\left\|\nabla F_i(x^k)\right\|_1 > 100$, then F_i is scaled by a factor of $10/|\nabla f(x^k)_{ii}|$. A similar heuristic was used by Chen and Mangasarian [11]. We terminated either when the unscaled merit function satisfied the stopping criterion $\theta(x^k) < \epsilon$ or when the merit function for the scaled problem satisfied the tighter stopping criterion $\theta(x^k) < 10^{-4}\epsilon$.

4.3. Computational results. The above algorithmic framework was coded in ANSI C, using double precision arithmetic and incorporating an interface with the GAMS modeling language. We used parameter values $\sigma = .1$, $\alpha = 0.5$, $m_{max} = 10$, $\beta = .5$, and $\epsilon = 10^{-12}$. At each iteration of Step 4 in Figure 3.1, we set $\mu = \beta^2\theta(\tilde{x}^k)/36n$.

The algorithm was run on all of the problems with fewer than 110 variables in the MCPLIB and GAMSLIB problem libraries. A listing of these problems is provided in [6]. Additionally, the algorithm was run on the 125-variable vonthmcp problem, which is known to be particularly challenging. Results are summarized in Tables 1 and 2 only for those problems that required at least one call to the homotopy algorithm. Problems not appearing in these two tables were solved by the damped Newton method without using the homotopy algorithm.

For each problem, we list the size of the problem (that is, the number of variables), the starting point used, the number of calls to the homotopy algorithm, the number of Jacobian evaluations required (both by FIXPDF and by the damped Newton method), and the final value of θ for the unscaled problem. Notice that in some cases, the algorithm terminated based on the stopping criteria for the scaled problem. In these cases, the final unscaled θ values are larger than 10^{-12}.

The algorithm solved all but four of the 215 test cases in the two libraries. It is particularly noteworthy that the method solved all of the pgvon105, pgvon106, vonthmcp, and billups problems, since these problems were troublesome for all of the algorithms tested in [6]. The strength of the algorithm is perhaps best illustrated by the billups problem, whose merit function has a local minimum of roughly 10^{-4} near

TABLE 1
MCPLIB test problems.

Problem name	Size	St. pt.	Homotopy calls	Jac. evals		θ final
				FIXPDF	Newton	
bertsekas	15	1	2	689	14	2.02e-21
bertsekas	15	2	1	31	20	1.10e-15
bertsekas	15	3	2	523	20	8.28e-22
bertsekas	15	4	2	689	14	2.02e-21
bertsekas	15	5	1	33	16	1.19e-21
bertsekas	15	6	1	33	18	6.59e-22
billups	1	1	1	88	2	5.55e-30
billups	1	2	1	87	3	5.55e-30
billups	1	3	1	87	3	5.55e-30
colvdual	20	1	1	33	11	5.89e-22
colvdual	20	2	2	172	11	6.94e-14
colvdual	20	4	1	33	17	2.25e-22
colvnlp	15	1	1	31	11	9.56e-17
colvnlp	15	4	1	31	12	2.09e-18
colvnlp	15	5	1	31	12	1.90e-18
colvnlp	15	6	1	31	12	1.34e-18
ehl_k40	41	1	1	33	15	1.46e-19
ehl_k40	41	3	1	168	20	8.07e-12
ehl_k60	61	2	1	33	44	5.07e-22
ehl_k60	61	3	2	211	55	3.52e-14
ehl_k80	81	2	1	33	51	1.28e-17
ehl_kost	101	1	1	33	18	1.61e-17
ehl_kost	101	2	1	33	50	2.07e-17
freebert	15	1	2	499	12	6.05e-22
freebert	15	3	2	496	7	2.98e-22
freebert	15	4	1	460	8	2.21e-13
freebert	15	5	1	33	16	1.22e-21
freebert	15	6	2	477	12	3.43e-22
freebert	15	7	1	31	14	7.08e-16
hanskoop	14	1	3	137	11	4.46e-16
hanskoop	14	3	1	74	24	3.30e-15
hanskoop	14	5	1	47	19	3.80e-13
hanskoop	14	7	1	304	27	1.44e-13
hanskoop	14	9	1	115	31	5.93e-21
josephy	4	1	1	22	11	1.23e-22
josephy	4	2	2	51	8	4.45e-14
josephy	4	3	1	31	12	1.22e-13
josephy	4	4	2	38	7	5.21e-14
josephy	4	5	1	26	7	3.70e-16
josephy	4	6	1	29	10	6.62e-15

the starting points. Algorithms that rely on descent of a merit function often fail on this problem because it is very difficult to escape the local minimum. However, the homotopy algorithm had no difficulties since the global monotonicity assumption is satisfied.

5. Conclusions. The algorithm described in this paper represents a qualitatively different approach for solving complementarity problems. Because of its basis in probability-one homotopy algorithms, it has a strong global convergence theory that suggests it may be successful on problems that cannot be solved by other methods. The fact that the method was able to solve all but four of the test cases supports this claim. However, the method, at present, is very slow. On a number of test problems, the algorithm had to calculate an extremely large number of Jacobian matrices. When compared to the performance of other recent algorithms [6, 21] on this test library,

TABLE 1 (*cont.*)

Problem name	Size	St. pt.	Homotopy calls	Jac. evals FIXPDF	Jac. evals Newton	θ final
josephy	4	7	1	20	8	6.06e-15
josephy	4	8	1	17	5	2.65e-20
kojshin	4	2	2	39	10	4.38e-15
kojshin	4	3	1	98	16	9.42e-14
kojshin	4	4	1	28	7	7.27e-18
kojshin	4	6	2	56	6	4.58e-14
pgvon105	105	1	5	2831	294	2.21e-12
pgvon105	105	2	5	454	48	1.35e-14
pgvon105	105	3	4	126	31	3.47e-15
pgvon105	106	4	3	2480	101	2.69e-14
pgvon106	106	1	5	533	41	2.80e-13
pgvon106	106	2	8	7131	61	2.35e-13
pgvon106	106	3	3	337	62	8.15e-17
pgvon106	106	4	5	556	81	1.04e-09
pgvon106	106	5	8	4449	126	4.18e-11
pgvon106	106	6	3	684	56	5.09e-12
pies	42	1	6	760	23	2.66e-14
powell	16	1	4	178	14	3.37e-15
powell	16	2	4	3014	19	2.99e-14
powell	16	3	5	756	17	7.31e-15
powell	16	4	6	1259	9	2.04e-12
powell	16	5	1	9991	2	1.34e+02 (failed)
powell	16	6	4	83	13	6.10e-15
scarfanum	13	1	2	78	12	1.67e-13
scarfanum	13	2	2	128	18	5.04e-13
scarfanum	13	3	2	60	13	1.78e-20
scarfasum	14	1	1	34	10	2.73e-13
scarfasum	14	2	2	67	10	3.79e-14
scarfasum	14	3	2	91	15	3.39e-13
scarfbnum	39	1	1	125	27	1.93e-13
scarfbnum	39	2	2	615	70	1.14e-13
scarfbsum	40	2	3	983	5	5.17e-14
sppe	27	2	1	18	8	3.62e-13
sppe	27	3	1	51	7	5.04e-15
tobin	42	1	3	105	15	1.16e-16
tobin	42	2	1	22	52	1.12e-12
tobin	42	3	2	113	34	1.17e-19

the homotopy method is not competitive in terms of computer time. Nevertheless, because of its potential to solve more difficult problems, the homotopy method may, in many situations, be more efficient in real time, since it may require less human intervention to produce a solution. Further, the generalized damped Newton method used in Step 2 of Figure 3.1 fails often, necessitating the use of the homotopy algorithm in many cases. In principle, a more sophisticated descent algorithm could be used so that the homotopy method would only be needed in rare circumstances.

REFERENCES

[1] J. C. ALEXANDER, *The topological theory of an embedding method*, in Continuation Methods, H. Wacker, ed., Academic Press, New York, 1978, pp. 37–68.

[2] J. C. ALEXANDER, R. B. KELLOGG, T.-Y. LI, AND J. A. YORKE, *Piecewise Smooth Continuation*, manuscript, 1979.

[3] J. C. ALEXANDER, T.-Y. LI, AND J .A. YORKE, *Piecewise smooth homotopies*, in Homotopy Methods and Global Convergence, B. C. Eaves, F. J. Gould, H.-O. Peitgen, and M. J. Todd, eds., Plenum Press, New York, 1983, pp. 1–14.

TABLE 2
GAMSLIB test problems.

Problem name	Size	St. pt.	Homotopy calls	Jac. evals		θ final
				FIXPDF	Newton	
cirimge	115	3	1	73	16	4.37e-17
harkmcp	32	1	1	166	8	8.20e-15
harkmcp	92	4	4	82	13	4.81e-12
harmge	11	1	1	5461	6	3.45e+14 (failed)
harmge	11	2	2	168	55	3.42e-28
sammge	23	2	1	29	10	7.29e-23
sammge	23	3	1	131	9	2.33e-17
sammge	23	4	1	30	10	4.80e-20
sammge	23	5	1	147	11	1.18e-16
sammge	23	6	2	66	12	5.24e-14
sammge	23	7	1	146	12	8.84e-17
sammge	23	8	2	91	12	6.99e-13
sammge	23	9	1	202	12	4.56e-23
sammge	23	11	2	66	8	4.57e-15
sammge	23	12	1	131	9	3.13e-17
sammge	23	13	1	45	11	2.56e-15
sammge	23	15	1	44	21	1.07e-19
sammge	23	16	2	56	11	9.26e-14
sammge	23	17	4	142	19	2.15e-14
sammge	23	18	2	92	21	3.48e-20
threemge	14	11	1	2443	4	8.72e+10 (failed)
transmcp	11	4	2	51	11	8.45e-13
vonthmcp	125	1	1	274	79	1.97e-15
vonthmge	80	1	1	8115	30	6.98e+24 (failed)

[4] S. C. BILLUPS, *Algorithms for Complementarity Problems and Generalized Equations*, Ph.D. thesis, University of Wisconsin–Madison, Madison, WI, 1995.
[5] S. C. BILLUPS, *Improving the robustness of descent-based methods for semi-smooth equations using proximal perturbations*, Math. Program., 87 (2000), pp. 153–176.
[6] S. C. BILLUPS, S. P. DIRKSE, AND M. C. FERRIS, *A comparison of large scale mixed complementarity problem solvers*, Comput. Optim. Appl., 7 (1997), pp. 3–25.
[7] S. C. BILLUPS AND M. C. FERRIS, *QPCOMP: A quadratic program based solver for mixed complementarity problems*, Math. Programming, 76 (1997), pp. 533–562.
[8] B. CHEN AND X. CHEN, *A global and local superlinear continuation-smoothing method for P_0 and R_0 NCP or monotone NCP*, SIAM J. Optim., 9 (1999), pp. 624–645.
[9] B. CHEN AND P. T. HARKER, *A non-interior-point continuation method for linear complementarity problems*, SIAM J. Matrix Anal. and Appl., 14 (1993), pp. 1168–1190.
[10] C. CHEN AND O. L. MANGASARIAN, *Smoothing methods for convex inequalities and linear complementarity problems*, Math. Programming, 78 (1995), pp. 51–70.
[11] C. CHEN AND O. L. MANGASARIAN, *A class of smoothing functions for nonlinear and mixed complementarity problems*, Comput. Optim. Appl., 5 (1996), pp. 97–138.
[12] X. CHEN AND L. QI, *A parameterized Newton method and a quasi-Newton method for solving nonsmooth equations*, Comput. Optim. Appl., 3 (1994), pp. 157–179.
[13] S.-N. CHOW, J. MALLET-PARET, AND J. A. YORKE, *Finding zeros of maps: Homotopy methods that are constructive with probability one*, Math. Comput., 32 (1978), pp. 887–899.
[14] F. H. CLARKE, *Optimization and Nonsmooth Analysis,* John Wiley, New York, 1983.
[15] T. DE LUCA, F. FACCHINEI, AND C. KANZOW, *A semismooth equation approach to the solution of nonlinear complementarity problems*, Math. Programming, 75 (1996), pp. 407–439.
[16] F. FACCHINEI AND J. SOARES, *A new merit function for nonlinear complementarity problems and a related algorithm*, SIAM J. Optim., 7 (1997), pp. 225–247.
[17] A. FISCHER, *A special Newton–type optimization method*, Optimization, 24 (1992), pp. 269–284.
[18] A. FISCHER, *An NCP–function and its use for the solution of complementarity problems*, in Recent Advances in Nonsmooth Optimization, D. Z. Du, L. Qi, and R. S. Womersley, eds., World Scientific Publishers, River Edge, NJ, 1995, pp. 88–105.
[19] S. A. GABRIEL AND J. J. MORÉ, *Smoothing of mixed complementarity problems*, in Complementarity and Variational Problems: State of the Art, M. C. Ferris and J. S. Pang, eds.,

SIAM, Philadelphia, 1997, pp. 105–116.
[20] C. Kanzow, *Some noninterior continuation methods for linear complementarity problems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 851–868.
[21] C. Kanzow and H. Pieper, *Jacobian smoothing methods for nonlinear complementarity problems*, SIAM J. Optim., 9 (1999), pp. 342–373.
[22] L. Qi, *Convergence analysis of some algorithms for solving nonsmooth equations*, Math. Oper. Res., 18 (1993), pp. 227–244.
[23] L. Qi, *Regular pseudo-smooth NCP and BVIP functions and globally and quadratically convergent generalized Newton methods for complementarity and variational inequality problems*, Math. Oper. Res., 24 (1999), pp. 440–471.
[24] L. Qi and X. Chen, *A globally convergent successive approximation method for severely nonsmooth equations*, SIAM J. Control Optim., 33 (1995), pp. 402–418.
[25] H. Sellami, *A Continuation Method for Normal Maps*, Ph.D. thesis, University of Wisconsin–Madison, Madison, WI, 1994.
[26] H. Sellami and S. M. Robinson, *Homotopies based on nonsmooth equations for solving nonlinear variational inequalities*, in Nonlinear Optimization and Applications, G. Di Pillo and F. Giannessi, eds., Plenum Press, New York, 1996, pp. 327–343.
[27] H. Sellami and S. M. Robinson, *Implementation of a continuation method for normal maps*, Math. Programming, 76 (1997), pp. 563–578.
[28] S. Smale, *Algorithms for solving equations*, in Proceedings of the International Congress of Mathematicians, 1986, pp. 172–195.
[29] P. Tseng, *Growth behavior of a class of merit functions for the nonlinear complementarity problem*, J. Optim. Theor. Appl., 89 (1996), pp. 17–37.
[30] L. T. Watson, *An algorithm that is globally convergent with probability one for a class of nonlinear two-point boundary value problems*, SIAM J. Numer. Anal., 16 (1979), pp. 394–401.
[31] L. T. Watson, *A globally convergent algorithm for computing fixed points of C^2 maps*, Appl. Math. Comput., 5 (1979), pp. 297–311.
[32] L. T. Watson, *Solving the nonlinear complementarity problem by a homotopy method*, SIAM J. Control Optim., 17 (1979), pp. 36–46.
[33] L. T. Watson, S. C. Billups, and A. P. Morgan, *Algorithm* 652*: HOMPACK: A suite of codes for globally convergent homotopy algorithms*, ACM Trans. Math. Software, 13 (1987), pp. 281–310.
[34] I. Zang, *A smoothing-out technique for min-max optimization*, Math. Programming, 19 (1980), pp. 61–71.

SIAM J. OPTIM.
Vol. 12, No. 3, pp. 606–626

A PROBABILITY-ONE HOMOTOPY ALGORITHM FOR NONSMOOTH EQUATIONS AND MIXED COMPLEMENTARITY PROBLEMS*

STEPHEN C. BILLUPS† AND LAYNE T. WATSON‡

Abstract. Convergence theory for a new probability-one homotopy algorithm for solving nonsmooth equations is given. This algorithm is able to solve problems involving highly nonlinear equations, where the norm of the residual has nonglobal local minima. The algorithm is based on constructing homotopy mappings that are smooth in the interior of their domains. The algorithm is specialized to solve mixed complementarity problems (MCP) through the use of MCP functions and associated smoothers. This specialized algorithm includes an option to ensure that all iterates remain feasible. Easily satisfiable sufficient conditions are given to ensure that the homotopy zero curve remains feasible, and global convergence properties for the MCP algorithm are proved. Computational results on the MCPLIB test library demonstrate the effectiveness of the algorithm.

Key words. nonsmooth equations, complementarity problems, homotopy methods, smoothing, path following

AMS subject classifications. 65F10, 65F50, 65H10, 65K10

PII. S105262340037758X

1. Introduction.

The primary attraction of homotopy algorithms is that they are able to reliably solve systems of equations involving highly nonlinear functions, where the norm of the residual may have nonglobal local minima. This is because, unlike line search or trust region methods, homotopy methods do not rely on descent of a merit function. Instead, they work by following a path, which under certain weak assumptions is known to lead to a solution. Standard probability-one homotopy algorithms require that the system of equations involve only *smooth* (C^2) functions. This paper presents the convergence theory for a new probability-one homotopy algorithm for solving *nonsmooth* systems of equations and specializes this algorithm to solve mixed complementarity problems. The algorithm uses smoothing functions to construct a homotopy mapping that is C^2 in the interior of its domain. This allows the zero curve of the homotopy mapping to be tracked using software from the HOMPACK90 suite of homotopy codes [31]. A preliminary version of this algorithm was presented at the Second International Conference on Complementarity Problems [5]. The algorithm proposed here has two significant improvements: first, a new end game strategy, which makes better use of available information about the behavior of the homotopy zero curve; second, an option for mixed complementarity problems that ensures that all iterates generated by the algorithm are feasible. This is important because many applications involve functions that are not defined outside of the feasible region. A similar feasibility property can be achieved for smoothing Newton methods [21].

*Received by the editors September 5, 2000; accepted for publication (in revised form) July 23, 2001; published electronically January 9, 2002.

http://www.siam.org/journals/siopt/12-3/37758.html

†Department of Mathematics, University of Colorado at Denver, Denver, CO 80217-3364 (sbillups@carbon.cudenver.edu). This author's research was partially supported through NSF grant DMS-9973321.

‡Departments of Computer Science and Mathematics, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061-0106 (ltw@cayuga.cs.vt.edu). This author's research was partially supported by AFOSR grant F496320-99-1-0128 and NSF grant DMS-9625968.

An extension of the probability-one homotopy theory to nonsmooth systems of equations is presented here. A globally convergent probability-one homotopy algorithm for nonsmooth systems of equations is then derived, with supporting convergence theory, and specialized for mixed complementarity problems. For the case of mixed complementarity problems, new convergence results are presented, which establish easily satisfiable sufficient conditions to ensure that the homotopy zero curve always remains strictly feasible.

The homotopy algorithm proposed here for nonsmooth systems of equations is similar in spirit to that in [23] and [24] (based on piecewise smooth maps). While different homotopy formulations might be theoretically equivalent in terms of solution power, the distinction between such piecewise smooth continuation methods and probability-one homotopy methods is significant for practical numerical computation. Probability-one homotopy methods are guaranteed to avoid numerical singularities, and a probability-one formulation can exploit the existence of very robust, accurate, and efficient mathematical software specifically tailored for such maps [30], [31].

In order to describe the algorithm, a significant amount of background material is needed. This is given in section 2, which discusses notation, nonsmooth equations, a generalized Newton method for nonsmooth equations (which will be used in the end game), probability-one homotopy methods, complementarity problems, and smoothing functions. Section 3 describes a probability-one homotopy algorithm for nonsmooth equations. This algorithm is then specialized to solve mixed complementarity problems in section 4. Section 5 addresses implementation details and computational results, and section 6 concludes.

2. Background.

2.1. Notation. When discussing vectors and vector-valued functions, subscripts are used to indicate components, whereas superscripts are used to indicate the iteration number or some other label. In contrast, for scalars or scalar-valued functions, subscripts refer to labels so that superscripts can be used for exponentiation. The vector of all ones is represented by e.

Unless otherwise specified, $\|\cdot\|$ denotes the Euclidean norm. For a set $C \subset \mathbb{R}^n$, $\pi_C(x)$ represents the orthogonal projection (with respect to the Euclidean norm) of x onto C. The symbol $\mathbb{R}_+$ refers to the nonnegative real numbers. The extended real numbers are denoted by $\overline{\mathbb{R}} := \mathbb{R} \bigcup \{-\infty, +\infty\}$.

Real-valued functions are denoted with lower-case letters like f or ϕ, whereas vector-valued functions are represented by upper-case letters like F or Φ. For a function $F : C \subset \mathbb{R}^n \to \mathbb{R}^m$, $\nabla F(x)$ is the $m \times n$ matrix whose i,jth element is $\partial F_i(x)/\partial x_j$. Let $D \subset \mathbb{R}^m$. Then $F^{-1}(D)$ is the set-valued inverse defined by $F^{-1}(D) := \{x \mid F(x) \in D\}$.

Given a function $F : \mathbb{R}^n \to \mathbb{R}^m$, the directional derivative of F at x in the direction d is denoted by $F'(x;d) := \lim_{t \downarrow 0} (F(x+td) - F(x))/t$, provided the limit exists.

2.2. Nonsmooth equations. This paper is concerned with solving equations of the form $F(x) = 0$, where the function $F : \mathbb{R}^n \to \mathbb{R}^n$ is locally Lipschitzian, but not necessarily continuously differentiable. Such nonsmooth equations provide a unifying framework for the study of many important classes of problems, including constrained optimization, finite-dimensional variational inequalities, complementarity problems, equilibrium problems, generalized equations, partial differential equations, and fixed point problems. The following definitions will be used throughout the paper.

By Rademacher's theorem, since F is locally Lipschitzian, it is differentiable almost everywhere. Let D_F be the set where F is differentiable. Define the *B-subdifferential* by

$$\partial_B F(x) := \left\{ V \;\middle|\; \exists \{x^k\} \to x, x^k \in D_F, \quad \text{with} \quad V = \lim_{k\to\infty} \nabla F(x_k) \right\}.$$

The Clarke subdifferential $\partial F(x)$ is the convex hull of $\partial_B F(x)$.

F is said to be *semismooth* [22] at x if it is directionally differentiable at x and for any $V \in \partial F(x+h)$, $h \to 0$,

$$Vh - F'(x;h) = o(\|h\|).$$

F is said to be *strongly semismooth* [10] if, additionally,

$$Vh - F'(x;h) = O(\|h\|^2).$$

A semismooth function $F : \mathbb{R}^n \to \mathbb{R}^n$ is *BD-regular* at x if all elements in $\partial_B F(x)$ are nonsingular, and F is *strongly regular* at x if all elements in $\partial F(x)$ are nonsingular.

2.3. Newton's method for nonsmooth equations. One approach to solving the nonsmooth equation $F(x) = 0$ is a generalization of Newton's method to semismooth equations, which was proposed by Qi [19]. Qi's method is used together with an Armijo line search in the end game of the homotopy algorithm proposed here. Qi's algorithm, which is discussed in detail in [3], is shown in Figure 2.1. In this algorithm θ is the merit function defined by $\theta(x) := \frac{1}{2} F(x)^T F(x)$. Theorem 2.1, which is restated from [22] and [10], shows that this algorithm has the same fast local convergence properties as the standard (smooth) Newton's method under natural generalizations of the standard assumptions.

Step 1 [Initialization] Select line search parameters α, $\sigma \in (0,1)$, a positive integer m_{max}, a starting point $x^0 \in \mathbb{R}^n$, and a stopping tolerance *tol*. Set $k = 0$.

Step 2 [Direction generation] Choose $V^k \in \partial_B F(x^k)$. If V^k is singular, stop, returning the point x^k along with a failure message. Otherwise choose the direction

$$d^k = -(V^k)^{-1} F(x^k). \tag{2.1}$$

Step 3 [Step length determination] Let m_k be the smallest nonnegative integer $m \le m_{max}$ such that

$$\theta(x^k + \alpha^m d^k) - \theta(x^k) \le -\sigma \alpha^m \theta(x^k). \tag{2.2}$$

If no such m_k exists, stop; the algorithm failed. Otherwise set $x^{k+1} = x^k + \alpha^{m_k} d^k$.

Step 4 [Termination check] If $\theta(x^{k+1}) < tol$, stop, returning the point x^{k+1}. Otherwise, return to step 2, with k replaced by $k+1$.

FIG. 2.1. *Generalized damped Newton method.*

THEOREM 2.1. *Suppose that x^* is a solution of $F(x) = 0$ and that F is semismooth and BD-regular at x^*. Then the iteration method defined by $x^{k+1} = x^k + d^k$, where d^k is given by (2.1), is well defined and convergent to x^* Q-superlinearly in a neighborhood of x^*. If F is strongly semismooth at x^*, the iteration sequence converges to x^* Q-quadratically.*

One consequence of this local convergence theorem is that within a neighborhood of a BD-regular solution x^*, the line search criterion (2.2) will be satisfied by $m_k = 0$. Thus, the inner algorithm will take full Newton steps and achieve the fast local convergence rates specified by the theorem.

The damped Newton method described above works very well when started near a solution, or when applied to problems that are nearly linear in the sense that their merit functions do not contain local minima that are not solutions.

For highly nonlinear problems, the damped Newton method tends to fail without a carefully chosen starting point. The reason, of course, is that unless started close to a solution, the iterates may converge only to a local minimum of the merit function. This motivates the consideration of homotopy methods, which are truly globally convergent.

2.4. Homotopy methods. The main theory underlying the present homotopy method is summarized in the following proposition from [5]. This proposition is similar to results presented in [26] and [8, Theorem 2.4]; however, it does not assume F itself to be differentiable. The path γ_a defined in the proposition "reaches a zero of F" in the sense that it contains a sequence $\{(\lambda_k, x^k)\}$ that converges to $(1, \bar{x})$, where $\bar{x}$ is a zero of F.

PROPOSITION 2.2. *Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be a Lipschitz continuous function and suppose there is a C^2 map*

$$\rho : \mathbb{R}^m \times [0, 1) \times \mathbb{R}^n \to \mathbb{R}^n$$

such that

1. *$\nabla\rho(a, \lambda, x)$ has rank n on the set $\rho^{-1}(\{0\})$;*
2. *the equation $\rho_a(0, x) = 0$, where $\rho_a(\lambda, x) := \rho(a, \lambda, x)$, has a unique solution $x^a \in \mathbb{R}^n$ for every fixed $a \in \mathbb{R}^m$;*
3. *$\nabla_x\rho_a(0, x^a)$ has rank n for every $a \in \mathbb{R}^m$;*
4. *ρ is continuously extendible (in the sense of Buck [6]) to the domain $\mathbb{R}^m \times [0, 1] \times \mathbb{R}^n$, and $\rho_a(1, x) = F(x)$ for all $x \in \mathbb{R}^n$ and $a \in \mathbb{R}^m$; and*
5. *γ_a, the connected component of $\rho_a^{-1}(\{0\})$ containing $(0, x^a)$, is bounded for almost every $a \in \mathbb{R}^m$.*

Then for almost every $a \in \mathbb{R}^m$ there is a zero curve γ_a of ρ_a, along which $\nabla\rho_a$ has rank n, emanating from $(0, x^a)$ and reaching a zero $\bar{x}$ of F at $\lambda = 1$. Further, γ_a does not intersect itself and is disjoint from any other zeros of ρ_a. Also, if γ_a reaches a point $(1, \bar{x})$ and F is strongly regular at $\bar{x}$, then γ_a has finite arc length.

Because γ_a is a smooth curve, it can be parameterized by its arc length away from $(0, x^a)$. This yields a function $(\lambda(s), x(s))$, representing the point on γ_a of arc length s away from $(0, x^a)$.

The construction of a globally convergent probability-one homotopy algorithm entails: (1) constructing a map ρ according to Proposition 2.2, (2) choosing $a \in \mathbb{R}^m$, (3) finding x^a solving $\rho_a(0, x) = 0$, and (4) tracking γ_a starting from $(0, x^a)$ until $\lambda = 1$. Assuming an appropriate ρ exists, the theory guarantees that for almost all a (in the sense of Lebesgue measure), γ_a exists and leads to a solution; hence the term "probability-one."

A simple (and occasionally useful in practice) homotopy mapping is $\rho : \mathbb{R}^n \times [0,1) \times \mathbb{R}^n \to \mathbb{R}^n$ given by

$$\rho(a, \lambda, x) := \lambda F(x) + (1 - \lambda)(x - a). \tag{2.3}$$

If F is C^2, then ρ trivially satisfies properties (1), (2), (3), and (4) but not necessarily (5) of Proposition 2.2. The following theorem gives conditions on F under which the fifth condition is satisfied. This result will be generalized to nonsmooth functions in Theorem 3.2.

THEOREM 2.3. (See [28].) *Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be a C^2 function such that, for some $\tilde{x} \in \mathbb{R}^n$ and $r > 0$,*

$$(x - \tilde{x})^T F(x) \geq 0 \text{ whenever } \|x - \tilde{x}\| = r. \tag{2.4}$$

Then F has a zero in a closed ball of radius r about $\tilde{x}$, and for almost every a in the interior of this ball there is a zero curve γ_a of

$$\rho_a(\lambda, x) := \lambda F(x) + (1 - \lambda)(x - a),$$

along which $\nabla \rho_a(\lambda, x)$ has full rank, emanating from $(0, a)$ and reaching a zero $\bar{x}$ of F at $\lambda = 1$. Further, γ_a has finite arc length if $\nabla F(\bar{x})$ is nonsingular.

The actual statement of the theorem in [28] fixes $\tilde{x} = 0$. However, the proof can be modified trivially to yield the more general theorem above. (See the proof of [5, Theorem 2.11] for the necessary modifications.) It is interesting to note that in many applications, (2.4) holds for all r sufficiently large (not just for some fixed r). This makes the choice of $\tilde{x}$ irrelevant. Furthermore, in such cases, a can be chosen arbitrarily (instead of from some neighborhood of $\tilde{x}$), thus making the method truly globally convergent (with probability one).

Equation (2.4) will be referred to as the *global monotonicity* property. If a C^2 function F possesses this property, these theoretical results have some profound implications: the guaranteed existence of a path between almost any starting point and a solution $\bar{x}$ to $F(x) = 0$, which has finite arc length if $\operatorname{rank} \nabla F(\bar{x}) = n$. In theory, to find a solution, one must simply follow the path to a point of γ_a where $\lambda = 1$. In practice, however, the task of constructing a ρ for which γ_a is short and smooth is very difficult, although this has been done for large classes of problems.

Several packages exist to solve root-finding problems using homotopy techniques [31]. The implementation here uses the routine STEPNX from the HOMPACK90 suite of software [30], [31, section 3], which tracks the zero curve of a homotopy mapping specified by the user.

2.5. Complementarity problems. Given a continuously differentiable function $G : \mathbb{R}^n \to \mathbb{R}^n$, the nonlinear complementarity problem NCP(G) is to find some $x \in \mathbb{R}^n$ so that

$$0 \leq x \perp G(x) \geq 0, \tag{2.5}$$

where $x \perp G(x)$ means that $x^T G(x) = 0$.

Given a rectangular region $\mathbb{B}_{l,u} := \prod_{i=1}^n [l_i, u_i] \subset \overline{\mathbb{R}}^n$ defined by two vectors l and u in $\mathbb{R}^n$, where $-\infty \leq l < u \leq \infty$, and a function $G : \mathbb{R}^n \to \mathbb{R}^n$, the mixed complementarity problem MCP($G, \mathbb{B}_{l,u}$) is to find an $x \in \mathbb{B}_{l,u}$ such that for each $i \in \{1, \ldots, n\}$, either (1) $x_i = l_i$ and $G_i(x) \geq 0$, (2) $G_i(x) = 0$, or (3) $x_i = u_i$ and $G_i(x) \leq 0$. This is equivalent to the condition that $\operatorname{mid}(x - l, x - u, G(x)) = 0$, where mid

represents the componentwise median function. When these conditions are satisfied, write $G(x) \perp x$ and say that x is complementary to $G(x)$. Assume henceforth that G is C^2.

It is well known that NCP(G) can be reformulated as a system of equations. This was first shown by Mangasarian [17]. An excellent review of reformulations of NCP can be found in [20]. To discuss such reformulations requires several definitions, which are equivalent to the NCP function and the BVIP function defined in [20].

DEFINITION 2.4. *A function $\phi : \mathbb{R}^2 \to \mathbb{R}$ is called an NCP function, provided $\phi(a, b) = 0$, if and only if* $\min(a, b) = 0$.

DEFINITION 2.5. *A function $\psi : \mathbb{R}\bigcup\{-\infty\} \times \mathbb{R}\bigcup\{\infty\} \times \mathbb{R}^2 \to \mathbb{R}$ is called an MCP function, provided $\psi(l, u, a, b) = 0$, if and only if* $\operatorname{mid}(a - l, a - u, b) = 0$.

It is useful to further distinguish NCP and MCP functions according to their orientations, as follows.

DEFINITION 2.6. *An NCP function ϕ is called* positively oriented *if, for all $a, b \in \mathbb{R}$,*

$$\operatorname{sign}(\phi(a, b)) = \operatorname{sign}(\min(a, b)).$$

An MCP function ψ is called positively oriented *if*

$$\operatorname{sign}(\psi(l, u, a, b)) = \operatorname{sign}(\operatorname{mid}(a - l, a - u, b))$$

for all $l \in \mathbb{R}\bigcup\{-\infty\}, u \in \mathbb{R}\bigcup\{\infty\}$, $l < u$, and $a, b \in \mathbb{R}$.

An NCP function that has been very popular recently is the Fischer–Burmeister function [14] $\phi : \mathbb{R}^2 \to \mathbb{R}$, defined by

$$\phi^{FB}(a, b) := a + b - \sqrt{a^2 + b^2}. \tag{2.6}$$

It is easily seen that $\phi^{FB}(a, b) = 0$ if and only if $0 \le a \perp b \ge 0$. Thus, by defining the function $F : \mathbb{R}^n \to \mathbb{R}^n$ by

$$F_i(x) := \phi^{FB}(x_i, G_i(x)), \tag{2.7}$$

it is clear that $x \in \mathbb{R}^n$ solves NCP(G) if and only if $F(x) = 0$.

While ϕ^{FB} is not differentiable at the origin, $(\phi^{FB})^2$ is continuously differentiable everywhere. This property, together with the fact that ϕ^{FB} is semismooth, makes this reformulation well suited for use in globalization strategies for nonsmooth Newton-based methods (see, for example, [9]).

Given a positively oriented NCP function ϕ, and the convention that $\phi(\infty, b) = \lim_{a\to\infty} \phi(a, b)$ and $\phi(a, \infty) = \lim_{b\to\infty} \phi(a, b)$, an MCP function ψ can be constructed using the following formula, first proposed in [1]:

$$\psi(l, u, a, b) := \phi(a - l, -\phi(u - a, -b)). \tag{2.8}$$

Constructing the function $F : \mathbb{R}^n \to \mathbb{R}^n$ by

$$F_i(x) := \psi(l_i, u_i, x_i, G_i(x)) \tag{2.9}$$

yields a reformulation of the MCP($G, \mathbb{B}_{l,u}$); $F(x) = 0$ if and only if x is a solution to MCP($G, \mathbb{B}_{l,u}$) [2].

Note that for the Fischer–Burmeister function, $\lim_{a\to\infty} \phi^{FB}(a, b) = b$ and $\lim_{b\to\infty} \phi^{FB}(a, b) = a$. Thus, for the MCP case, if l_i is finite and $u_i = \infty$, then $F_i(x) = \phi^{FB}(x_i - l_i, G_i(x))$; if u_i is finite and $l_i = -\infty$, then $F_i(x) = -\phi^{FB}(u_i - x_i, -G_i(x))$; and if neither bound is finite, then $F_i(x) = G_i(x)$.

2.6. Smoothing operators. Consider the system $F(x) = 0$, where F is a nonsmooth function, and suppose there exists a family of functions F^μ parameterized by a *smoothing parameter* μ so that $\lim_{\mu \downarrow 0} F^\mu = F$ in some sense. Under suitable conditions, the solutions to the systems $F^\mu(x) = 0$ converge to a solution to $F(x) = 0$ along a smooth trajectory [7].

DEFINITION 2.7. *Given a nonsmooth continuous function $\phi : \mathbb{R}^p \to \mathbb{R}$, a* smoother *for ϕ is a continuous function $\tilde{\phi} : \mathbb{R}^p \times \mathbb{R}_+ \to \mathbb{R}$ such that*

1. *$\tilde{\phi}(x, 0) = \phi(x)$, and*
2. *$\tilde{\phi}$ is continuously differentiable on the set $\mathbb{R}^p \times \mathbb{R}_{++}$.*

*If $\tilde{\phi}$ is C^2 on $\mathbb{R}^p \times \mathbb{R}_{++}$, call $\tilde{\phi}$ a C^2-*smoother*.*

For convenience, define $\phi_\mu(x) := \tilde{\phi}(x, \mu)$. To define smoothers for functions $F : \mathbb{R}^n \to \mathbb{R}^n$, say that $F^\mu : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}^n$ is a smoother for F if, for each $i \in \{1 \dots n\}$, F_i^μ is a smoother for F_i.

In the case of complementarity problems, the NCP functions and MCP functions generally have well understood nonsmoothness structure, so C^2-smoothers for these functions can usually be easily constructed. As an example, the following C^2-smoother for the Fischer–Burmeister function was proposed by Kanzow [16]:

$$\tilde{\phi}^K(a, b, \mu) := a + b - \sqrt{a^2 + b^2 + 2\mu}. \tag{2.10}$$

The following smoother is more useful here, since its partial derivative with respect to μ is bounded near the origin:

$$\tilde{\phi}^{BW}(a, b, \mu) := a + b - \sqrt{a^2 + b^2 + \mu^2}. \tag{2.11}$$

Given a smoother $\tilde{\phi}$ for an NCP function ϕ and the convention that $\tilde{\phi}(\infty, b, \mu) = \lim_{a \to \infty} \tilde{\phi}(a, b, \mu)$ and $\tilde{\phi}(a, \infty, \mu) = \lim_{b \to \infty} \tilde{\phi}(a, b, \mu)$, a smoother $\tilde{\psi}$ for the MCP function ψ defined by (2.8) can be constructed according to the formula

$$\tilde{\psi}(l, u, a, b, \mu) := \phi_\mu(a - l, -\phi_\mu(u - a, -b)). \tag{2.12}$$

Smoothers for (2.7) and (2.9) are then given, respectively, by

$$F_i^\mu(x) := \phi_\mu(x_i, G_i(x)) \quad \text{and} \tag{2.13}$$

$$F_i^\mu(x) := \psi_\mu(l_i, u_i, x_i, G_i(x)). \tag{2.14}$$

Note that for the smoother defined by (2.11), $\lim_{a \to \infty} \tilde{\phi}^{BW}(a, b, \mu) = b$ and $\lim_{b \to \infty} \tilde{\phi}^{BW}(a, b, \mu) = a$. Thus, for the MCP case, if $u_i = \infty$ and l_i is finite, then $F_i^\mu(x) = \tilde{\phi}^{BW}(x_i - l_i, G_i(x), \mu)$; if u_i is finite and $l_i = -\infty$, then $F_i^\mu(x) = -\tilde{\phi}^{BW}(u_i - x_i, -G_i(x), \mu)$; and if neither bound is finite, then $F_i^\mu(x) = G_i(x)$.

3. The algorithm. This section summarizes the probability-one homotopy algorithm for solving nonsmooth equations. It contrasts with an earlier hybrid Newton-homotopy method described in [2]. The earlier method begins by using a nonsmooth version of a damped Newton method to solve the root-finding problem $F(x) = 0$. If the Newton algorithm stalls, a standard homotopy method is invoked to solve a particular smoothed version of the original problem, $F^\mu(x) = 0$, where μ is fixed. The smoothing parameter μ is chosen based on the level of a merit function on F at the last point $\hat{x}$ generated by the Newton method. Starting from $\hat{x}$, a homotopy method

is carried out until it produces a point that yields a better merit value than the previous Newton iterate. The Newton method is then started again and the process repeats until a point is produced that is close enough to a solution or the homotopy method fails. One key feature of that hybrid method is that each time the Newton method stalls, a different homotopy map is constructed. The smoothing parameter μ is chosen based on the level of the merit function when the Newton method stalls, so the homotopy that is then used is

$$\rho_a^\mu(\lambda, x) := \lambda F^\mu(x) + (1-\lambda)(x-a).$$

An alternative approach, described here, is to adopt a pure probability-one homotopy algorithm by fixing the homotopy map and tracking a single homotopy zero curve into the Newton domain of convergence around a solution. Essentially, the idea is to use a standard probability-one homotopy algorithm, but with a specially designed "end game" near a solution. The key to this approach is to define a homotopy mapping that couples the smoothing parameter with the homotopy parameter.

3.1. The homotopy map. Given a function F and an associated C^2-smoother F^μ, construct a homotopy mapping with F^μ, where the smoothing parameter μ is a function of the homotopy parameter λ so that $\mu \downarrow 0$ as $\lambda \uparrow 1$. If this homotopy satisfies the conditions in Proposition 2.2, a well behaved path exists from almost any starting point to a solution, and standard curve tracking techniques can reliably solve the equation $F(x) = 0$.

Throughout this section, assume that F is a Lipschitz continuous function on $\mathbb{R}^n$ and that F^μ is a C^2-smoother for F. Take $\mu : [0,1] \to \mathbb{R}_+$ to be a decreasing C^2 function such that $\mu(\lambda) > 0$ for $\lambda < 1$ and $\mu(1) = 0$. For example,

$$\mu(\lambda) := \alpha(1-\lambda) \tag{3.1}$$

for some parameter $\alpha > 0$. Define the homotopy map $\rho_a : [0,1) \times \mathbb{R}^n \to \mathbb{R}^n$, nonlinear in λ, by

$$\rho_a(\lambda, x) := \lambda\, F^{\mu(\lambda)}(x) + (1-\lambda)(x-a), \tag{3.2}$$

and let γ_a be the connected component of the set $\rho_a^{-1}(\{0\})$ that contains $(0, a)$. Notice that this mapping is a generalization of (2.3), since if F is C^2, then $F^\mu := F$ suffices.

In order to ensure that a well behaved zero curve exists, conditions on F and its smoother are required so that Proposition 2.2 can be invoked. The following weak assumption on the smoother will be useful in the theory that follows.

ASSUMPTION 3.1. *There is a nondecreasing function $\eta : \mathbb{R}_+ \to \mathbb{R}_+$ satisfying $\lim_{\nu \downarrow 0} \eta(\nu) = 0$ such that for all x in $\mathbb{R}^n$ and all ν in $\mathbb{R}_+$*

$$\|F^\nu(x) - F(x)\|_\infty \le \eta(\nu).$$

Note (by [2, Proposition 2.14]) that if F^ν is constructed by (2.14), with ϕ_μ defined either by (2.10) or (2.11), then Assumption 3.1 is satisfied with $\eta(\nu) := 3\sqrt{2\nu}$ or $\eta(\nu) := 3\nu$, respectively.

The following theorem [5, Theorem 2.11] is a generalization of Theorem 2.3.

THEOREM 3.2. *Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be a Lipschitz continuous function such that for some fixed $r > 0$ and $\tilde{x} \in \mathbb{R}^n$,*

$$(x - \tilde{x})^T F(x) \ge 0 \text{ whenever } \|x - \tilde{x}\| = r,$$

and let F^μ be a smoother for F satisfying Assumption 3.1. *Further, suppose that the smoothing parameter $\mu(\lambda)$ is such that*

$$\eta(\mu(\lambda)) < \frac{1-\lambda}{\lambda} M \text{ for } 0 < \lambda \leq 1 \tag{3.3}$$

for some $M \in (0, r)$. Then γ_a is bounded for almost every $a \in \mathbb{R}^n$ such that $\|a - \tilde{x}\| < \tilde{r} := r - M$.

A direct application of Proposition 2.2 gives the main convergence theorem.

THEOREM 3.3. *Under the assumptions of Theorem* 3.2, *F has a zero in a closed ball of radius r about $\tilde{x}$, and for almost every a in the interior of a ball of radius $\tilde{r}$ about $\tilde{x}$ there is a zero curve γ_a of*

$$\rho(a, \lambda, x) := \rho_a(\lambda, x) := \lambda F^{\mu(\lambda)}(x) + (1-\lambda)(x-a),$$

along which $\nabla\rho_a(\lambda, x)$ has full rank, emanating from $(0, a)$ and reaching a zero $\bar{x}$ of F at $\lambda = 1$. Further, γ_a has finite arc length if F is strongly regular at $\bar{x}$.

Observe that in applications, the r in Theorem 3.2 can be arbitrarily large; hence so can $\tilde{r} = r - M$, and thus $\|a - \tilde{x}\| < \tilde{r}$ is really no restriction at all.

3.2. Tracking the zero curve. As discussed in section 2.4, the zero curve can, with probability one, be parameterized by arc length: Let $(\lambda(s), x(s))$ be the point on γ_a of arc length s away from $(0, x^a)$. Tracking the zero curve involves generating a sequence of points $\{y^k\} \subset \mathbb{R}^{n+1}$, with $y^0 = (0, x^a)$, that lie approximately on the curve in order of increasing arc length. That is, $y^k \approx (\lambda(s_k), x(s_k))$, where $\{s_k\}$ is some increasing sequence of arc lengths.

The subroutine STEPNX from HOMPACK90 [31] is used to handle the curve tracking. At each iteration, STEPNX uses a predictor-corrector algorithm to generate the next point on the curve. The prediction phase requires for each iterate y^k the corresponding unit tangent vector to the curve, $(y')^k \approx (\lambda'(s_k), x'(s_k))$. This is accomplished by finding an element η of the null space of $\nabla\rho_a(y^k)$ and setting $(y')^k := \pm\eta/\|\eta\|$, where the sign is chosen so that $(y')^k$ makes an acute angle with $(y')^{k-1}$, for $k > 0$. On the first iterate, the sign is chosen so that the first component (corresponding to λ) of $(y')^0$ is positive.

At each iteration after the first, STEPNX approximates the zero curve with a Hermite cubic polynomial $c^k(s)$, which is constructed using the last two points y^{k-1} and y^k, along with the associated unit tangent vectors $(y')^{k-1}$ and $(y')^k$. A step of length h along this cubic yields the predicted point $w^{k,0} := c(s_k + h)$. The first iteration uses a linear predictor instead, which is constructed using the starting point y^0 and its associated unit tangent vector.

Once the predicted point is calculated, a normal flow corrector algorithm [31] is used to return to the zero curve. Starting with the initial point $w^{k,0}$, the corrector iterates $w^{k,j}, j = 1, \ldots$, are calculated via the formula $w^{k,j+1} := w^{k,j} + z^{k,j}$, $j = 0, 1, \ldots$, where the step $z^{k,j}$ is the unique minimum-norm solution to the equation

$$\nabla\rho_a(w^{k,j}) z^{k,j} = -\rho_a(w^{k,j}). \tag{3.4}$$

The corrector algorithm terminates when one of the following conditions is satisfied: the normalized correction step $z^{k,j} / \left(1 + \left\|w^{k,j}\right\|\right)$ is sufficiently small, some maximum number of iterations (usually 4) is exceeded, or a rank-deficient Jacobian matrix is encountered in (3.4). In the first case, set $y^{k+1} := w^{k,j}$, calculate an optimal step size h for the next iteration, and proceed to the next prediction step. In the second

case, discard the point and return to the prediction phase, using a smaller step size if possible; otherwise, terminate curve tracking with an error return. In the third case, terminate the curve tracking, since $\operatorname{rank} \nabla \rho_a < n$ should theoretically not happen and indicates serious difficulty. The step size in h is also never reduced beyond relative machine precision.

3.2.1. Step size control. At each iteration, STEPNX estimates an "optimal" step size to be used in computing the predicted point. This calculation is governed by several user-defined parameters. Successful termination of the corrector phase occurs when the norm of the residual $\left\|\rho(w^{k,j})\right\|$ is sufficiently small. In some cases, this can happen even when the converged point is not close to the true zero curve. As the tracking progresses, the computed points may slowly drift farther and farther from the zero curve, while continuing to meet the criterion on the norm of the residual. Eventually, the iterates may leave the Newton domain of attraction, and the corrector phase may fail to converge, no matter how small the predictor step is. To avoid such difficulties, STEPNX calculates several quantities that measure the "quality" of the step.

The first quantity is the contraction factor

$$\left\|z^{k,1}\right\| / \left\|z^{k,0}\right\|,$$

which measures how much the Newton step shrinks from the first corrector iteration to the second. The second quantity is the residual factor

$$\left\|\rho_a(w^{k,1})\right\| / \left\|\rho_a(w^{k,0})\right\|.$$

The third quantity is the distance factor

$$\left\|w^{k,1} - y^{k+1}\right\| / \left\|w^{k,0} - y^{k+1}\right\|,$$

which approximates how much the distance from the zero curve shrinks from the first iteration to the second. Since Newton's method has quadratic local convergence, each of these quantities should be small when the predicted point is close to the zero curve. Through the use of input parameters, the user is able to specify ideal values (`lideal, rideal, dideal`, respectively) for each of these quantities. If the quantities are smaller than the ideal, the step size will be increased; if the quantities are larger than ideal, the step size will be decreased. The amount of increase or decrease is also controlled by user-defined parameters. Generally, default values for all of these parameters work very well. However, occasionally, it is necessary to choose more conservative parameter values in order to avoid losing the zero curve.

As a final consideration, the default limit on the number of Newton iterations in the corrector phase is 4 (a HOMPACK90 parameter). In some cases, increasing this limit to 6 or 8 improved performance.

3.3. The end game. The standard homotopy method used by HOMPACK90 concludes the curve tracking with an end game strategy that zeros in on a point (λ, x) on the zero curve with $\lambda = 1$. This end game strategy, which is a robust blend of secant iterations with Newton corrections, is begun when a point (λ, x) is found on the zero curve with $\lambda > 1$. However, this approach requires that $\rho(\lambda, x)$ be defined for $\lambda > 1$—a requirement that is not desirable here since the smoother $F^{\mu(\lambda)}$ may not be defined for $\lambda > 1$. Therefore the standard end game is replaced with the generalized Newton method given in Figure 2.1, which is begun while $\lambda < 1$ still.

The Newton end game is invoked when one of the following criteria is satisfied:

1. The point generated by the cubic predictor (with step length h) has $\lambda > 1$.
2. A linear predictor with the same step length has $\lambda > 1$.
3. The corrector phase of the algorithm generates a point with $\lambda > 1$.

In all cases, a starting point for the Newton end game is the prediction of where the zero curve crosses the hyperplane $\lambda = 1$. The precise details follow.

1. First, try to find a point (λ^c, x^c) for which the cubic approximation has $\lambda^c = 1$. If this point occurs within a step length shorter than $2h$, then x^c will be the starting point.
2. Otherwise, find a point (λ^l, x^l) for which the linear approximation has $\lambda^l = 1$. Then x^l will be the starting point.

If the curve tracking fails for any reason before the end game criteria are met, then attempt the nonsmooth Newton method with the starting point x, where (λ, x) is the last point found on the zero curve.

The starting point generated by the above procedure is usually quite good. However, in some cases, the Newton end game may fail to converge. In that event, simply return to tracking the zero curve, picking up from the last point y^k on γ_a, but with the step size (computed by STEPNX) cut in half, and with the STEPNX tracking tolerances `abserr` and `relerr` also reduced.

Note that this approach differs from the end game strategy described in [5], which simply invoked the Newton end game with a starting point x whenever a point (λ, x) was found on the zero curve with λ sufficiently close to 1. The new end game strategy has two main advantages over this earlier approach. First, using the cubic predictor to estimate where the zero curve crosses $\lambda = 1$ results in a significantly more accurate approximation for the solution as a starting point for Newton's method. Second, the new method takes better advantage of available information in determining when to enter the end game. Specifically, on difficult problems, the Newton domain of convergence near the final solution will be small, so it is desirable to track the zero curve very close to $\lambda = 1$ before trying Newton's method. This is exactly what happens since, in this case, the step size will likely be very small. In contrast, for easier problems, larger step sizes will be used, and the end game will be started earlier. Again this is acceptable because the Newton domain of convergence around the solution will likely be large.

In order to solve the system $F(x) = 0$, the nonsmooth Newton method requires that F be semismooth. If, in addition, F is BD-regular at a solution x^*, Newton's method will converge superlinearly in some neighborhood about x^*. Theoretically, to use the homotopy approach and guarantee the end game's success, F should satisfy the global monotonicity property and be strongly regular at every solution. This guarantees that the homotopy's zero curve crosses the hyperplane $\lambda = 1$ transversally rather than tangentially, and ensures that the zero curve will have finite arc length. For most homotopies used in practice in other contexts, even if the zero curve γ_a is tangent to the hyperplane $\lambda = 1$, a point with $\lambda > 1$ near $\rho_a^{-1}(\{0\})$ will be generated, and the usual end game provided in HOMPACK90 will succeed (to modest accuracy, since $\nabla F(\bar{x})$ is singular).

4. Solving mixed complementarity problems. This section specializes the algorithm described above in order to solve mixed complementarity problems. The approach taken here is to reformulate the MCP by defining the function $F : \mathbb{R}^n \to \mathbb{R}^n$ according to (2.8) and (2.9), where ϕ is a positively oriented NCP function, and defining a smoother for F according to (2.12) and (2.14), where ϕ_μ is a smoother

for ϕ. Once these functions are defined, the homotopy algorithm described in the previous section can be used to find a zero of F, which corresponds to a solution of MCP. Because of the special structure of these functions, stronger convergence results are possible than for the general nonsmooth equations problem. In particular, the following results establish that it is possible to construct a homotopy method that follows a strictly feasible path (that is, a path whose x-components remain in the interior of the feasible region). This is unusual for probability-one homotopy methods. In fact, due to the necessity of maintaining transversality, it is usually very difficult to construct probability-one homotopy methods that have feasible paths [25, 29]. Interestingly, probability-one methods usually view it as an advantage that their paths go infeasible, since they "cut across" infeasible regions to get to the feasible solution. However, since many practical problems involve functions that are not defined outside the feasible region, it is important to have feasible algorithms available as well.

The first results presented in this section are tailored to particular choices of ϕ and ϕ_μ, namely the Fischer–Burmeister NCP function (2.6) and the smoother (2.11). More general results are given in Theorem 4.3 and Corollary 4.4. In describing these results it will be useful to refer to the following index set:

$$I_{l,u} = \{ i \mid -\infty < l_i < u_i < \infty \}.$$

That is, $I_{l,u}$ is the set of indices for which both the lower and upper bounds are finite.

THEOREM 4.1. *Let ϕ be the positively oriented NCP function in* (2.6)*, and let $\tilde{\phi}$ be the smoother for ϕ in* (2.11)*. Let ψ be defined by* (2.8) *with associated smoother $\tilde{\psi}$ defined by* (2.12)*. Choose $a \in \operatorname{int} \mathbb{B}_{l,u}$. Let F^μ be defined by* (2.14)*, where $\mu : [0,1] \to \mathbb{R}_+$ is a decreasing C^2 function satisfying $\mu(1) = 0$ and*

$$\mu(\lambda)^2 \le 2\frac{1-\lambda}{\lambda}(u_i - a_i)(u_i - l_i) \quad \text{for all } i \in I_{l,u},\ \lambda \in (0,1]. \tag{4.1}$$

Define $\rho_a : [0,1) \times \mathbb{R}^n \to \mathbb{R}^n$ by (3.2)*, and let γ_a be the connected component of $\rho_a^{-1}(\{0\})$ containing $(0,a)$. Then γ_a is contained in $[0,1) \times (\operatorname{int} \mathbb{B}_{l,u})$.*

Proof. Let $(\hat{\lambda}, \hat{x})$ be an arbitrary point on γ_a. If $\hat{\lambda} = 0$, then $\hat{x} = a \in \operatorname{int} \mathbb{B}_{l,u}$; so assume $0 < \hat{\lambda} < 1$. First suppose that $\hat{x}_i \le l_i$ for some i. Then

$$0 = \rho_i(\hat{\lambda}, \hat{x}) = \hat{\lambda} F_i^{\mu(\hat{\lambda})}(\hat{x}) + (1 - \hat{\lambda})(\hat{x}_i - a_i)$$

or

$$F_i^{\mu(\hat{\lambda})}(\hat{x}) = -\frac{1-\hat{\lambda}}{\hat{\lambda}}(\hat{x}_i - a_i) > 0, \tag{4.2}$$

where the last inequality follows from $\hat{x}_i \le l_i < a_i$, since a is interior to $\mathbb{B}_{l,u}$. Also $F_i^{\mu(\hat{\lambda})}(\hat{x}) = \tilde{\phi}(\hat{x}_i - l_i, \zeta, \mu)$, where $\zeta := -\tilde{\phi}(u_i - \hat{x}_i, -G_i(\hat{x}), \mu)$. Thus

$$F_i^{\mu(\hat{\lambda})}(\hat{x}) = \tilde{\phi}(\hat{x}_i - l_i, \zeta, \mu) \le \phi(\hat{x}_i - l_i, \zeta) \le 0,$$

contradicting (4.2). It follows that every point (λ, x) on γ_a satisfies $l < x$.

Now suppose $\hat{x}_i \ge u_i$ for some i. Note that this implies that u_i is finite. In this case (analogous to (4.2)),

$$F_i^{\mu(\hat{\lambda})}(\hat{x}) = -\frac{1-\hat{\lambda}}{\hat{\lambda}}(\hat{x}_i - a_i) \le -\frac{1-\hat{\lambda}}{\hat{\lambda}}(u_i - a_i) \tag{4.3}$$

and $\zeta = -\tilde{\phi}(u_i - \hat{x}_i, -G_i(\hat{x}), \mu) > 0$ since $\mu(\lambda) > 0$ for $\lambda < 1$. If $l_i = -\infty$, then $F_i^{\mu(\hat{\lambda})}(\hat{x}) = \zeta > 0$, contradicting (4.3). If l_i is finite, then from (6) and (11), for any $\alpha, \beta \in \mathbb{R}$,

$$\tilde{\phi}(\alpha, \beta, \mu) - \phi(\alpha, \beta) > -\frac{\mu^2}{2\sqrt{\alpha^2 + \beta^2}}. \tag{4.4}$$

Then, using $\zeta > 0$, $\hat{x}_i \geq u_i$, the monotonicity of $\tilde{\phi}$, (4.4), and (4.1) give

$$\begin{aligned} F_i^{\mu(\hat{\lambda})}(\hat{x}) &= \tilde{\phi}(\hat{x}_i - l_i, \zeta, \mu) \\ &\geq \tilde{\phi}(u_i - l_i, 0, \mu) \\ &> \phi(u_i - l_i, 0) - \frac{\mu^2}{2\sqrt{(u_i - l_i)^2}} \\ &= -\frac{\mu^2}{2(u_i - l_i)} \\ &\geq -\frac{1 - \hat{\lambda}}{\hat{\lambda}}(u_i - a_i), \end{aligned}$$

contradicting (4.3). Therefore every point $(\hat{\lambda}, \hat{x})$ on γ_a satisfies $l < \hat{x} < u$. ∎

Note that if $I_{l,u}$ is empty, then the condition on $\mu(\lambda)$ in the above theorem is achieved by any decreasing C^2 function satisfying $\mu(1) = 0$. If $I_{l,u}$ is not empty, the condition is easily achieved by choosing a deep in the interior of the feasible region $\mathbb{B}_{l,u}$. For example, if $u_i - a_i \geq \frac{1}{2}(u_i - l_i)$ for all $i \in I_{l,u}$, then

$$\mu(\lambda) = \left[\min_{i \in I_{l,u}} (u_i - l_i)\right](1 - \lambda)$$

suffices, since, for $0 < \lambda \leq 1$,

$$\begin{aligned} \mu(\lambda)^2 &= \left[\min_{i \in I_{l,u}} (u_i - l_i)\right]^2 (1 - \lambda)^2 \\ &\leq 2\left[\min_{i \in I_{l,u}} (u_i - a_i)(u_i - l_i)\right](1 - \lambda)^2 \\ &\leq 2\left[\min_{i \in I_{l,u}} (u_i - a_i)(u_i - l_i)\right]\frac{(1 - \lambda)}{\lambda}. \end{aligned}$$

The above theorem has two important consequences. First, as mentioned earlier, because γ_a always stays in the feasible region, it is possible to implement the algorithm without ever having to evaluate functions outside of the feasible region. The second consequence is the guarantee that when all bounds are finite, the zero curve γ_a is bounded. The implications of this are stated in the following corollary.

COROLLARY 4.2. *Let ϕ and ϕ_μ be defined by* (2.6) *and* (2.11)*, respectively. Assume that all the bounds of the MCP are finite, choose $\kappa \in (0, \sqrt{2})$, and take*

$$\mu(\lambda) = \kappa\left[\min_i (u_i - l_i)\right](1 - \lambda). \tag{4.5}$$

Then for almost all $a \in \operatorname{int} \mathbb{B}_{l,u}$ satisfying $u_i - a_i \geq \kappa^2(u_i - l_i)/2$ for $1 \leq i \leq n$ and ρ_a defined as in Theorem 4.1*, there is a zero curve γ_a of ρ_a emanating from $(0, a)$,*

along which $\nabla\rho_a(\lambda, x)$ *has full rank, that remains in* $[0,1)\times\left(\operatorname{int}\mathbb{B}_{l,u}\right)$ *and reaches a point* $(1,\bar{x})$*, where* $\bar{x}$ *solves the MCP.* γ_a *does not intersect itself, is disjoint from any other zeros of* ρ_a*, and has finite arc length if* F *is strongly regular at* $\bar{x}$*.*

Proof. The first four hypotheses of Proposition 2.2 are satisfied trivially. The choice of ϕ_μ, $\mu(\lambda)$, and the restrictions on a are sufficient for carrying out the proof of Theorem 4.1. Hence γ_a remains in $[0,1)\times\left(\operatorname{int}\mathbb{B}_{l,u}\right)$ and is bounded, since $\mathbb{B}_{l,u}$ is bounded. □

The remainder of this section generalizes the above results to other choices of ϕ and ϕ_μ.

THEOREM 4.3. *Let* ϕ *be a positively oriented NCP function, and let* $\tilde{\phi}$ *be a* C^2*-smoother for* ϕ*, monotone in its first two variables, satisfying*

$$\phi(\alpha,\beta)\geq\tilde{\phi}(\alpha,\beta,\mu)\quad\text{for all }\alpha,\beta\in\overline{\mathbb{R}},\ \mu>0,\ \text{and} \tag{4.6}$$

$$\tilde{\phi}(\alpha,0,\mu)>-\frac{c\mu^p}{\alpha}\qquad\text{for }\mu>0,\ 0<\alpha<\infty, \tag{4.7}$$

where c *and* p *are positive constants. Define* ψ *by* (2.8) *and the smoother* $\tilde{\psi}$ *by* (2.12)*. Choose* $a\in\operatorname{int}\mathbb{B}_{l,u}$*, and let* $\mu:[0,1]\to\mathbb{R}_+$ *be a decreasing* C^2 *function with* $\mu(1)=0$ *satisfying*

$$\mu(\lambda)^p\leq\frac{1-\lambda}{c\lambda}(u_i-a_i)(u_i-l_i)\quad\text{for } i\in I_{l,u},\ \lambda\in(0,1]. \tag{4.8}$$

Define F^μ *by* (2.14)*, define* $\rho_a:[0,1)\times\mathbb{R}^n\to\mathbb{R}^n$ *by* (3.2)*, and let* γ_a *be the connected component of* $\rho_a^{-1}(\{0\})$ *containing* $(0,a)$*. Then* γ_a *is contained in* $[0,1)\times(\operatorname{int}\mathbb{B}_{l,u})$*.*

Proof. The proof is identical to the proof of Theorem 4.1 except that in place of (4.4), the inequality (4.7) is used. Then by similar arguments using (4.8),

$$\begin{aligned}F_i^{\mu(\hat{\lambda})}(\hat{x})&>-\frac{c\mu^p}{u_i-l_i}\\&\geq-\frac{1-\hat{\lambda}}{\hat{\lambda}}(u_i-a_i),\end{aligned}$$

contradicting (4.3). □

COROLLARY 4.4. *Let* ϕ*,* $\tilde{\phi}$*,* ψ*,* $\tilde{\psi}$*, and* F^μ *be defined as in Theorem* 4.3*. Assume that all the bounds of the MCP are finite, choose* $\kappa\in(0,1)$*, and take*

$$\mu(\lambda)=\kappa\left(\frac{1-\lambda}{c}\right)^{1/p}\left[\min_i(u_i-l_i)\right]^{2/p}.$$

Then for almost all $a\in\operatorname{int}\mathbb{B}_{l,u}$ *satisfying* $u_i-a_i\geq\kappa^p(u_i-l_i)$ *for* $1\leq i\leq n$ *and* ρ_a *defined as in Theorem* 4.1*, there is a zero curve* γ_a *of* ρ_a *emanating from* $(0,a)$*, along which* $\nabla\rho_a(\lambda,x)$ *has full rank, that remains in* $[0,1)\times\left(\operatorname{int}\mathbb{B}_{l,u}\right)$ *and reaches a point* $(1,\bar{x})$*, where* $\bar{x}$ *solves the MCP.* γ_a *does not intersect itself, is disjoint from any other zeros of* ρ_a*, and has finite arc length if* F *is strongly regular at* $\bar{x}$*.*

4.1. Ensuring feasibility. Since some MCP applications involve functions that are not defined outside the feasible region, the algorithm includes an option to ensure that all iterates are feasible. The following discussion assumes that the MCP algorithm is based on the particular choices of ϕ and ϕ_μ given by (2.6) and (2.11).

Feasibility of the path γ_a can be assured by Theorem 4.1, provided that the initial point a and the function $\mu(\lambda)$ are chosen appropriately. The following procedure achieves this while choosing the initial point a near the starting point x^0 provided by the user: Define a by

$$a_i := \begin{cases} \operatorname{mid}(l_i + \nu_i, x_i^0, u_i - \nu_i) & \text{if } i \in I_{l,u}, \\ \max(l_i + \nu, x_i^0) & \text{for } u_i = \infty, l_i \text{ finite}, \\ \min(u_i - \nu, x_i^0) & \text{for } l_i = -\infty, u_i \text{ finite}, \\ x_i^0 & \text{if } l_i = -\infty, u_i = \infty, \end{cases} \tag{4.9}$$

where $\nu_i := \kappa_{\min}^2(u_i - l_i)/2$ for $i \in I_{l,u}$, and $\kappa_{\min} \in (0,1)$ and $\nu > 0$ are constants that ensure the strict feasibility of a. Next, define $\mu(\lambda)$ by (3.1), with α given by

$$\alpha = \begin{cases} \min\left(c, \kappa\left[\min_{i \in I_{l,u}}(u_i - l_i)\right]\right) & \text{if } I_{l,u} \neq \emptyset, \\ c & \text{otherwise}, \end{cases} \tag{4.10}$$

where c is some positive constant, and κ is defined as follows if $I_{l,u} \neq \emptyset$:

$$\kappa := \min_{i \in I_{l,u}} \sqrt{\frac{2(u_i - a_i)}{u_i - l_i}}. \tag{4.11}$$

Note that if $I_{l,u}$ is not empty, this choice of a and κ ensures that $\kappa \geq \kappa_{\min}$ and also that (4.1) is satisfied. Thus, the assumptions of Theorem 4.1 are satisfied, so γ_a remains strictly within the feasible region. Feasibility is maintained by exploiting STEPNX's built-in logic for handling domain violations. Precisely, whenever a STEPNX call to evaluate $F(x)$ produces an infeasible point (either in the prediction phase or the correction phase), that domain violation is reported to STEPNX. The result is that STEPNX cuts the step size in half (after sanity checks to prevent an infinite loop) and calculates a new predicted point. Since the zero curve is strictly feasible for $\lambda < 1$, eventually (assuming adequate machine precision) a feasible step will be taken.

Finally, to ensure feasibility of the iterates generated in the end game, the generalized damped Newton method in Figure 2.1 is modified according to the general descent framework described in [13]. Specifically, the Newton direction d^k is projected back onto the feasible region to produce the modified direction

$$\tilde{d}^k := \pi_{\mathbb{B}_{l,u}}(x^k + d^k) - x^k.$$

Note that $x^k + \tilde{d}^k$ is feasible. Step 3 in Figure 2.1 is then replaced with the following.

Step 3′ If $\theta(x^k + \tilde{d}^k) \leq (1 - \sigma)\theta(x^k)$, set $x^{k+1} = x^k + \tilde{d}^k$. Otherwise, take a projected gradient step as follows: Let m_k be the smallest nonnegative integer $m \leq m_{max}$ such that

$$\theta(x^k(\alpha^m)) \leq \theta(x^k) - \sigma \nabla\theta(x^k)(x^k - x^k(\alpha^m)), \tag{4.12}$$

where $x^k(t) := \pi_{\mathbb{B}_{l,u}}(x^k - t\nabla\theta(x^k))$. If no such m_k exists, stop; the algorithm failed. Otherwise, set $x^{k+1} = x^k(\alpha^k)$.

Note that for any feasible x^*, $\|x^k + \tilde{d}^k - x^*\| \leq \|x^k + d^k - x^*\|$. This ensures, by [13, Theorem 4.5] and Theorem 2.1, that in a neighborhood of a strongly regular solution $\bar{x}$, the iterates generated by the feasible end game strategy described above converge Q-superlinearly to $\bar{x}$.

The projected gradient step in the above algorithm requires that θ be continuously differentiable. This is true when ϕ is the Fischer–Burmeister function (2.6), but is not true in general.

5. Solver implementation and testing. The MCP algorithm described in the previous section was implemented using the Fischer–Burmeister NCP function for ϕ and the smoother defined by (2.11). The nonsmooth Newton method described in Figure 2.1 was used for the Newton end game. To construct the homotopy mapping defined in (3.2), the parameter a was constructed according to (4.9), with $\kappa_{\min} := 0.1$, and $\nu = 0.0001$. The function $\mu(\lambda)$ was defined by (4.10), with $c = 1.0$ and κ defined by (4.11).

The algorithm was implemented in C with a link to the Fortran 90 subroutine STEPNX from HOMPACK90. The code is interfaced with the GAMS modeling language, enabling it to be tested using the MCPLIB suite of GAMS test problems [11], [4]. All linear algebra was performed using the LUSOL sparse factorization routine [15] from MINOS [18].

Computational results on the MCPLIB problems are shown in Table 5.1. Many of the problems in this test library include multiple runs, which vary the starting point x^0 or other parameters defining the problem. All of the problems were run using default parameter settings, and the number of successes and failures over all runs are reported in the third column of Table 5.1. The notation $m(n)$ means that the problem included $m+n$ runs, and for those there were m successes and n failures. The default parameters were chosen as follows:

- Curve tracking parameters: `abserr` = `relerr` = 10^{-4}. Maximum step size $h_{\max} = 100,000$. The normal default for this parameter used by HOMPACK90 is $h_{\max} = 1$. However, many problems in the MCPLIB test library were poorly scaled, and so had very long zero curves. The large value of $h_{\max}$ was therefore used to allow these curves to be tracked in a reasonable number of iterations. All other curve tracking parameters were the defaults chosen by STEPNX.
- Newton parameters (See Figure 2.1): $\alpha = \sigma = 0.5$, $m_{\max} = 20$. Maximum number of Newton iterations = 30.
- Stopping criteria: An iterate x^k was considered to solve the problem when $\|F(x^k)\|_\infty \,/\, \left(1 + \|x^k\|_\infty\right) < 10^{-6}$.

In cases where the problem was not solved by the default parameters, the algorithm was restarted using more conservative parameters: `abserr` = `relerr` = 10^{-6}, `dideal` = 0.01, `lideal` = 0.01, `rideal` = 0.005, and $h_{\max} = \max(.1, \texttt{arclen}/100)$, where `arclen` is the arc length of the zero curve calculated using the default parameters. Results from these runs are shown in the fourth column of Table 5.1.

For the problems that were not solved by the conservative settings, the last column of Table 5.1 describes the reason for failure. The notation "∞" indicates that the zero curve appeared to go off to infinity. This behavior is common for problems that do not satisfy the global monotonicity assumption. The notation "lost" indicates that STEPNX was unable to continue tracking the zero curve. This is generally due to a poorly conditioned Jacobian matrix. The notation "r" indicates failure due to exceeding resource limits—either the limit of 5000 homotopy steps or 1000 CPU seconds. Finally, the notation "v" indicates failure due to domain violations.

While the untuned algorithm with default parameters failed to solve a number of problems that have been solved by other algorithms, it is encouraging to note that it performed very well on some problems that are generally regarded as very hard. Notable among these are the billups, pgvon105, pgvon106, and simple-ex problems. Thus, the homotopy algorithm should be viewed as an important supplement to other approaches.

TABLE 5.1
MCPLIB test problems.

Problem name	Size	Default settings Success(Failure)	Conservative settings Success(Failure)	Notes
badfree	5	1(0)		
bert_oc	5000	3(1)	3(1)	r
bertsekas	15	5(1)	6(0)	
billups	1	3(0)		
bratu	5625	1(0)		
choi	13	1(0)		
colvdual	20	4(0)		
colvnlp	15	6(0)		
colvtemp	20	4(0)		
cycle	1	1(0)		
degen	2	1(0)		
duopoly	63	0(1)	0(1)	∞
ehl_k40	41	2(1)	3(0)	
ehl_k60	61	2(1)	3(0)	
ehl_k80	81	2(1)	3(0)	
ehl_kost	101	1(2)	1(2)	lost
electric	158	0(1)	0(1)	∞
eta2100	296	0(1)	1(0)	
explcp	16	1(0)		
forcebsm	184	0(1)	0(1)	∞
forcedsa	186	0(1)	0(1)	∞
freebert	15	7(0)		
gafni	5	3(0)		
games	16	25(0)		
hanskoop	14	10(0)		
hydroc06	29	0(1)	0(1)	∞
hydroc20	99	0(1)	0(1)	∞
jel	6	2(0)		
josephy	4	8(0)		
kojshin	4	8(0)		
lincont	419	0(1)	0(1)	∞
mathinum	3	6(0)		
mathisum	4	7(0)		
methan08	31	0(1)	0(1)	∞
multi-v	48	0(3)	0(3)	lost
nash	10	4(0)		
ne-hard	3	1(0)		
obstacle	2500	7(1)	8(0)	

It should also be noted that the algorithm solved several problems for which it was not able to track the zero curve all the way to $\lambda = 1$. This occurred for the bert_oc, obstacle, and opt_cont* problems. However, for these problems the Newton end game was able to find the solution.

Except for the cases "v" and "r", the failures are of two types: numerical instability or unbounded homotopy zero curve γ_a. No attempt was made to scale, reformulate, or precondition the test problems, or to tune the tracking parameters for a particular problem. There is little doubt that a concerted pursuit of all of these options would have removed all of the failures due to numerical instability. For instance, failures denoted "lost" are cured by tracking with smaller error tolerances and permitted steps. The "r" failures are removed by scaling or accepting the CPU time required for long paths. "v" failures are cured by never permitting STEPNX to generate infeasible points. These case-by-case "fixes" were intentionally not done to illustrate the homotopy performance with fixed settings over a large class of problems.

TABLE 5.1
MCPLIB test problems (cont.).

Problem name	Size	Default settings Success(Failure)	Conservative settings Success(Failure)	Notes
olg	249	0(1)	0(1)	lost
opt_cont127	4096	1(0)		
opt_cont	288	1(0)		
opt_cont255	8192	1(0)		
opt_cont31	1024	1(0)		
opt_cont511	16384	1(0)		
pgvon105	105	4(0)		
pgvon106	106	5(1)	6(0)	
pies	42	0(1)	1(0)	
powell	16	5(1)	5(1)	∞
powell_mcp	8	6(0)		
qp	4	1(0)		
romer	214	0(2)	0(2)	lost
scarbsum	40	1(1)	2(0)	
scarfanum	13	4(0)		
scarfasum	14	1(3)	1(3)	v
scarfbnum	39	0(2)	2(0)	
scarfbsum	40	1(1)	2(0)	
shubik	30	7(41)	13(35)	r
simple-ex	17	1(0)		
simple-red	13	1(0)		
sppe	27	3(0)		
tinloi	146	10(54)	64(0)	
tobin	42	4(0)		
trade12	600	1(1)	1(1)	lost
trafelas	2376	0(2)	0(2)	r

The unbounded zero curves are a more fundamental problem, indicating that the default homotopy map (3.2) is inadequate (which is no surprise, since in engineering practice the default map is virtually never used). It is likely that replacing (3.2) by $\lambda F^{\mu(\lambda)}(x) + (1-\lambda)G(a,\lambda,x)$, where G is carefully crafted for each problem, could remove the other failures. This remains a topic for future work.

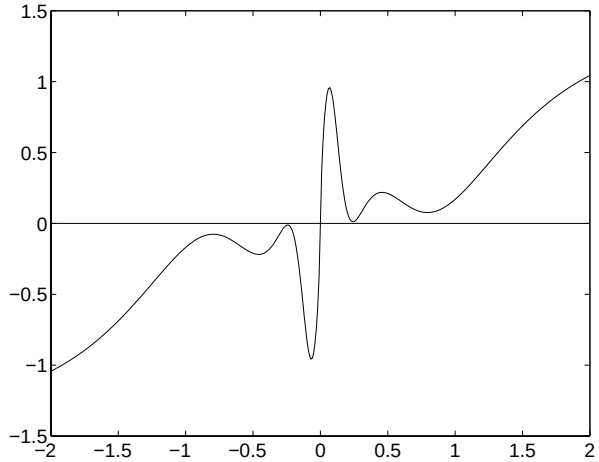


FIG. 5.1. $f(x) := \arctan(100x)/\pi + \sin(5x/(x^2+0.2))/2 + 0.1x$.

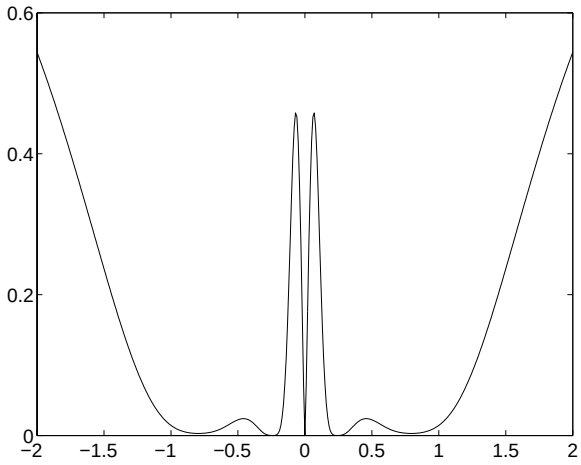


FIG. 5.2. *Merit function* $\theta(x) := 0.5f(x)^2$.

Finally, to emphasize the robustness of the homotopy method, consider the one-dimensional equation $f(x) = 0$, where

$$f(x) := \arctan(100x)/\pi + \sin(5x/(x^2 + 0.2))/2 + 0.1x.$$

This function, shown in Figure 5.1, has a unique root at $x = 0$. For algorithms that rely on descent of a merit function, this root is difficult to find because, as illustrated in Figure 5.2, the global minimum of the merit function $\theta(x) := f(x)^2/2$ is in a very narrow valley. Nevertheless, the proposed probability-one homotopy algorithm easily found the root, tracking the homotopy zero curve in 32 steps from a starting point of $x^0 = 0.5$. As a comparison, PATH version 4.0 [12] was used from the same starting point. After 449 iterations, PATH terminated at $x = .24233$, corresponding to a local minimum of θ. This function $\theta(x)$, while artificial, is representative of merit functions encountered in applications such as protein folding, analog circuit simulation, and aircraft configuration design.

6. Conclusions. This paper describes a probability-one homotopy algorithm for solving nonsmooth systems of equations and complementarity problems. These methods are an extension to nonsmooth equations of the probability-one homotopy methods described in [8], [27], [30], [31], and they are attractive because they are able to solve a qualitatively different class of problems than methods relying on merit functions. This claim is justified both theoretically and computationally. The key to success of the method is the global monotonicity assumption. When this is satisfied, the zero curve is known to lead to a solution. This result is formalized in Theorem 3.3. In the case of complementarity problems, an easily satisfiable condition was established, which ensures that the homotopy zero curve always remains strictly feasible. This condition can always be enforced in the algorithm by choosing the initial point a properly. A simple consequence of this result is that, for finitely bounded mixed complementarity problems, the zero curve is bounded and, by Proposition 2.2, is guaranteed to lead to a solution.

Topics for future research include the effect of the choice of the smoothing function ψ_μ used to define F^μ, and the choice of the start function G, in the general homotopy map $\lambda F^{\mu(\lambda)}(x) + (1 - \lambda)G(a, \lambda, x)$. A systematic numerical comparison,

for several different smoothing functions, of smoothing Newton methods, piecewise smooth continuation, and the present probability-one homotopy algorithm also seems worthwhile.

REFERENCES

[1] S. C. Billups, *Algorithms for Complementarity Problems and Generalized Equations*, Ph.D. thesis, University of Wisconsin–Madison, Madison, WI, 1995.

[2] S. C. Billups, *A homotopy based algorithm for mixed complementarity problems*, SIAM J. Optim., 12 (2002), pp. 583–605.

[3] S. C. Billups, *Improving the robustness of descent-based methods for semi-smooth equations using proximal perturbations*, Math. Program., 87 (2000), pp. 153–176.

[4] S. C. Billups, S. P. Dirkse, and M. C. Ferris, *A comparison of large scale mixed complementarity problem solvers*, Comput. Optim. Appl., 7 (1997), pp. 3–25.

[5] S. C. Billups, A. L. Speight, and L. T. Watson, *Nonmonotone path following methods for nonsmooth equations and complementarity problems*, in Complementarity: Applications, Algorithms and Extensions, M. C. Ferris, O. L. Mangasarian, and J.-S. Pang, eds., Kluwer Academic Publishers, Norwell, MA, 2001, pp. 19–41.

[6] C. Buck, *Advanced Calculus*, 3rd ed., McGraw–Hill, New York, NY, 1978.

[7] B. Chen and X. Chen, *A global and local superlinear continuation-smoothing method for $P_0 + R_0$ NCP or monotone NCP*, SIAM J. Optim., 9 (1999), pp. 624–645.

[8] S.-N. Chow, J. Mallet-Paret, and J. A. Yorke, *Finding zeros of maps: Homotopy methods that are constructive with probability one*, Math. Comp., 32 (1978), pp. 887–899.

[9] T. De Luca, F. Facchinei, and C. Kanzow, *A semismooth equation approach to the solution of nonlinear complementarity problems*, Math. Programming, 75 (1996), pp. 407–439.

[10] T. De Luca, F. Facchinei, and C. Kanzow, *A theoretical and numerical comparison of some semismooth algorithms for complementarity problems*, Comput. Optim. Appl., 16 (2000), pp. 173–205.

[11] S. P. Dirkse and M. C. Ferris, *MCPLIB: A collection of nonlinear mixed complementarity problems*, Optim. Methods Softw., 5 (1995), pp. 319–345.

[12] S. P. Dirkse and M. C. Ferris, *The PATH solver: A non-monotone stabilization scheme for mixed complementarity problems*, Optim. Methods Softw., 5 (1995), pp. 123–156.

[13] M. C. Ferris, C. Kanzow, and T. S. Munson, *Feasible descent algorithms for mixed complementarity problems*, Math. Program., 86 (1999), pp. 475–497.

[14] A. Fischer, *A special Newton–type optimization method*, Optimization, 24 (1992), pp. 269–284.

[15] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *Maintaining LU factors of a general sparse matrix*, Linear Algebra Appl., 88/89 (1987), pp. 239–270.

[16] C. Kanzow, *Some equation–based methods for the nonlinear complementarity problem*, Optim. Methods Softw., 3 (1994), pp. 327–340.

[17] O. L. Mangasarian, *Equivalence of the complementarity problem to a system of nonlinear equations*, SIAM J. Appl. Math., 31 (1976), pp. 89–92.

[18] B. A. Murtagh and M. A. Saunders, *MINOS* 5.0 *User's Guide*, Technical report SOL 83.20, Stanford University, Stanford, CA, 1983.

[19] L. Qi, *Convergence analysis of some algorithms for solving nonsmooth equations*, Math. Oper. Res., 18 (1993), pp. 227–244.

[20] L. Qi, *Regular pseudo-smooth NCP and BVIP functions and globally and quadratically convergent generalized Newton methods for complementarity and variational inequality problems*, Math. Oper. Res., 24 (1999), pp. 440–471.

[21] L. Qi, D. Sun, and G. Zhou, *A new look at smoothing Newton methods for nonlinear complementarity problems and box constrained variational inequalities*, Math. Program., 87 (2000), pp. 1–35.

[22] L. Qi and J. Sun, *A nonsmooth version of Newton's method*, Math. Programming, 58 (1993), pp. 353–368.

[23] H. Sellami, *A Continuation Method for Normal Maps*, Ph.D. thesis, University of Wisconsin–Madison, Madison, WI, 1994.

[24] H. Sellami and S. M. Robinson, *Implementation of a continuation method for normal maps*, Math. Programming, 76 (1997), pp. 563–578.

[25] G. Vasudevan, L. T. Watson, and F. H. Lutze, *Homotopy approach for solving constrained optimization problems*, IEEE Trans. Automat. Control, 36 (1991), pp. 494–498.

[26] L. T. Watson, *An algorithm that is globally convergent with probability one for a class of nonlinear two-point boundary value problems*, SIAM J. Numer. Anal., 16 (1979), pp. 394–401.

[27] L. T. Watson, *A globally convergent algorithm for computing fixed points of C^2 maps*, Appl. Math. Comput., 5 (1979), pp. 297–311.
[28] L. T. Watson, *Solving the nonlinear complementarity problem by a homotopy method*, SIAM J. Control Optim., 17 (1979), pp. 36–46.
[29] L. T. Watson, *Theory of globally convergent probability-one homotopies for nonlinear programming*, SIAM J. Optim., 11 (2000), pp. 761–780.
[30] L. T. Watson, S. C. Billups, and A. P. Morgan, *Algorithm* 652: *HOMPACK: A suite of codes for globally convergent homotopy algorithms*, ACM Trans. Math. Software, 13 (1987), pp. 281–310.
[31] L. T. Watson, M. Sosonkina, R. C. Melville, A. P. Morgan, and H. F. Walker, *Algorithm* 777: *HOMPACK*90*: A suite of FORTRAN* 90 *codes for globally convergent homotopy algorithms*, ACM Trans. Math. Software, 23 (1997), pp. 514–549.

SIAM J. OPTIM.
Vol. 12, No. 3, pp. 627–655

A NEW CONDITION MEASURE, PRECONDITIONERS, AND RELATIONS BETWEEN DIFFERENT MEASURES OF CONDITIONING FOR CONIC LINEAR SYSTEMS*

MARINA EPELMAN† AND ROBERT M. FREUND‡

Abstract. In recent years, a body of research into "condition numbers" for convex optimization has been developed, aimed at capturing the intuitive notion of problem behavior. This research has been shown to be relevant in studying the efficiency of algorithms (including interior-point algorithms) for convex optimization as well as other behavioral characteristics of these problems such as problem geometry, deformation under data perturbation, etc. This paper studies measures of conditioning for a conic linear system of the form (FP_d): $Ax = b,\ x \in C_X$, whose data is $d = (A, b)$. We present a new measure of conditioning, denoted μ_d, and we show implications of μ_d for problem geometry and algorithm complexity and demonstrate that the value of $\mu = \mu_d$ is independent of the specific data representation of (FP_d). We then prove certain relations among a variety of condition measures for (FP_d), including μ_d, σ_d, $\overline{\chi}_d$, and $\mathcal{C}(d)$. We discuss some drawbacks of using the condition number $\mathcal{C}(d)$ as the sole measure of conditioning of a conic linear system, and we introduce the notion of a "preconditioner" for (FP_d), which results in an equivalent formulation ($\mathrm{FP}_{\tilde{d}}$) of (FP_d) with a better condition number $\mathcal{C}(\tilde{d})$. We characterize the best such preconditioner and provide an algorithm and complexity analysis for constructing an equivalent data instance $\tilde{d}$ whose condition number $\mathcal{C}(\tilde{d})$ is within a known factor of the best possible.

Key words. complexity of convex programming, conditioning, preconditioners

AMS subject classifications. 90C, 90C05, 90C25, 90C60

PII. S1052623400373829

1. Introduction.

The subject of this paper is the further study and development of a new measure of conditioning for the convex feasibility problem in conic linear form:

$$(\mathrm{FP}_d) : Ax = b,\ x \in C_X, \tag{1}$$

where $A \in \mathcal{L}(X, Y)$ is a linear operator between n- and m-dimensional spaces X and Y, $b \in Y$, and $C_X \subset X$ is a closed convex cone, $C_X \neq X$. We denote the data for the problem (FP_d) by $d = (A, b)$ (the cone C_X is regarded as fixed and given) and the set of solutions of (FP_d) by

$$X_d \stackrel{\triangle}{=} \{x \in X : Ax = b,\ x \in C_X\}.$$

The problem (FP_d) is an important tool in mathematical programming. It provides a very general format for studying the feasible regions of convex optimization problems (in fact, any convex feasibility problem can be modeled as a conic linear system) and includes linear programming and semidefinite programming feasibility problems as special cases. Over the last decade many important developments in linear programming, most notably the theory of interior-point methods, have been extended

*Received by the editors June 14, 2000; accepted for publication (in revised form) July 2, 2001; published electronically January 9, 2002.
http://www.siam.org/journals/siopt/12-3/37382.html

†University of Michigan, Industrial and Operations Engineering, 1205 Beal Avenue, Ann Arbor, MI 48109-2117 (mepelman@umich.edu). This author's research has been partially supported through an NSF Graduate Research Fellowship.

‡M.I.T. Sloan School of Management, 50 Memorial Drive, Cambridge, MA 02142-1347 (rfreund @mit.edu).

to convex problems in this form. In recent years, largely prompted by these developments, researchers have developed new and powerful theories of condition numbers for convex optimization, aimed at capturing the intuitive notion of problem behavior; this body of research has been shown to be important in studying the efficiency of algorithms, including interior-point algorithms, for convex optimization as well as other behavioral characteristics of these problems such as problem geometry, deformation under data perturbation, etc.

In this paper, we (i) develop a new measure of conditioning μ_d for (FP$_d$) that is invariant under equivalent data representations of the problem, (ii) establish the connection of the condition numbers μ_d and $\mathcal{C}(d)$ to some of the measures of conditioning arising in recent linear programming literature, and (iii) develop a theory of "preconditioners" for improving the condition number of (FP$_d$). We begin by briefly reviewing developments in the theory of measures of conditioning in recent literature as well as by providing an overview of the issues addressed in this paper.

The study of the computational complexity of linear programming originated with the analysis of the simplex algorithm, which, while extremely efficient in practice, was shown by Klee and Minty [15] to have worst-case complexity exponential in the number of variables. Khachiyan [14] demonstrated that linear programming problems were in fact polynomially solvable via the ellipsoid algorithm. Under the assumption that the problem data is rational, the ellipsoid algorithm requires at most $O(n^2L)$ iterations, where n is the number of variables and L is the *problem size*, which is roughly equal to the number of bits required to represent the problem data. The development of interior-point methods gave rise to algorithms that are efficient in theory as well as in practice (unlike the ellipsoid algorithm). The first such algorithm, developed by Karmarkar [13], has a complexity bound of $O(nL)$ iterations, and the algorithm introduced by Renegar [23] has a complexity bound of $O(\sqrt{n}L)$ iterations, which is currently the best known bound for linear programming. Many interior-point algorithms have also proven to be extremely efficient computationally and are often superior to the simplex algorithm.

Despite the importance of the above results, there are several serious drawbacks in analyzing algorithm performance in the bit-complexity framework. One such drawback is the fact that computers use floating point arithmetic, rather than integer arithmetic, in performing computations. As a result, two problems can have data that are extremely close but have drastically different values of L. The analysis of the performance of algorithms for solving these problems will yield different performance estimates, yet actual performance of the algorithms will likely be similar due to their similar numerical properties. See Wright [39] for a detailed discussion. A second drawback is that the complexity analysis of linear programming algorithms in terms of L largely relies on the combinatorial structure of the linear program; in particular, it relies on the fact that the set of feasible solutions is a polyhedron and the solution is attained at one of the extreme points of this polyhedron.

A relevant way to measure the intuitive notion of conditioning of a convex optimization (or feasibility) problem via the so-called *distance to ill-posedness* and the closely related *condition number* was developed by Renegar in [24] in a more specific setting, but then generalized more fully in [25] and in [26] to convex optimization and feasibility problems in conic linear form. Recall that $d = (A, b)$ is the data for the problem (FP$_d$) of (1). The *condition number* $\mathcal{C}(d)$ of (FP$_d$) is essentially a scale-invariant reciprocal of the smallest data perturbation $\Delta d = (\Delta A, \Delta b)$ for which the system (FP$_{d+\Delta d}$) changes its feasibility status. The problem (FP$_d$) is well-conditioned

to the extent that $\mathcal{C}(d)$ is small; when the problem (FP$_d$) is "ill-posed" (i.e., arbitrarily small perturbations of the data can yield both feasible and infeasible problem instances), then $\mathcal{C}(d) = +\infty$.

One of the important issues addressed by researchers is the relationship between the condition number $\mathcal{C}(d)$ and the geometry of the feasible region of (FP$_d$). Renegar [24] demonstrated that when a feasible instance of (FP$_d$) is well-posed ($\mathcal{C}(d) < \infty$), there exists a point x feasible for (FP$_d$) which satisfies $\|x\| \leq \mathcal{C}(d)$. Furthermore, it is shown in [8] that under the above assumption the set of feasible solutions contains a so-called "reliable" solution: A solution $\hat{x}$ of (FP$_d$) is reliable if, roughly speaking, (i) the distance from $\hat{x}$ to the boundary of the cone C_X, $\operatorname{dist}(\hat{x}, \partial C_X)$, is not excessively small; (ii) the norm of the solution $\|\hat{x}\|$ is not excessively large; and (iii) the ratio $\frac{\|\hat{x}\|}{\operatorname{dist}(\hat{x}, \partial C_X)}$ is not excessively large. The importance of reliable solutions is motivated in part by considerations of finite-precision computations. The results in [8] also demonstrate that when the system (FP$_d$) is feasible, there exists a feasible point $\hat{x}$ such that

$$\frac{\|\hat{x}\|}{\operatorname{dist}(\hat{x}, \partial C_X)} \leq c_1 \mathcal{C}(d), \quad \operatorname{dist}(\hat{x}, \partial C_X) \geq c_2 \frac{1}{\mathcal{C}(d)}, \quad \|\hat{x}\| \leq c_3 \mathcal{C}(d), \tag{2}$$

where the constants c_1, c_2, and c_3 depend only on the "width" of the cone C_X (to be formally defined shortly) and are independent of the data d of the problem (FP$_d$) (but may depend on n).

The condition number $\mathcal{C}(d)$ was also shown to be crucial for analyzing the complexity of algorithms for solving (FP$_d$). Renegar [26] presented an interior-point algorithm for solving (FP$_d$) with the complexity bound of $O(\sqrt{\vartheta} \ln(\vartheta\, \mathcal{C}(d)))$ iterations, where ϑ is the complexity parameter of a self-concordant barrier for the cone C_X. In [9] it was shown that a suitably modified version of the ellipsoid algorithm will solve (FP$_d$) in $O(n^2 \ln(\mathcal{C}(d)))$ iterations. (The constants in both complexity bounds depend on the width of C_X.) In [4], a generalization of a row-action algorithm is shown to compute a reliable solution of (FP$_d$) in the sense of (2). The complexity of this algorithm is also closely tied to $\mathcal{C}(d)$.

The recent literature has explored many other important properties of the problem (FP$_d$) tied to the distance to ill-posedness and the condition number $\mathcal{C}(d)$. Renegar [24] studied the relation of $\mathcal{C}(d)$ to sensitivity of solutions of (FP$_d$) under perturbations in the problem data. (This issue was also investigated earlier by Robinson [28].) Peña and Renegar [22] discussed the role of $\mathcal{C}(d)$ in the complexity of computing approximate solutions of (FP$_d$). Freund and Vera [7] and Peña [20] addressed the theoretical complexity and practical aspects of computing the distance to ill-posedness. Vera [38] considered the numerical properties of an interior-point method for solving (FP$_d$) (and, in fact, a more general problem of optimizing a linear function over the feasible region of (FP$_d$)) in the case when (FP$_d$) is a linear programming problem. He considered the algorithm in the floating point arithmetic model, and demonstrated that the algorithm will approximately solve the optimization problem in polynomial time, while requiring roughly $O(\ln(\mathcal{C}(d)))$ significant digits of precision for computation. For additional discussion of ill-posedness and the condition number, see Filipowski [6, 5], Nunez and Freund [19], Nunez [18], Peña [21, 20], and Vera [35, 36, 37].

As we hope the above discussion conveys, the condition number $\mathcal{C}(d)$ is a relevant and important measure of conditioning of the problem (FP$_d$). Note that when (FP$_d$) is in fact a linear programming feasibility problem, $\mathcal{C}(d)$ provides a measure of

conditioning that, unlike L, does not rely on the assumption that the problem data is rational, and is relevant in the floating point model of computation.

Nevertheless, there are some potential drawbacks in using $\mathcal{C}(d)$ as a sole measure of conditioning of the problem (FP$_d$). To illustrate this point, note that problem (FP$_d$) of (1) can be interpreted as the problem of finding a point x in the intersection of the cone C_X with an affine subspace $\mathcal{A} \subset X$, defined as

$$\mathcal{A} \stackrel{\triangle}{=} \{x : Ax = b\} = \{x : x = x_0 + x_N,\ x_N \in \text{Null}(A)\},$$

where $x_0 \in X$ is an arbitrary point satisfying $Ax_0 = b$, and Null(A) is the null space of A. Notice that the description of the affine subspace $\mathcal{A}$ by the data instance $d = (A, b)$ is not unique. It easy to find an equivalent data instance $\tilde{d} = (\tilde{A}, \tilde{b})$ such that

$$\{x : \tilde{A}x = \tilde{b}\} = \{x : Ax = b\} = \mathcal{A}$$

(take, for example, $\tilde{b} = Bb$ and $\tilde{A} = BA$, where B is any nonsingular linear operator $B : Y \to Y$). Then the problem

$$(\text{FP}_{\tilde{d}}) : \tilde{A}x = \tilde{b},\ x \in C_X$$

is equivalent to problem (FP$_d$) in the sense that their feasible regions are identical; we can think of the systems (FP$_d$) and (FP$_{\tilde{d}}$) as different but equivalent formulations of the same feasibility problem

$$(\text{FP}): \text{find } x \in \mathcal{A} \cap C_X.$$

Since the condition number $\mathcal{C}(d)$ is, in general, different from $\mathcal{C}(\tilde{d})$, analyzing many of the properties of the problem (FP) above in terms of the condition number will lead to different results, depending on which formulation, (FP$_d$) or (FP$_{\tilde{d}}$), is being used. This observation is somewhat disconcerting, since many of these properties are of purely geometric nature. For example, the existence of a solution of small norm and the existence of a reliable solution depend only on the geometry of the feasible region, i.e., of the set $\mathcal{A} \cap C_X$, and do not depend on a specific data instance d used to "represent" the affine space $\mathcal{A}$.

An interesting research direction, therefore, is the development of relevant measures of conditioning of the problem (FP$_d$) that depend on the affine space $\mathcal{A}$ rather than on a particular data instance d used to represent it and that allow us to analyze some of the properties of the problem independently of the data used to represent the problem. The recent literature contains some results on developing such measures when (FP$_d$) is a linear programming feasibility problem. In particular, two condition measures, $\bar{\chi}_d$ and σ_d, were used in the analysis of interior-point algorithms for linear programming (Vavasis and Ye [32, 33, 34]). These measures, discussed in detail in section 4, provide a new perspective on the analysis of linear programming problems; for example, like the condition number $\mathcal{C}(d)$, they do not require the data for the problem to be rational. Also, they have the desired property that they are independent of the specific data instance d used to describe the problem and can be defined considering only the affine subspace $\mathcal{A}$. Further analysis of these measures in the setting of linear programming feasibility problems can be found in Ho [11], Todd, Tunçel, and Ye [29], and Tunçel [30].

In this paper we define a new measure of conditioning, μ_d, for feasible instances of the problem (FP$_d$) of (1), which is independent of the specific data representation

of the problem. We explore the relationship between μ_d and measures $\bar{\chi}_d$, σ_d, and $\mathcal{C}(d)$. (In particular, we demonstrate that the measure σ_d is directly related to μ_d in the special case of linear programming.) We show that $\mu_d \leq \mathcal{C}(d)$, i.e., μ_d is less conservative, and that for any data instance $\tilde{d}$ equivalent to d, $\mu_d \leq \mathcal{C}(\tilde{d})$. We also demonstrate that many important properties of the system (FP_d) previously analyzed in terms of $\mathcal{C}(d)$ can be analyzed through μ_d (independently of the data representation).

On the other hand, some properties of (FP_d) are not purely geometric and depend on the data d. Therefore, it might be beneficial, given a data instance d, to construct a data instance $\tilde{d}$ which is equivalent to d but is better conditioned in the sense that $\mathcal{C}(\tilde{d}) < \mathcal{C}(d)$. We develop a characterization of all equivalent data instances $\tilde{d}$ by introducing the concept of a *preconditioner* and provide an upper bound on the condition number $\mathcal{C}(\tilde{d})$ of the "best" equivalent data instance $\tilde{d}$. We also analyze the complexity of computing an equivalent data instance whose resulting condition number is within a known factor of this bound. To this end, we construct an algorithm for computing such a data instance and analyze its complexity.

An outline of the paper is as follows. Section 2 contains notation, definitions, assumptions, and preliminary results. In section 3 we introduce the new measure of conditioning μ_d for (FP_d), establish several results relating μ_d to geometric properties of the feasible region of (FP_d), and analyze the performance of several algorithms for solving (FP_d) in terms of μ_d. In section 4 we study the relationship between μ_d and other measures of conditioning, completely characterizing the relationship between $\mathcal{C}(d)$ and μ_d, as well as σ_d and $\bar{\chi}_d$, in the linear programming setting. In section 5, we develop the notion of a preconditioner for the problem (FP_d), establish an upper bound on the condition number $\mathcal{C}(\tilde{d})$ of the best equivalent data instance $\tilde{d}$, and construct and analyze an algorithm for computing an equivalent data instance whose condition number is within a known factor of this bound. Section 6 contains some final conclusions and indicates potential topics of future research.

2. Preliminaries. We work in the setup of finite-dimensional normed linear vector spaces. Both X and Y are normed linear spaces of finite dimension n and m, respectively, endowed with norms $\|x\|$ for $x \in X$ and $\|y\|$ for $y \in Y$. For $\bar{x} \in X$, let $B(\bar{x}, r)$ denote the ball centered at $\bar{x}$ with radius r, i.e., $B(\bar{x}, r) = \{x \in X : \|x - \bar{x}\| \leq r\}$, and define $B(\bar{y}, r)$ analogously for $\bar{y} \in Y$. We denote the set of real numbers by $\Re$ and the set of nonnegative real numbers by $\Re_+$. The set of real k-by-k symmetric matrices is denoted by $S^{k\times k}$. The set $S^{k\times k}$ is a closed linear space of dimension $n = \frac{k(k+1)}{2}$. We denote the set of symmetric positive semidefinite k-by-k matrices by $S^{k\times k}_+$. $S^{k\times k}_+$ is a closed convex cone in $S^{k\times k}$. The interior of the cone $S^{k\times k}_+$ is precisely the set of k-by-k positive definite matrices, and is denoted by $S^{k\times k}_{++}$.

We associate with X and Y the dual spaces X^* and Y^* of linear functionals defined on X and Y, respectively. Let $c \in X^*$. In order to maintain consistency with standard linear algebra notation in mathematical programming, we will denote the linear function $c(x)$ by $c^t x$. Similarly, for $f \in Y^*$ we denote $f(y)$ by $f^t y$. We denote $A(x)$ by Ax, and we denote the dual operator of A by $A^t : Y^* \to X^*$.

The dual norm induced on $c \in X^*$ is defined as

$$\|c\|_* \stackrel{\triangle}{=} \max\{c^t x : x \in X,\ \|x\| \leq 1\}, \tag{3}$$

and the Hölder inequality $c^t x \leq \|c\|_* \|x\|$ follows easily from this definition. The dual norm induced on $f \in Y^*$ is defined similarly.

We now present the development of the concepts of condition numbers and data perturbation for (FP$_d$) in detail. Recall that $d = (A, b)$ is the data for the problem (FP$_d$). Let

$$\mathcal{D} = \{d = (A, b) : A \in L(X, Y), b \in Y\}$$

denote the space of all data $d = (A, b)$ for (FP$_d$). For $d = (A, b) \in \mathcal{D}$ we define the norm on the Cartesian product $L(X, Y) \times Y$ to be

$$\|d\| = \|(A, b)\| = \max\{\|A\|, \|b\|\},$$

where $\|b\|$ is the norm specified for Y and $\|A\|$ is the operator norm, namely

$$\|A\| = \max\{\|Ax\| : \|x\| \leq 1\}.$$

We define

$$\mathcal{F} = \{(A, b) \in \mathcal{D} : \text{ there exists } x \text{ satisfying } Ax = b,\ x \in C_X\}$$

to be the set of data instances d for which (FP$_d$) is feasible. Its complement is denoted by $\mathcal{F}^{\mathcal{C}}$, the set of data instances for which (FP$_d$) is infeasible. The boundary of $\mathcal{F}$ and of $\mathcal{F}^{\mathcal{C}}$ is precisely the set $\mathcal{B} = \partial\mathcal{F} = \partial\mathcal{F}^{\mathcal{C}} = \text{cl}(\mathcal{F}) \cap \text{cl}(\mathcal{F}^{\mathcal{C}})$, where ∂S denotes the boundary and cl(S) denotes the closure of a set S. Note that if $d = (A, b) \in \mathcal{B}$, then (FP$_d$) is ill-posed in the sense that arbitrarily small changes in the data $d = (A, b)$ can yield instances of (FP$_d$) that are feasible as well as instances of (FP$_d$) that are infeasible. Also, note that $\mathcal{B} \neq \emptyset$, since $d = 0 \in \mathcal{B}$.

For a data instance $d = (A, b) \in \mathcal{D}$, the *distance to ill-posedness* is defined to be

$$\rho(d) \stackrel{\triangle}{=} \inf\{\|\Delta d\| : d + \Delta d \in \mathcal{B}\} = \begin{cases} \inf\{\|d - \bar{d}\| : \bar{d} \in \mathcal{F}^C\} & \text{if } d \in \mathcal{F}, \\ \inf\{\|d - \bar{d}\| : \bar{d} \in \mathcal{F}\} & \text{if } d \in \mathcal{F}^C; \end{cases} \tag{4}$$

see Renegar [24, 25, 26]. The *condition number* $\mathcal{C}(d)$ of the data instance d is defined to be

$$\mathcal{C}(d) = \frac{\|d\|}{\rho(d)} \tag{5}$$

when $\rho(d) > 0$, and $\mathcal{C}(d) = \infty$ when $\rho(d) = 0$. The condition number $\mathcal{C}(d)$ is a measure of the relative conditioning of the data instance d and can be viewed as a scale-invariant reciprocal of $\rho(d)$, as it is elementary to demonstrate that $\mathcal{C}(d) = \mathcal{C}(\alpha d)$ for any positive scalar α. It is easy to show that $\rho(0) = 0$, and hence $\mathcal{C}(d) \geq 1$.

If C is a convex cone in X, then the dual cone of C, denoted by C^*, is defined by

$$C^* = \{z \in X^* : z^t x \geq 0 \text{ for any } x \in C\}. \tag{6}$$

We will say that a cone C is *regular* if C is a closed convex cone, has a nonempty interior, and is pointed (i.e., contains no line). If C is a closed convex cone, then C is regular if and only if C^* is regular.

We will use the following definition of the *width* of a regular cone C.

DEFINITION 1. *If C is a regular cone in X, the* width *of C is given by*

$$\tau_C \stackrel{\triangle}{=} \max_{x,r} \left\{ \frac{r}{\|x\|} : B(x, r) \subset C \right\}.$$

Note that $\tau_C \in (0,1]$, since C is pointed and has a nonempty interior, and τ_C is attained for some $(\bar{x},\bar{r})$ as well as along the ray $(\alpha\bar{x},\alpha\bar{r})$ for all $\alpha > 0$. By choosing the value of α appropriately, we can find $u \in C$ such that

$$\|u\| = 1 \text{ and } \tau_C \text{ is attained for } (x,r) = (u,\tau_C). \tag{7}$$

DEFINITION 2. *If C is a regular cone in X, define the norm approximation coefficient by*

$$\delta_C \stackrel{\triangle}{=} \operatorname{dist}(0,\, \partial\operatorname{conv}(C(1), -C(1)), \tag{8}$$

where $C(1) \stackrel{\triangle}{=} \{x \in C : \|x\| \le 1\}$, and $\partial\operatorname{conv}(C(1),-C(1))$ is the boundary of the convex hull of the set $C(1) \cup (-C(1))$.

The norm approximation coefficient δ_C measures the extent to which the unit ball $B(0,1) \subset X$ can be approximated by the set $\operatorname{conv}(C(1),-C(1))$. As a consequence, it measures the extent to which the norm of a linear operator can be approximated over the set $C(1)$.

PROPOSITION 3. *Suppose $A \in L(X,Y)$. Then $\|A\| \le \frac{1}{\delta_C}\max\{\|Ax\| : x \in C(1)\}$.*

LEMMA 4. *Suppose C is a regular cone with width τ_C. Then*

$$\delta_C \ge \frac{\tau_C}{1+\tau_C} \ge \frac{\tau_C}{2}. \tag{9}$$

Proof. Let $\bar{x} \in X$ be an arbitrary vector satisfying $\|\bar{x}\| \le \frac{\tau_C}{1+\tau_C}$. To establish the lemma we need to show that $\bar{x} \in \operatorname{conv}(C(1),-C(1))$.

Let $x = \frac{\bar{x}(1+\tau_C)}{\tau_C}$. If u is as in (7), then $u+\tau_C x \in C$ and $u-\tau_C x \in C$. Furthermore,

$$\frac{u+\tau_C x}{1+\tau_C} \in C(1) \quad \text{and} \quad \frac{-u+\tau_C x}{1+\tau_C} \in -C(1),$$

and so

$$\bar{x} = \frac{\tau_C}{1+\tau_C}x = \frac{1}{2}\left(\frac{u+\tau_C x}{1+\tau_C}\right) + \frac{1}{2}\left(\frac{-u+\tau_C x}{1+\tau_C}\right) \in \operatorname{conv}(C(1),-C(1)). \qquad \square$$

We will assume throughout this paper that the system (FP_d) of (1) is feasible. At this point we make no further assumptions on the cone C_X and the norms on the spaces X and Y unless stated otherwise. (We will make some additional assumptions in sections 4 and 5.)

When (FP_d) is feasible, $\rho(d)$ can be expressed via the following characterization:

$$\rho(d) = \max\{r : B(0,r) \subseteq \mathcal{H}_d\}, \tag{10}$$

where

$$\mathcal{H}_d \stackrel{\triangle}{=} \{b\theta - Ax : \theta \ge 0,\ x \in C_X,\ |\theta| + \|x\| \le 1\} \subset Y. \tag{11}$$

Note that $0 \in \mathcal{H}_d$ whenever (FP_d) is feasible, and $\rho(d) > 0$ precisely when $0 \in \operatorname{int}\mathcal{H}_d$. This interpretation, presented by Renegar in [26], will serve as an important tool in developing further understanding of the properties of the system (FP_d).

The next result follows from the definition of $\mathcal{H}_d$ and Proposition 3.

COROLLARY 5. *Suppose that $d = (A,b) \in \mathcal{D}$ and C_X is regular. Then $\|d\| \le \frac{1}{\delta_{C_X}}\max\{\|h\| : h \in \mathcal{H}_d\}$.*

3. The symmetry measure μ_d. In this section we define a new measure of conditioning of (FP$_d$), μ_d, which we refer to as the "symmetry measure," and we establish some of its properties relevant in the analysis of (FP$_d$). We begin by recalling the *symmetry* of a set with respect to a point, in the following definition.

DEFINITION 6. *Let $D \subset Y$ be a bounded convex set. For $y \in \operatorname{int} D$ we define* $\operatorname{sym}(D, y)$ *to be the symmetry of D about y, i.e.,*

$$\operatorname{sym}(D, y) \stackrel{\triangle}{=} \sup\{t \mid y + v \in D \Rightarrow\ y - tv \in D\}.$$

If $y \in \partial D$, we define $\operatorname{sym}(D, y) = 0$.

This definition of symmetry is equivalent to that given in [26]. Observe that $\operatorname{sym}(D, y) \in [0, 1]$, with $\operatorname{sym}(D, y) = 1$ if D is perfectly symmetric about y, and $\operatorname{sym}(D, y) = 0$ precisely when $y \in \partial D$. Moreover, the definition of $\operatorname{sym}(D, y)$ is independent of the norm on the space Y.

LEMMA 7. *Suppose that D is a compact convex set with a nonempty interior, and let $y \in \operatorname{int} D$. Then there exists an extreme point w of D such that* $\operatorname{sym}(D, y) = \operatorname{sym}_w(D, y) \stackrel{\triangle}{=} \sup\{t \mid y - t(w - y) \in D\}$.

Proof. Define $f(w) = \operatorname{sym}_w(D, y) = \sup\{t \mid y - t(w - y) \in D\}$. It follows that $f(w)$ is a quasi-concave function on D. This implies that the minimum of $f(w)$ is attained at an extreme point of D; see, for example, section 3.5.3 of [1]. ∎

To define the *symmetry measure* of the problem (FP$_d$) recall that if (FP$_d$) is feasible, then $0 \in \mathcal{H}_d$, where $\mathcal{H}_d$ is defined in (11). Hence, the following quantity is well-defined.

DEFINITION 8. *Suppose the system* (FP$_d$) *is feasible. We define*

$$\mu_d \stackrel{\triangle}{=} \frac{1}{\operatorname{sym}(\mathcal{H}_d, 0)} \tag{12}$$

when $\operatorname{sym}(\mathcal{H}_d, 0) > 0$, *and* $\mu_d = +\infty$ *when* $\operatorname{sym}(\mathcal{H}_d, 0) = 0$.

From the above definition, $\mu_d \geq 1$ and $\mu_d = +\infty$ precisely when $0 \in \partial\mathcal{H}_d$, i.e., precisely when (FP$_d$) is ill-posed.

3.1. The symmetry measure and geometric properties of solutions of (FP$_d$). We now establish two results that characterize geometric properties of the feasible region X_d of the system (FP$_d$) in terms of μ_d. Theorem 9 establishes a bound on the size of a solution of (FP$_d$) in terms of μ_d; this result is similar to the bound in terms of the condition number $\mathcal{C}(d)$ in [24]. Theorem 10 demonstrates existence of a reliable solution of (FP$_d$). This is similar to the result (2) presented in [8]; however, here the bounds on the size of the solution, its distance to the boundary of the cone C_X, and the ratio of the above quantities are established in terms of μ_d rather than $\mathcal{C}(d)$. Also, unlike for the condition number $\mathcal{C}(d)$, we can establish a converse result for μ_d; namely, if the feasible region possesses nice geometry, i.e., contains a reliable solution, then μ_d can be nicely bounded by a function of the parameters associated with the reliable solution. This result is proven in Theorem 11.

THEOREM 9. *Suppose $\mu_d < \infty$. Then there exists $x \in X_d$ such that $\|x\| \leq \mu_d$.*

Proof. By the definition of μ_d, $-\frac{1}{\mu_d} b = -\operatorname{sym}(\mathcal{H}_d, 0) b \in \mathcal{H}_d$, since $b \in \mathcal{H}_d$. Therefore there exists (θ, x) satisfying $\theta \geq 0,\ x \in C_X,\ |\theta| + \|x\| \leq 1$, and $b\theta - Ax = -\frac{1}{\mu_d} b$. Let $\hat{x} = x/(\theta + \frac{1}{\mu_d})$. Then $\hat{x} \in X_d$ and $\|\hat{x}\| = \|x\|/(\theta + \frac{1}{\mu_d}) \leq \mu_d$. ∎

THEOREM 10. *Suppose C_X is a regular cone with width τ, and that $\mu_d < \infty$. Then there exist $\hat{x}$ and $r > 0$ such that*

1. $\hat{x} \in X_d$,
2. $\|\hat{x}\| \leq 2\mu_d + 1$,
3. $\operatorname{dist}(\hat{x}, \partial C_X) \geq r \geq \frac{\tau}{2\mu_d+1}$,
4. $\frac{\|\hat{x}\|}{r} \leq \frac{2\mu_d+1}{\tau}$.

Proof. Let u be as in (7). Then $\frac{1}{2}b - \frac{1}{2}Au \in \mathcal{H}_d$. From the definition of μ_d we conclude that $-\frac{1}{\mu_d}\left(\frac{1}{2}b - \frac{1}{2}Au\right) \in \mathcal{H}_d$, whereby there exists $(\bar{\theta}, \bar{x}) \in \Re_+ \times C_X$, $|\bar{\theta}| + \|\bar{x}\| \leq 1$, satisfying $b\bar{\theta} - A\bar{x} = -\frac{1}{\mu_d}(\frac{1}{2}b - \frac{1}{2}Au)$.

Let $\hat{x} = \frac{2\mu_d\bar{x}+u}{2\mu_d\bar{\theta}+1}$. It is easy to verify that $\hat{x} \in X_d$, so that condition 1 of the theorem is satisfied. Moreover, $\|\hat{x}\| = \frac{\|2\mu_d\bar{x}+u\|}{2\mu_d\bar{\theta}+1} \leq 2\mu_d + 1$, establishing condition 2.

Next, let $r = \frac{\tau}{2\mu_d\bar{\theta}+1}$. Since $B(u, \tau) \subset C_X$ and $\bar{x} \in C_X$, we conclude that $B(u + 2\mu_d\bar{x}, \tau) \subset C_X$, and therefore $B(\frac{u+2\mu_d\bar{x}}{2\mu_d\bar{\theta}+1}, \frac{\tau}{2\mu_d\bar{\theta}+1}) = B(\hat{x}, r) \subset C_X$. Also, since $\bar{\theta} \leq 1$, $r \geq \frac{\tau}{2\mu_d+1}$, establishing condition 3. Finally,

$$\frac{\|\hat{x}\|}{r} = \frac{\|2\mu_d\bar{x} + u\|}{2\mu_d\bar{\theta} + 1} \cdot \frac{2\mu_d\bar{\theta} + 1}{\tau} \leq \frac{2\mu_d + 1}{\tau},$$

implying condition 4 and concluding the proof of the theorem. ∎

We conclude from Theorems 9 and 10 that, much like for the condition number $\mathcal{C}(d)$, if the symmetry measure μ_d is small, then the feasible region X_d possesses nice geometry. We now establish a converse result.

THEOREM 11. *Suppose C_X is a regular cone and there exists $\hat{x} \in X_d$ and $r > 0$ such that $\operatorname{dist}(\hat{x}, \partial C_X) \geq r$. Let $\gamma = \max\{\|\hat{x}\|, \frac{1}{r}, \frac{\|\hat{x}\|}{r}\}$. Then $\mu_d \leq 1 + 2\gamma$.*

Proof. Let $\delta = \|\hat{x}\| + 1$ and $\pi = \min\{r, 1\}$. We first show that $\operatorname{sym}(\mathcal{H}_d, 0) \geq \frac{\pi}{\delta+\pi}$.

Let $y \in \mathcal{H}_d$. From the definition of $\mathcal{H}_d$, $y = b\bar{\theta} - A\bar{x}$ for some $(\bar{\theta}, \bar{x}) \in \Re_+ \times C_X$, $|\bar{\theta}| + \|\bar{x}\| \leq 1$. Therefore

$$\frac{\pi}{\delta+\pi}(-y) = \frac{\pi}{\delta+\pi}(-b\bar{\theta} + A\bar{x}) + \frac{1}{\delta+\pi}(b - A\hat{x}) = b\left(\frac{-\pi\bar{\theta}+1}{\delta+\pi}\right) - A\left(\frac{-\pi\bar{x}+\hat{x}}{\delta+\pi}\right).$$

Let $\check{\theta} = \frac{-\pi\bar{\theta}+1}{\delta+\pi}$ and $\check{x} = \frac{-\pi\bar{x}+\hat{x}}{\delta+\pi}$. Since $\pi \leq 1$ and $\bar{\theta} \leq 1$, we have $\check{\theta} \geq 0$. Moreover, since $\pi \leq r$ and $\|\bar{x}\| \leq 1$, we have $\check{x} \in C_X$. Finally, $|\check{\theta}| + \|\check{x}\| \leq \frac{1}{\delta+\pi}(1 + \pi\|\bar{x}\| + \|\hat{x}\|) \leq 1$, and therefore $-\frac{\pi}{\delta+\pi}y \in \mathcal{H}_d$ for an arbitrary $y \in \mathcal{H}_d$, establishing that $\operatorname{sym}(\mathcal{H}_d, 0) \geq \frac{\pi}{\delta+\pi}$. Hence,

$$\mu_d = \frac{1}{\operatorname{sym}(\mathcal{H}_d, 0)} \leq \frac{\delta+\pi}{\pi} = 1 + \frac{1}{\min\{r,1\}} + \frac{\|\hat{x}\|}{\min\{r,1\}} \leq 1 + \max\{\gamma, 1\} + \gamma \leq 1 + 2\gamma.$$

The last inequality follows from the observation that $r \leq \|\hat{x}\|$ (since C_X is pointed and thus $\|\hat{x}\| \geq \operatorname{dist}(\hat{x}, \partial C_X) \geq r$) and thus $\gamma \geq \frac{\|\hat{x}\|}{r} \geq 1$. ∎

The result in Theorem 11 is quite specific to μ_d; no such result is possible for the condition number $\mathcal{C}(d)$. In fact, the example following Remark 19 in section 4 shows that $\mathcal{C}(d)$ can be arbitrarily large even when γ is fixed.

3.2. The symmetry measure and the complexity of computing a solution of ($\mathbf{FP_d}$). In this subsection we present complexity bounds for solving (FP_d) via an interior-point algorithm and via the ellipsoid algorithm, and we show that the

complexity of solving (FP$_d$) depends on $\ln(\mu_d)$ as well as on other naturally appearing quantities. For this subsection, we assume that the space X is an n-dimensional Euclidean space with Euclidean norm $\|x\| = \|x\|_2 = \sqrt{x^t x}$ for $x \in X$. We also assume that C_X is a regular cone with width τ and that the vector u of (7) is known.

When the cone C_X is represented as the closure of the domain of a self-concordant barrier function, a solution of (FP$_d$) can be found using the barrier method developed by Renegar, based on the theory of self-concordant functions of Nesterov and Nemirovskii [17]. Below we briefly review the barrier method as articulated in [27] and then state the main complexity result.

The version of the barrier method that we will use is designed to approximately solve a problem of the form

$$z_* = \inf\{c^t\omega : \omega \in S \cap L\}, \tag{13}$$

where S is a bounded set whose interior is convex and is the domain of a self-concordant barrier function $f(\omega)$ with complexity parameter ϑ_f (see [17] and [27] for details), and L is a closed subspace (or a translate of a closed subspace). The barrier method takes as input a point $\omega' \in \operatorname{int} S \cap L$, and proceeds by approximately following the *central path*, i.e., the sequence of solutions of the problems

$$z(\eta) = \inf_{\omega \in L} \eta \cdot c^t\omega + f(\omega),$$

where $\eta > 0$ is the *barrier parameter*. In particular, after the initialization stage, the method generates an increasing sequence of barrier parameters $\eta_k > 0$ and iterates $\omega_k \in \operatorname{int} S \cap L$ that satisfy

$$c^t\omega_k - \frac{6\vartheta_f}{5\eta_k} \le z_* \le c^t\omega_k, \qquad k = 0, 1, 2, \ldots. \tag{14}$$

It follows from the analysis in [27] that if the barrier method is initialized at the point $\omega' \in \operatorname{int} S \cap L$, then it will take at most

$$O\left(\sqrt{\vartheta_f} \ln\left(\frac{\vartheta_f(z^* - z_*)}{\operatorname{sym}(S \cap L, \omega')} \cdot \bar{\eta}\right)\right) \tag{15}$$

iterations to bring the value of the barrier parameter η above the threshold of $\bar{\eta} \ge \eta_0$ while maintaining (14). (Here, $z^* = \sup\{c^t\omega : \omega \in S \cap L\}$.) This implies the main convergence result for the barrier method, which follows.

THEOREM 12 (see [27, Theorem 2.4.10]). *Assume that S is a bounded set whose interior is convex and is the domain of a self-concordant barrier function $f(\omega)$ with complexity parameter ϑ_f, and that L is a closed subspace (or a translate of a closed subspace). Assume that the barrier method is initialized at a point $\omega' \in \operatorname{int} S \cap L$. If $0 < \epsilon < 1$, then within*

$$O\left(\sqrt{\vartheta_f} \ln\left(\frac{\vartheta_f}{\epsilon \operatorname{sym}(S \cap L, \omega')}\right)\right)$$

iterations of the method, all points ω computed thereafter satisfy $\omega \in \operatorname{int} S \cap L$ and

$$\frac{c^t\omega - z_*}{z^* - z_*} \le \epsilon.$$

In order to find a solution of (FP$_d$) we will construct a closely related problem of the form (13) and apply the barrier method to this problem. This construction was

carried out in [26], where the complexity of solving (FP_d) was analyzed in terms of $\mathcal{C}(d)$. The optimization problem we consider is

$$\begin{array}{lll} z_* = & \inf_{\theta,x,t} & t \\ & \text{subject to (s.t.)} & b\theta - Ax = t(\frac{1}{2}b - \frac{1}{2}Au), \\ & & x \in \operatorname{int} C_X, \\ & & \|x\| < 1, \\ & & 0 < \theta < 1, \\ & & -1 < t < 2, \end{array} \tag{16}$$

where u is chosen as in (7). We will use the barrier method to find a feasible solution $(\hat{\theta}, \hat{x}, \hat{t})$ of (16) such that $\hat{t} \le 0$, and use the transformation $x = \hat{x} - \frac{1}{2}\hat{t}u/(\hat{\theta} - \frac{1}{2}\hat{t})$ to obtain a solution of (FP_d).

Let z^* be the optimal value of the problem obtained from (16) by replacing "inf" with "sup". Let $\tilde{f}(x)$ be the self-concordant barrier function defined on $\operatorname{int} C_X$ and let $\vartheta_{\tilde{f}}$ be the complexity parameter of $\tilde{f}(x)$. Then the set $S \stackrel{\triangle}{=} \{(\theta, x, t) : x \in \operatorname{int} C_X,\ \|x\| < 1,\ 0 < \theta < 1,\ -1 < t < 2\}$ is convex and bounded, and is the domain of the self-concordant barrier function

$$f(\omega) = f(\theta, x, t) = \tilde{f}(x) - \ln(1 - \|x\|^2) - \ln\theta - \ln(1 - \theta) - \ln(t + 1) - \ln(2 - t)$$

with complexity parameter $\vartheta_f \le \vartheta_{\tilde{f}} + 5$. (See, for example, [26] or [27] for details.) If we define $L \stackrel{\triangle}{=} \{(\theta, x, t) : b\theta - Ax = t(\frac{1}{2}b - \frac{1}{2}Au)\}$, then problem (16) is of the form (13), and we can apply the barrier method initialized at the point $\omega' = (\theta', x', t') = (\frac{1}{2}, \frac{1}{2}u, 1)$. The following proposition provides bounds on all of the parameters necessary in the analysis of the complexity of the barrier method via Theorem 12.

PROPOSITION 13. $z^* \le 2,\ -1 \le z_* \le -\frac{1}{\mu_d},\ \operatorname{sym}(S \cap L, \omega') \ge \frac{1}{12\tau}$.

Proof. The upper bound on z^* and the lower bound on z_* follow from the last constraint of (16).

Let $y = \frac{1}{2}b - \frac{1}{2}Au \in \mathcal{H}_d$. From the definition of μ_d we conclude that $-\frac{y}{\mu_d} \in \mathcal{H}_d$, so there exists (θ, x) such that $\theta \ge 0,\ x \in C_X,\ |\theta| + \|x\| \le 1,\ b\theta - Ax = -\frac{1}{\mu_d}(\frac{1}{2}b - \frac{1}{2}Au)$. Therefore $(\theta, x, -1/\mu_d)$ is in the closure of the feasible set of (16), and so $z_* \le -\frac{1}{\mu_d}$.

To establish the last statement of the proposition, we appeal to Proposition 3.3 of Renegar [26], where it is shown that ω' defined above satisfies

$$\operatorname{sym}(S \cap L, \omega') \ge \frac{1}{4}\operatorname{sym}\left(C_X(1), \frac{1}{2}u\right), \text{ where } C_X(1) = \{x : x \in C_X,\ \|x\| \le 1\}.$$

Since $B\left(\frac{1}{2}u, \frac{1}{2}\tau\right) \subset C_X(1)$, it is easy to verify that $\operatorname{sym}\left(C_X(1), \frac{1}{2}u\right) \ge \frac{\tau}{3}$, establishing the proposition. ∎

THEOREM 14. *Suppose that the barrier method for problem* (16) *is initialized at the point* $(\frac{1}{2}, \frac{1}{2}u, 1)$. *Then within*

$$O\left(\sqrt{\vartheta_{\tilde{f}}} \ln\left(\frac{\vartheta_{\tilde{f}}\mu_d}{\tau}\right)\right)$$

iterations, any iterate $(\hat{\theta}, \hat{x}, \hat{t})$ *of the algorithm will satisfy* $\hat{t} \le 0$, *and therefore* $x = \hat{x} - \frac{1}{2}\hat{t}u/(\hat{\theta} - \frac{1}{2}\hat{t})$ *is a solution of (FP$_d$).*

Proof. First note that for any iterate $(\hat{\theta}, \hat{x}, \hat{t})$ of the algorithm, $\hat{\theta} > 0$ and $\hat{x} \in \operatorname{int} C_X$. Therefore, it is easy to check that when $\hat{t} \leq 0$, x is well-defined and is a solution of (FP_d).

It remains to verify the number of iterations needed to generate an iterate such that $\hat{t} \leq 0$. Let $\epsilon = \frac{1}{3\mu_d}$. Applying Theorem 12 and substituting the bounds of Proposition 13 into the complexity bound, we conclude that after at most

$$O\left(\sqrt{\vartheta_f} \ln\left(\frac{\vartheta_f}{\epsilon \operatorname{sym}(S \cap L, \omega')}\right)\right) = O\left(\sqrt{\vartheta_{\tilde{f}}} \ln\left(\frac{\vartheta_{\tilde{f}} \mu_d}{\tau}\right)\right)$$

iterations of the barrier method, any iterate $(\hat{\theta}, \hat{x}, \hat{t})$ will satisfy

$$\hat{t} \leq \epsilon(z_* - z^*) + z^* \leq \frac{1}{3\mu_d}(2 - (-1)) - \frac{1}{\mu_d} = 0,$$

from which the theorem follows. ∎

When the cone C_X is represented via a separation oracle, a solution of (FP_d) can be found using a version of the ellipsoid algorithm. (See, for example, [2] and [10].) Below is a generic theorem for analyzing the ellipsoid algorithm for finding a point ω in a convex set $S \subset \Re^k$ given by a separation oracle.

THEOREM 15. *Suppose that a convex set $S \subset \Re^k$ given by a separation oracle contains a Euclidean ball of radius r centered at some point $\hat{\omega}$, and that an upper bound R on the quantity $(\|\hat{\omega}\|_2 + r)$ is known. Then if the ellipsoid algorithm is initiated with a Euclidean ball of radius R centered at $\omega^0 = 0$, the algorithm will compute a point in S in at most*

$$\lceil 2k(k+1) \ln(R/r) \rceil$$

iterations, where each iteration must perform a feasibility cut on S.

The main problem with trying to apply Theorem 15 directly to (FP_d) is that one needs to know the upper bound R in advance. Because such an upper bound is generically unknown in advance for (FP_d), we approach solving (FP_d) by considering finding a point in the following set:

$$S \stackrel{\triangle}{=} \{(\theta, x) : \theta > 0,\ x \in C_X,\ b\theta - Ax = 0\}, \tag{17}$$

which is a convex set in the linear subspace $T \stackrel{\triangle}{=} \{(\theta, x) : b\theta - Ax = 0\}$ of dimension $k = n + 1 - m$. Observe that it is easy to construct a separation oracle for S in the linear subspace T, provided that one has a separation oracle for C_X. We will use the ellipsoid algorithm to find a point $(\hat{\theta}, \hat{x}) \in S$ (working in the linear subspace T), and we use the obvious transformation $x = \frac{\hat{x}}{\hat{\theta}}$ to transform the output of the algorithm into a solution of (FP_d).

PROPOSITION 16. *Let S be as in* (17). *Then there exists a point $(\hat{\theta}, \hat{x}) \in S$ and $\hat{r} > 0$ such that*

$$B((\hat{\theta}, \hat{x}), \hat{r}) \cap \{(\theta, x) : b\theta - Ax = 0\} \subset S, \quad \|(\hat{\theta}, \hat{x})\| + \hat{r} \leq 3, \quad \text{and } \hat{r} \geq \frac{\tau}{2\mu_d}.$$

Proof. Let $y = \frac{1}{2}b - \frac{1}{2}Au \in \mathcal{H}_d$. From the definition of μ_d we conclude that $-\frac{y}{\mu_d} \in \mathcal{H}_d$, whereby there exists $(\bar{\theta}, \bar{x})$ such that

$$|\bar{\theta}| + \|\bar{x}\| \leq 1, \quad \bar{\theta} \geq 0, \quad \bar{x} \in C_X, \quad b\bar{\theta} - A\bar{x} = -\frac{1}{\mu_d}\left(\frac{1}{2}b - \frac{1}{2}Au\right).$$

Let $\hat{\omega} = (\hat{\theta}, \hat{x}) \stackrel{\triangle}{=} (\bar{\theta} + \frac{1}{2\mu_d}, \bar{x} + \frac{1}{2\mu_d}u)$ and $\hat{r} = \frac{\tau}{2\mu_d}$. Then $\hat{\omega} \in S$, $B(\hat{\omega}, \hat{r}) \cap \{(\theta, x) : b\theta - Ax = 0\} \subset S$ and

$$\|\hat{\omega}\|_2 + \hat{r} = \sqrt{(\bar{\theta} + \frac{1}{2\mu_d})^2 + \|\bar{x} + \frac{1}{2\mu_d}u\|^2} + \frac{\tau}{2\mu_d} \leq |\hat{\theta}| + \|\hat{x}\| + \frac{1}{2\mu_d} + \frac{\|u\|}{2\mu_d} + \frac{\tau}{2\mu_d} \leq 3. \qquad \square$$

The following theorem is an immediate consequence of Theorem 15 and Proposition 16.

THEOREM 17. *Suppose that the ellipsoid algorithm is applied in the linear subspace T to find a point in the set S, initialized with the Euclidean ball (in the space T) of radius $R = 3$ centered at $(\theta^0, x^0) = (0, 0)$. Then the ellipsoid algorithm will find a point in S (and hence, by transformation, a solution of* (FP_d)*) in at most*

$$\left\lceil 2(n - m + 1)(n - m + 2) \ln\left(\frac{6\mu_d}{\tau}\right) \right\rceil$$

iterations.

4. Symmetry measure and other measures of conditioning for $(\mathbf{FP}_d)$.

4.1. Symmetry measure and the condition number. In this subsection we establish a relationship between μ_d and $\mathcal{C}(d)$. As demonstrated in Theorem 18, if an instance of (FP_d) is "well-conditioned" in the sense that $\mathcal{C}(d)$ is small, then μ_d is also small. This relationship, however, is one-sided, since μ_d may carry no upper-bound information about $\mathcal{C}(d)$. In particular, in Remark 19 we exhibit a sequence of instances of (FP_d) with $\mathcal{C}(d)$ becoming arbitrarily large while μ_d remains fixed.

THEOREM 18. $\mu_d \leq \mathcal{C}(d)$.

Proof. If $\rho(d) = 0$, then $\mathcal{C}(d) = \infty$, and the statement of the theorem holds trivially. Suppose $\rho(d) > 0$. Since $B(0, \rho(d)) \subseteq \mathcal{H}_d$, we conclude that for any $v \in \mathcal{H}_d$, $-\frac{\rho(d)}{\|v\|}v \in \mathcal{H}_d$. Therefore

$$\frac{1}{\mu_d} = \mathrm{sym}(\mathcal{H}_d, 0) \geq \inf_{v \in \mathcal{H}_d} \frac{\rho(d)}{\|v\|} \geq \frac{\rho(d)}{\|d\|} = \frac{1}{\mathcal{C}(d)},$$

proving the theorem. $\square$

REMARK 19. *μ_d may carry no upper-bound information about $\mathcal{C}(d)$.*

To see why this is true, consider the parametric family of problems $(\mathrm{FP}_{d_\epsilon})$, where $d_\epsilon = (A_\epsilon, b)$:

$$b = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \text{ and } A_\epsilon = \begin{bmatrix} 1 & 1 & -1 & -1 \\ \epsilon & -\epsilon & \epsilon & -\epsilon \end{bmatrix},$$

$C_X = \Re^4_+$ and $\|x\| \stackrel{\triangle}{=} \|x\|_1$ for $x \in X$, and $\|y\| = \|y\|_2$ for $y \in Y$. Consider the values of the parameter $\epsilon \in (0, 1]$. The set $\mathcal{H}_{d_\epsilon}$ is symmetric about 0, so $\mu_{d_\epsilon} = 1$ for any value of ϵ. On the other hand, $\rho(d_\epsilon) = \epsilon$ and $\|d_\epsilon\| = \sqrt{1 + \epsilon^2}$. Therefore,

$$\mathcal{C}(d_\epsilon) = \frac{\sqrt{1 + \epsilon^2}}{\epsilon} \geq \frac{1}{\epsilon},$$

and so $\mathcal{C}(d)$ can be arbitrarily large while μ_d remains constant. Furthermore, letting $\hat{x} = (1, 1, 1, 1)$ and $r = 1$, we see that γ in Theorem 11 has fixed value $\gamma = 4$ for any $\epsilon \in (0, 1]$.

So far, we have made no assumptions on the norm on the space Y; in fact, it can be easily seen that μ_d is invariant under changes in the norm on Y. (This is not true for $\mathcal{C}(d)$.) We conclude this section by providing another interpretation of the relationship between the measures μ_d and $\mathcal{C}(d)$. As Theorem 20 indicates, when the space Y is endowed with the appropriate norm, μ_d and $\mathcal{C}(d)$ are within a constant factor of each other. To see this, define

$$\mathcal{T}_d \stackrel{\triangle}{=} -\mathcal{H}_d \cap \mathcal{H}_d. \tag{18}$$

Then $\mathcal{T}_d$ is a convex set that is symmetric about 0, and $0 \in \operatorname{int} \mathcal{T}_d$ when $\mu_d < \infty$. Therefore we can define the norm $\|\cdot\|$ on Y to be the norm induced by considering $\mathcal{T}_d$ to be the unit ball, namely:

$$\|y\| \stackrel{\triangle}{=} \min\{\alpha : y \in \alpha \mathcal{T}_d\}. \tag{19}$$

THEOREM 20. *Suppose C_X is regular and $\mu_d < \infty$. If the norm on Y is given by (19), then $\rho(d) = 1$ and $\mathcal{C}(d) \leq \frac{\mu_d}{\delta}$, where δ is the norm approximation coefficient of the cone C_X.*

Proof. The characterization of $\rho(d)$ in (10) easily implies that $\rho(d) = 1$. It remains to establish the bound on the condition number $\mathcal{C}(d)$. We have

$$\mathcal{C}(d) = \frac{\|d\|}{\rho(d)} = \|d\| \leq \frac{1}{\delta} \max\{\|y\| : y \in \mathcal{H}_d\} \leq \frac{\mu_d}{\delta}.$$

The first inequality above follows from Corollary 5. To verify the second inequality above, suppose that $y \in \mathcal{H}_d$. Then $\frac{1}{\mu_d} y \in \mathcal{H}_d$ because $\mu_d \geq 1$, and $-\frac{1}{\mu_d} y \in \mathcal{H}_d$ by the definition of μ_d. Therefore, $\frac{1}{\mu_d} y \in \mathcal{T}_d$, and so $\|\frac{1}{\mu_d} y\| \leq 1$, which implies that $\max\{\|y\| : y \in \mathcal{H}_d\} \leq \mu_d$. This inequality is sufficient to prove the theorem; one can, however, show that $\max\{\|y\| : y \in \mathcal{H}_d\} = \mu_d$. ∎

4.2. Relationships between the symmetry measure and other measures of conditioning for linear programming. In the special case when $C_X = \Re^n_+$, the problem (FP$_d$) becomes a linear feasibility problem and can be written as follows:

$$(\text{FP}_d): \ Ax = b, \qquad x \geq 0, \tag{20}$$

where $x \in \Re^n$, $b \in \Re^m$, and $A \in \Re^{m \times n}$. We assume in this subsection that (FP$_d$) has a strictly positive solution x^0, i.e., $Ax^0 = b$ and $x^0 > 0$, that the norm on X is $\|x\| \stackrel{\triangle}{=} \|x\|_1$, and that the norm on Y is $\|y\| \stackrel{\triangle}{=} \|y\|_2$.

Complexity analysis of linear programming sometimes relies on the complexity measures $\sigma_{(\cdot)}$ and $\bar{\chi}(\cdot)$. These measures are quite specific to the special case of linear programming, as opposed to $\mathcal{C}(d)$ and μ_d, which apply to more general conic problems. In this subsection we state both previously known as well as new results relating all of these condition measures, which in total provide a complete characterization of the relationship between these four measures of conditioning.

For simplicity of notation, we define an "expanded" matrix $\tilde{A} \stackrel{\triangle}{=} [b; -A] \in \Re^{m \times (n+1)}$. Notice that $\|\tilde{A}\| \stackrel{\triangle}{=} \max\{\|b\theta - Ax\| : \|(\theta, x)\|_1 \leq 1\} = \|d\|$.

We first review a slight variant on $\sigma_{(\cdot)}$ called σ_d, which was introduced and used in the complexity analysis of an interior-point algorithm for solving (FP$_d$) by Vavasis and Ye [32]:

$$\sigma_d \stackrel{\triangle}{=} \min_{j=1,\ldots,n+1} \max_w \{e_j^t w : \tilde{A} w = 0, \ e^t w = 1, \ w \geq 0\},$$

where $e_j,\ j = 1, \ldots, n+1$, denotes the jth unit vector and $e \in \Re^{n+1}$ is the vector of all ones. Note that while the above does not coincide with the usual definition of σ, it does under our assumption that (FP_d) has a strictly positive solution.

We also review a slight variant on $\bar{\chi}_{(\cdot)}$ called $\bar{\chi}_d$, which has been used by Vavasis and Ye [33, 34] and Megiddo, Mizuno, and Tsuchiya [16] in the complexity analysis of another interior-point algorithm:

$$\bar{\chi}_d \stackrel{\triangle}{=} \sup\{\|\tilde{A}^t(\tilde{A}D\tilde{A}^t)^{-1}\tilde{A}D\| : D \in S_{++}^{(n+1)\times(n+1)},\ D \text{ diagonal}\}.$$

An alternative characterization of $\bar{\chi}_d$ is

$$\bar{\chi}_d = \max\{\|B^{-1}\tilde{A}\| : B \in \mathcal{B}(\tilde{A})\}, \tag{21}$$

where $\mathcal{B}(\tilde{A})$ is the set of all bases (i.e., $m \times m$ nonsingular submatrices) of $\tilde{A}$. (See [29] for the proof of the equivalence of these characterizations.)

It has been established by Vavasis and Ye [32] that σ_d and $\bar{\chi}_d$ are related by the inequality

$$\sigma_d \geq \frac{1}{\bar{\chi}_d + 1}.$$

On the other hand, Tunçel in [31] established that, in general, σ_d may carry no upper-bound information about $\bar{\chi}_d$. Specifically, he provided a family of data instances d_ϵ such that for any $\epsilon > 0$, $\sigma_{d_\epsilon} = \frac{1}{2}$, but $\bar{\chi}_{d_\epsilon} \geq \frac{1}{\epsilon}$, and so $\bar{\chi}_{d_\epsilon}$ can be arbitrarily large.

Theorem 18 and Remark 19 established a relationship between μ_d and $\mathcal{C}(d)$. Below we establish relationships between the other pairs of measures μ_d, $\mathcal{C}(d)$, $\bar{\chi}_d$, and σ_d, or provide examples that show that no such relationship exists, in the spirit of [31].

REMARK 21. *$\mathcal{C}(d)$ and $\bar{\chi}_d$ may carry no upper-bound or lower-bound information about each other.*

To establish the above result, we provide two parametric families of matrices $\tilde{A}_\epsilon$ such that by varying the value of the parameter $\epsilon > 0$ we can make one of the above measures arbitrarily bad while keeping the other measure constant or bounded.

First consider the family of matrices $\tilde{A}_\epsilon = \left[\begin{smallmatrix} \epsilon & 0 & -\epsilon \\ 0 & 1 & -1 \end{smallmatrix}\right]$. For $\epsilon > 0$ and sufficiently small, $\rho(d_\epsilon) = \frac{\epsilon}{\sqrt{\epsilon^2+4}}$. Furthermore, $\|d_\epsilon\| = \sqrt{1+\epsilon^2}$, and so

$$\mathcal{C}(d_\epsilon) = \sqrt{\frac{\epsilon^2+1}{\epsilon^2+4}} \cdot \frac{1}{\epsilon} \to +\infty \text{ as } \epsilon \to 0.$$

On the other hand, it is easy to establish using (21) that $\bar{\chi}(d_\epsilon) = \sqrt{2}$ for any $\epsilon > 0$.

To establish the second claim of the remark, consider the family $\tilde{A}_\epsilon = [1\ \epsilon\ -1]$ with $0 < \epsilon < 1$. We have $\|d_\epsilon\| = 1$, $\rho(d_\epsilon) = 1$, and so $\mathcal{C}(d_\epsilon) = 1$ for any ϵ as above. On the other hand it is easy to establish using (21) that for any $\epsilon \in (0, 1)$, $\bar{\chi}_{d_\epsilon} = \frac{1}{\epsilon} \to +\infty$ as $\epsilon \to 0$.

PROPOSITION 22. *Suppose the system (FP_d) of (20) has a positive solution. Then $\sigma_d = \frac{1}{1+\mu_d}$.*

Proof. Observe that we can redefine σ_d as follows:

$$\sigma_d = \min_{j=1,\ldots,n+1} \sigma_j, \text{ where } \sigma_j \stackrel{\triangle}{=} \max\{e_j^t w : \tilde{A}w = 0,\ e^t w = 1,\ w \geq 0\}.$$

From Lemma 7, there exists an extreme point $\bar{w}$ of

$$\mathcal{H}_d = \{b\theta - Ax : (\theta, x) \geq 0,\ \|(\theta, x)\|_1 \leq 1\} = \{\tilde{A}w : w \geq 0,\ e^t w \leq 1\}$$

such that $\frac{1}{\mu_d} = \text{sym}_{\bar{w}}(\mathcal{H}_d, 0) = \sup\{t : -t\bar{w} \in \mathcal{H}_d\}$. Since the set of extreme points of the set $\mathcal{H}_d$ is contained in the set $\{\tilde{A}_1, \ldots, \tilde{A}_{n+1}\}$, where $\tilde{A}_j \in \Re^m$ is the jth column of the matrix $\tilde{A}$, we can characterize μ_d as

$$\frac{1}{\mu_d} = \min_{j=1,\ldots,n+1} \frac{1}{\mu_j}, \text{ where } \frac{1}{\mu_j} \stackrel{\triangle}{=} \sup\{t : -t\tilde{A}_j \in \mathcal{H}_d\}.$$

We will now show that for any j

$$\sigma_j = \frac{1}{1+\mu_j}. \tag{22}$$

Without loss of generality we can consider $j = 1$ and the corresponding column $\tilde{A}_1$ of $\tilde{A}$. If $\tilde{A}_1 = 0$, then $\sigma_1 = 1$, $\frac{1}{\mu_1} = +\infty$, and (22) holds as a limiting relationship.

Suppose that $A_1 \neq 0$, and therefore $\mu_1 > 0$ and $\sigma_1 < 1$. By definition of μ_1, $-\frac{1}{\mu_1}\tilde{A}_1 \in \mathcal{H}_d$, i.e., there exists a point $p \geq 0,\ e^t p = 1$ such that $-\frac{1}{\mu_1}\tilde{A}_1 = \tilde{A}p$. Define $w \stackrel{\triangle}{=} \frac{\mu_1 p + e_1}{1+\mu_1}$. Then $w \geq 0$, $e^t w = 1$, and $\tilde{A}w = 0$. Therefore, $\sigma_1 \geq w_1 \geq \frac{1}{1+\mu_1}$.

Suppose now that w is a solution of the linear program defining σ_1. Then $w_1 = \sigma_1$. Let $p = \frac{1}{1-\sigma_1}(w - \sigma_1 e_1)$. Then $p \geq 0$, $e^t p = 1$, and $\tilde{A}p = \frac{-\tilde{A}_1\sigma_1}{1-\sigma_1}$. Therefore, $\frac{1}{\mu_1} \geq \frac{\sigma_1}{1-\sigma_1}$, and so $\sigma_1 \leq \frac{1}{\mu_1+1}$. Combining this with the bound in the previous paragraph, we conclude that $\sigma_1 = \frac{1}{\mu_1+1}$, and by similar argument, $\sigma_j = \frac{1}{\mu_j+1},\ j = 1, \ldots, n+1$.

Suppose now that $\sigma_d = \sigma_j$ for some j. That means that $\sigma_j \leq \sigma_i$ for any index i, or, equivalently, $\frac{1}{\mu_j+1} \leq \frac{1}{\mu_i+1}$ and hence $\mu_j \geq \mu_i$ for any index i. Therefore, $\mu_d = \mu_j$ and hence $\sigma_d = \frac{1}{1+\mu_d}$. □

The following two remarks, which are easy consequences of Proposition 22, establish the remaining relationships between the four measures of conditioning.

REMARK 23. $\mu_d \leq \bar{\chi}_d$. *However, μ_d may carry no upper-bound information about $\bar{\chi}_d$.*

REMARK 24. $\sigma_d \geq \frac{1}{\mathcal{C}(d)+1}$. *However, σ_d may carry no upper-bound information about $\mathcal{C}(d)$.*

In light of Proposition 22, μ_d can in fact be viewed as a generalization of the Vavasis–Ye measure σ_d to a general conic linear system. Related to this, Ho in [11] provides an argument indicating that extending $\bar{\chi}_d$ to general conic systems is not possible.

5. Preconditioners for conic linear systems. In this section we present a characterization of all data instances $\tilde{d}$ equivalent to d (in the sense that $X_d = X_{\tilde{d}}$), by introducing the concept of a *preconditioner*, and we provide an upper bound on the condition number $\mathcal{C}(\tilde{d})$ of the "best" equivalent data instance $\tilde{d}$. We conclude by analyzing the complexity of computing an equivalent data instance whose condition number is within a known factor of this bound, by constructing an algorithm for computing such an instance and analyzing its complexity.

Consider the data instance $d = (A, b) \in \mathcal{D}$ defining the system (FP_d). Let $B \in \Re^{m\times m}$ be a given nonsingular matrix and consider the data instance $Bd \stackrel{\triangle}{=} B \cdot d = (BA, Bb)$, which gives rise to the system

$$(\text{FP}_{Bd}) :\ BAx = Bb, \qquad x \in C_X. \tag{23}$$

The systems (FP_d) and (FP_{Bd}) are equivalent; for this reason we say that the data instances d and Bd are equivalent as well. We can view the systems (FP_d) and (FP_{Bd})

as different formulations of the same feasibility problem (FP): find $x \in \mathcal{A} \cap C_X$, where $\mathcal{A}$ is the affine subspace

$$\mathcal{A} \stackrel{\triangle}{=} \{x : Ax = b\} = \{x : BAx = Bb\}. \tag{24}$$

However the condition numbers of the two systems, $\mathcal{C}(d)$ and $\mathcal{C}(Bd)$, are, in general, not equal.

On the other hand, consider the symmetry measures of the two systems, μ_d and μ_{Bd}. Observe that

$$\mathcal{H}_{Bd} \stackrel{\triangle}{=} \{Bb\theta - BAx : \theta \geq 0,\ x \in C_X,\ |\theta| + \|x\| \leq 1\} = B(\mathcal{H}_d);$$

i.e., the set $\mathcal{H}_{Bd}$ is the image of the set $\mathcal{H}_d$ under the linear transformation defined by B. Therefore, $\mathrm{sym}(\mathcal{H}_{Bd}, 0) = \mathrm{sym}(\mathcal{H}_d, 0)$, and $\mu_d = \mu_{Bd}$, since the symmetry of a set is preserved under nonsingular linear transformation, and so we can think of μ_d as depending on the affine space $\mathcal{A}$ defined in (24) but not on the specific data d. To highlight the independence of μ_d of the particular data d, we sometimes write $\mu_{\mathcal{A}}$ in place of μ_d. We record this formally in the following proposition.

PROPOSITION 25. *Let $d = (A, b) \in \mathcal{D}$, let $B \in \Re^{m \times m}$ be a nonsingular matrix, and define $\mathcal{A} \stackrel{\triangle}{=} \{x : Ax = b\}$. Then $\mu_d = \mu_{Bd} = \mu_{\mathcal{A}}$.*

We leave to the reader the proof of the next proposition.

PROPOSITION 26. *Suppose C_X is a regular cone. Let $d = (A, b) \in \mathcal{D}$ and $\tilde{d} = (\tilde{A}, \tilde{b}) \in \mathcal{D}$ be such that $X_d = X_{\tilde{d}}$. If $\mathcal{C}(d) < \infty$, then there exists a nonsingular matrix $B \in \Re^{m \times m}$ such that $\tilde{d} = Bd$.*

Suppose that a feasibility problem can be represented via two equivalent data instances d and $\tilde{d}$, and suppose that $\mathcal{C}(d) \ll \mathcal{C}(\tilde{d})$. If one were to predict, for example, the performance of the interior-point algorithm from section 3 for solving (FP_d) by analyzing its complexity in terms of the condition number, the bounds would be overly conservative if the problem were described by the data instance $\tilde{d}$. However, our analysis of the performance of the algorithm in terms of $\mu_{\mathcal{A}}$ yields a bound independent of the data instance used.

On the other hand, as detailed in the introduction, the condition number $\mathcal{C}(d)$ is a crucial parameter for analyzing properties of (FP_d) that depend on the representation of the problem ($\mathrm{FP}_{(\cdot)}$) by a specific data instance d, such as sensitivity of the feasible region to data perturbations, numerical properties of computations in algorithms for solving (FP_d), etc. Therefore, it might be beneficial to *precondition* the system (FP_d), i.e., to find another data instance $\tilde{d} = Bd$ for which $\mathcal{C}(\tilde{d}) < \mathcal{C}(d)$, and work with the corresponding system ($\mathrm{FP}_{\tilde{d}}$), which is better-behaved. In this light, we can view the matrix B above as a *preconditioner* for the system (FP_d), yielding the preconditioned system ($\mathrm{FP}_{\tilde{d}}$) with $\tilde{d} = Bd$, and Proposition 26 implies that any data instance $\tilde{d}$ for which $X_{\tilde{d}} = X_d$ can be obtained by preconditioning d with an appropriate B.

In the remainder of this section, we characterize a so-called best preconditioner, which is a preconditioner that gives rise to a condition number that is within a constant factor of the best possible, and we construct and analyze an algorithm for computing a preconditioner that yields a condition number that is within a known factor of this bound. For the remainder of this section, we assume that the space Y is the m-dimensional Euclidean space $\Re^m$ with Euclidean norm $\|y\| = \|y\|_2 = \sqrt{y^t y}$. We assume that the cone C_X is a regular cone with width τ and norm approximation coefficient δ. We also assume that $m \geq 2$. (In fact, the case $m = 1$ is trivial since in this case $\mu_{\mathcal{A}}$ and $\mathcal{C}(d)$ are within a factor of δ of each other, and thus the issue of preconditioning is essentially irrelevant.)

5.1. Best preconditioners and α-roundings. The main result of this subsection, Theorem 30, demonstrates the existence of a preconditioner $\bar{B}$ such that $\mathcal{C}(\bar{B}d)$ is within the factor $\frac{\sqrt{m}}{\delta}$ of $\mu_{\mathcal{A}}$. We begin by developing the tools to prove this result.

For any matrix $Q \in S^{m\times m}_{++}$ we define E_Q to be the ellipsoid $E_Q \stackrel{\triangle}{=} \{y \in Y : y^tQ^{-1}y \leq 1\}$.

DEFINITION 27. *Let $S \subset Y$ be a bounded set with a nonempty convex interior. For $\alpha \in (0,1]$, an ellipsoid E_Q is called an* α-rounding *of S if*

$$\alpha E_Q \subseteq S \subseteq E_Q.$$

We refer to the parameter α as the *tightness* of the rounding E_Q.

If the set S above satisfies $S = -S$ (i.e., is symmetric about 0), then S possesses a $\frac{1}{\sqrt{m}}$-rounding, i.e., there exists an ellipsoid E_Q such that $\frac{1}{\sqrt{m}}E_Q \subseteq S \subseteq E_Q$ (see John [12]). In particular, the ellipsoid of minimum volume containing S (often referred to as the Löwner–John ellipsoid of S) is a $\frac{1}{\sqrt{m}}$-rounding of S.

The following lemma allows us to interpret preconditioning of the system (FP_d) by B as constructing a $\frac{1}{\mathcal{C}(Bd)}$-rounding of the set $\mathcal{H}_d$.

LEMMA 28. *Let $B \in \Re^{m\times m}$ be a (nonsingular) preconditioner for the system (FP_d). Let $Q = \|Bd\|^2(B^tB)^{-1}$. Then*

$$\frac{1}{\mathcal{C}(Bd)}E_Q \subseteq \mathcal{H}_d \subseteq E_Q.$$

Proof. First, observe that $Q \in S^{m\times m}_{++}$, since B is nonsingular. To prove the first inclusion, let $h \in \frac{1}{\mathcal{C}(Bd)}E_Q$, i.e., $h^tQ^{-1}h \leq \frac{1}{\mathcal{C}(Bd)^2}$. Using the definition of Q, we have $h^t(B^tB)h \leq \frac{\|Bd\|^2}{\mathcal{C}(Bd)^2} = \rho(Bd)^2$, that is, $\|Bh\| \leq \rho(Bd)$. This implies $Bh \in \mathcal{H}_{Bd}$, and hence, $h \in \mathcal{H}_d$.

Next, suppose $h \in \mathcal{H}_d$, and so $Bh \in \mathcal{H}_{Bd}$. Then $\|Bh\| \leq \|Bd\|$, and therefore

$$h^tQ^{-1}h = h^t\left(\|Bd\|^2(B^tB)^{-1}\right)^{-1}h = \frac{\|Bh\|^2}{\|Bd\|^2} \leq 1,$$

i.e., $h \in E_Q$. ∎

LEMMA 29. *Let $Q \in S^{m\times m}_{++}$ be such that E_Q is an α-rounding of the set $\mathcal{T}_d$ of (18). Let $B = Q^{-\frac{1}{2}}$. Then B is a preconditioner for the system (FP_d) such that*

$$\mathcal{C}(Bd) \leq \frac{\mu_{\mathcal{A}}}{\alpha\delta} \leq \frac{2\mu_{\mathcal{A}}}{\alpha\tau}.$$

Proof. We establish the result by providing bounds on the distance to infeasibility $\rho(Bd)$ and the size of the data $\|Bd\|$ of the system (FP_{Bd}). First, we will show that $\rho(Bd) \geq \alpha$. Let $v \in Y$ satisfy $\|v\| \leq \alpha$. Then

$$(B^{-1}v)^tQ^{-1}B^{-1}v = (B^{-1}v)^t(B\cdot B)(B^{-1}v) = \|v\|^2 \leq \alpha^2,$$

and therefore $B^{-1}v \in \alpha E_Q \subseteq \mathcal{T}_d \subseteq \mathcal{H}_d$. Thus, $v \in \mathcal{H}_{Bd}$, and so $\rho(Bd) \geq \alpha$.

Next, recall from Corollary 5 that $\|Bd\| \leq \frac{1}{\delta}\max\{\|v\| : v \in \mathcal{H}_{Bd}\}$. Let $v \in \mathcal{H}_{Bd}$. Then $y = B^{-1}v \in \mathcal{H}_d$, and $\frac{1}{\mu_{\mathcal{A}}}y \in -\mathcal{H}_d \cap \mathcal{H}_d = \mathcal{T}_d \subseteq E_Q$. Hence $\|v\|^2 = y^tB^tBy = y^tQ^{-1}y \leq \mu^2_{\mathcal{A}}$, whereby $\|Bd\| \leq \frac{\mu_{\mathcal{A}}}{\delta}$.

Combining the obtained results, $\mathcal{C}(Bd) = \frac{\|Bd\|}{\rho(Bd)} \leq \frac{\mu_{\mathcal{A}}}{\delta\alpha} \leq \frac{2\mu_{\mathcal{A}}}{\tau\alpha}$. ∎

THEOREM 30. *Suppose that* (FP$_d$) *is feasible and* $\mathcal{C}(d) < +\infty$. *Then there exists a preconditioner* $\bar{B}$ *such that*

$$\mu_{\mathcal{A}} \le \mathcal{C}(\bar{B}d) \le \frac{\sqrt{m}}{\delta} \cdot \mu_{\mathcal{A}}. \tag{25}$$

Proof. By definition, $\mathcal{T}_d$ is a bounded convex set symmetric about 0. Since $\mathcal{C}(d) < \infty$, $\mathcal{T}_d$ has a nonempty interior. Therefore, there exists $Q \in S^{m\times m}_{++}$ such that E_Q is a $\frac{1}{\sqrt{m}}$-rounding of $\mathcal{T}_d$. Applying Lemma 29 with $\alpha = \frac{1}{\sqrt{m}}$, we obtain (25). ∎

REMARK 31. *In general, the upper bound in* (25) *is tight for any* m.

We verify this remark by example. Consider the system (FP$_d$) with $n = 2m$, $C_X = \Re^{2m}_+$, $\|x\| = \|x\|_1$ (so that $\delta = 1$), and the data $d = (A, b)$ as follows:

$$b = 0 \text{ and } A = [e_1,\ -e_1, \ldots, e_m,\ -e_m],$$

where e_i is the ith unit vector in $\Re^m$. Then $\mathcal{H}_d = \mathcal{T}_d = \text{conv}\{\pm e_i,\ i = 1, \ldots, m\}$, and it can be easily verified that $\mu_{\mathcal{A}} = 1$, $\rho(d) = \frac{1}{\sqrt{m}}$, and $\|d\| = 1$, and therefore $\mathcal{C}(d) = \sqrt{m}$. Suppose B is an arbitrary preconditioner. Using Lemma 28, we can construct a $\frac{1}{\mathcal{C}(Bd)}$-rounding of the set $\mathcal{T}_d$. However, it is impossible to construct an α-rounding of the set $\text{conv}\{\pm e_i,\ i = 1, \ldots, m\}$ with $\alpha > \frac{1}{\sqrt{m}}$; see, for example, [10]. Therefore, $\mathcal{C}(Bd) \ge \sqrt{m}$ for any preconditioner B.

5.2. On the complexity of computing a good preconditioner. We present an algorithm that computes a preconditioner $\tilde{B}$ for which

$$\mathcal{C}(\tilde{B}d) \le \frac{4m\mu_{\mathcal{A}}}{\delta}. \tag{26}$$

Recall that in Lemma 29 it was shown that a tight rounding of the set $\mathcal{T}_d$ gives rise to a good preconditioner for the system (FP$_d$). In Theorem 30 we relied on the existence of a $\frac{1}{\sqrt{m}}$-rounding of the set $\mathcal{T}_d$ to establish the existence of a preconditioner $\bar{B}$ such that $\mu_{\mathcal{A}} \le \mathcal{C}(\bar{B}d) \le \frac{\sqrt{m}}{\delta}\,\mu_{\mathcal{A}}$, i.e., $\mathcal{C}(\bar{B}d)$ is within the factor of $\frac{\sqrt{m}}{\delta}$ of the lower bound. In general, we are not able to efficiently compute a $\frac{1}{\sqrt{m}}$-rounding of the set $\mathcal{T}_d$. (See [10] for commentary on the difficulty of computing an approximate $\frac{1}{\sqrt{m}}$-rounding of a set $\mathcal{S}$ that does not have an efficient half-space representation.) However, the algorithm presented in this subsection will compute an ellipsoid which is a $\frac{1}{4m}$-rounding of $\mathcal{T}_d$ (also called a weak Löwner–John ellipsoid for $\mathcal{T}_d$). In particular, the algorithm of this subsection will compute a matrix $\tilde{Q} \in S^{m\times m}_{++}$ such that

$$\frac{1}{4m} E_{\tilde{Q}} \subseteq \mathcal{T}_d \subseteq E_{\tilde{Q}}, \tag{27}$$

which can be used to obtain a preconditioner $\tilde{B}$ satisfying (26) via Lemma 29. We denote this algorithm as *Algorithm WLJ* for "Weak Löwner–John."

In order to be able to efficiently implement the algorithm described in this section, we restrict the norm $\|x\|$ for $x \in X$ to the Euclidean norm $\|x\| = \|x\|_2$ (as well as maintain the assumption that $\|y\| = \|y\|_2$ for $y \in Y$). We further assume that the interior of the cone C^*_X is the domain of a self-concordant barrier $f^*(\cdot)$ with complexity parameter ϑ^*. The width of the cone C^*_X is denoted by τ^*. We assume that we know

and are given the vector $u_* \in C_X^*$ for which $\|u_*\| = 1$ and $B(u_*, \tau^*) \subset C_X^*$ as in (7). Finally, we assume that an upper bound $\bar{d}$ on $\|d\|$ is known and given or is easily computable. One could, for example, take

$$\bar{d} = \sqrt{n} \max\{\|b\|_2, \|A_1\|_2, \ldots, \|A_m\|_2\},$$

where A_j is the jth column of the matrix A. Then $\bar{d}$ approximates $\|d\|$ within the factor of $\sqrt{n}$, i.e., $\frac{1}{\sqrt{n}}\bar{d} \leq \|d\| \leq \bar{d}$.

The algorithm WLJ is a version of the parallel-cut ellipsoid algorithm; see [10]. A generic iteration of this algorithm can be described as follows. At the start of each iteration, we have a matrix $Q \in S_{++}^{m\times m}$ such that $\mathcal{T}_d \subseteq E_Q$. We compute the eigenvalue decomposition of the matrix Q. In particular, we compute the eigenvalues $0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_m$ of the matrix Q and their corresponding (orthonormal) eigenvectors $a_1, \ldots, a_m$. Then the axes of the ellipsoid E_Q are $v_i = \sqrt{\lambda_i} a_i,\ i = 1, \ldots, m$. We denote $V \stackrel{\triangle}{=} [v_1, \ldots, v_m] \in \Re^{m\times m}$. It is elementary to verify that $Q = VV^t$.

The algorithm then checks if the scaled axes $\pm\frac{1}{4\sqrt{m}} v_i$ are elements of $\mathcal{T}_d$ for $i = 1, \ldots, m$. If so, the algorithm correctly asserts that

$$\frac{1}{4m} E_Q = \frac{1}{\sqrt{m}} \cdot \frac{1}{4\sqrt{m}} E_Q \subset \text{conv}\left\{ \pm\frac{1}{4\sqrt{m}} v_i,\ i = 1, \ldots, m \right\} \subseteq \mathcal{T}_d \subseteq E_Q, \tag{28}$$

and the algorithm terminates. On the other hand, if the algorithm finds an axis $v = \pm v_j$ for some j for which $\frac{1}{4\sqrt{m}} v_j \notin \mathcal{T}_d$, then it finds a parallel cut separating the two points $\pm\frac{1}{2\sqrt{m}} v_j$ from the set $\mathcal{T}_d$, i.e., it produces a vector s such that

$$s^t v_j = 1 \text{ for some } v_j, \text{ and } \mathcal{T}_d \subseteq \left[E_Q \cap \left\{ y : -\frac{1}{2\sqrt{m}} \leq s^t y \leq \frac{1}{2\sqrt{m}} \right\} \right]. \tag{29}$$

This cut is then used to find an ellipsoid $E_{\hat{Q}}$ which satisfies

$$E_{\hat{Q}} \supset \left[E_Q \cap \left\{ y : -\frac{1}{2\sqrt{m}} \leq s^t y \leq \frac{1}{2\sqrt{m}} \right\} \right] \supseteq \mathcal{T}_d,$$

and for which

$$\frac{\text{vol}(E_{\hat{Q}})}{\text{vol}(E_Q)} \leq \frac{1}{2} e^{\frac{3}{8}}. \tag{30}$$

The formula for $\hat{Q}$ is

$$\hat{Q} = \frac{m}{m-1}\left(1 - \frac{1}{4m\xi}\right)\left(Q - \frac{m(4\xi - 1)}{4m\xi - 1} \cdot \frac{Qss^tQ}{\xi}\right), \tag{31}$$

where

$$\xi = s^t Q s = \|V^t s\|^2 \geq s^t v_j = 1; \tag{32}$$

see formula (3.1.20) of [10], for example.

In order to implement this algorithm, it is necessary to be able to check whether the rescaled axes $\pm\frac{1}{4\sqrt{m}} v_i$ are elements of $\mathcal{T}_d$, for $i = 1, \ldots, m$, and if not, it is then necessary to produce the vector s describing the parallel cut of (29). These two tasks are accomplished in a subroutine called *Weak Check*, which is outlined as follows, and for which a more complete description is furnished in the appendix.

Subroutine Weak Check.
Given the axes $v_1, \ldots, v_m$ of an ellipsoid $E_Q \supseteq \mathcal{T}_d$, either
(i) verify that $\pm\frac{1}{4\sqrt{m}} v_i \in \mathcal{T}_d$ for all $i = 1, \ldots, m$, or
(ii) find a vector s such that

$$s^t v_j = 1 \text{ for some } v_j, \text{ and } \mathcal{T}_d \subseteq \left[E_Q \cap \left\{ y : -\frac{1}{2\sqrt{m}} \le s^t y \le \frac{1}{2\sqrt{m}} \right\} \right]. \tag{33}$$

The formal description of algorithm WLJ is as follows.

ALGORITHM WLJ (Weak Löwner–John).

- *Initialization:* The algorithm is initialized with the matrix $Q^0 = \bar{d}^2 I$.
- Iteration $k \ge 1$.

  Step 1 Let $Q = Q^k$. Compute the eigenvalues $0 < \lambda_1 \le \lambda_2 \le \cdots \le \lambda_m$ of Q and the corresponding (orthonormal) eigenvectors $a_1, \ldots, a_m$. Define the axes of E_Q by $v_i = \sqrt{\lambda_i} a_i,\ i = 1, \ldots, m$.

  Step 2 Call subroutine *Weak Check* with the input $(v_1, \ldots, v_m)$. If the subroutine verifies that $\pm\frac{1}{4\sqrt{m}} v_i \in \mathcal{T}_d,\ i = 1, \ldots, m$, then return $\tilde{B} = Q^{-\frac{1}{2}}$ and terminate. Otherwise, subroutine *Weak Check* returns a vector s. Define $\hat{Q}$ by (31).

  Step 3 Let $Q^{k+1} = \hat{Q}$, $k \leftarrow k + 1$; go to Step 1.

To complete the description of Algorithm WLJ, one must specify the details of subroutine *Weak Check*. The purpose of subroutine *Weak Check* is to verify whether the rescaled axes $\pm\frac{1}{4\sqrt{m}} v_i$, $i = 1, \ldots, m$, are contained in $\mathcal{T}_d$, or to produce a parallel cut otherwise. This is accomplished by examining the following $2m$ optimization problems (P_{ϕ_v}), where $v = \pm v_i$, $i = 1, \ldots, m$:

$$(\mathrm{P}_{\phi_v}) \quad \phi_v = \max_{\phi} \quad \phi \quad \text{s.t.} \quad \phi v \in \mathcal{H}_d. \tag{34}$$

It is easy to verify that $\pm\frac{1}{4\sqrt{m}} v_i \in \mathcal{T}_d$ for all $i = 1, \ldots, m$ precisely when

$$\phi_Q \stackrel{\triangle}{=} \min_{\pm v_i} \phi_v \ge \frac{1}{4\sqrt{m}}. \tag{35}$$

(Here $\min_{\pm v_i} \phi_v$ stands for $\min\{\phi_{v_1}, -\phi_{v_1}, \ldots, \phi_{v_m}, -\phi_{v_m}\}$ in order to shorten the notation.) We will therefore implement subroutine *Weak Check* by means of approximately solving the $2m$ optimization problems (34) and checking whether condition (35) is satisfied. To solve the optimization problems (34) for every value of $v = \pm v_i,\ i = 1, \ldots, m$, we will apply the barrier method of [27] to a version of the Lagrangian dual of (P_{ϕ_v}). The formal description of this implementation is presented in the appendix, where the following complexity bound is proved.

LEMMA 32. *Subroutine* Weak Check *will terminate in at most*

$$O\left(m\sqrt{\vartheta^*} \ln\left(\frac{m\vartheta^*}{\tau^*} \cdot \frac{\bar{d}}{\sqrt{\lambda_1}} \cdot \sqrt{\frac{\lambda_m}{\lambda_1}} \right) \right) \tag{36}$$

iterations of the barrier method. Upon termination, it will either correctly verify that $\pm\frac{1}{4\sqrt{m}} v_i \in \mathcal{T}_d$ *for all* $i = 1, \ldots, m$, *or will return a vector* s *such that*

$$s^t v_j = 1 \text{ for some } v_j, \text{ and } \mathcal{T}_d \subseteq \left[E_Q \cap \left\{ y : -\frac{1}{2\sqrt{m}} \le s^t y \le \frac{1}{2\sqrt{m}} \right\} \right]. \tag{37}$$

Note that the skewness of the ellipsoid E_Q, which is square root of the ratio of the largest to the smallest eigenvalue of Q, comes to play in the complexity bound of subroutine *Weak Check*.

We now proceed to analyze the complexity of Algorithm WLJ. We first prove the volume reduction bound of (30) in Lemma 33. We then prove the main complexity of Algorithm WLJ in Theorem 34.

LEMMA 33. *Let Q be an iterate of Algorithm WLJ, and let $\hat{Q}$ be defined by* (31). *Then*

$$\frac{\text{vol}(E_{\hat{Q}})}{\text{vol}(E_Q)} \leq \frac{1}{2}e^{\frac{3}{8}}.$$

Proof. Let $R \in \Re^{m\times m}$ be an orthonormal matrix such that $RQ^{\frac{1}{2}}s = \|Q^{\frac{1}{2}}s\|e_1 = \sqrt{\xi}e_1$. Then $\hat{Q}$ can be expressed as

$$\hat{Q} = \frac{m}{m-1}\left(1 - \frac{1}{4m\xi}\right)Q^{\frac{1}{2}}R^t\left(I - \frac{m(4\xi-1)}{4m\xi-1}e_1e_1^t\right)RQ^{\frac{1}{2}}. \tag{38}$$

Therefore,

$$\begin{aligned}\det(\hat{Q}) &= \det\left(\frac{m}{m-1}\left(1 - \frac{1}{4m\xi}\right)Q^{\frac{1}{2}}R^t\left(I - \frac{m(4\xi-1)}{4m\xi-1}e_1e_1^t\right)RQ^{\frac{1}{2}}\right)\\ &= \left(\frac{m}{m-1}\left(1 - \frac{1}{4m\xi}\right)\right)^m\left(1 - \frac{m(4\xi-1)}{4m\xi-1}\right)\det(Q).\end{aligned}$$

We conclude that

$$\begin{aligned}\frac{\det(\hat{Q})}{\det(Q)} &= \left(\frac{m}{m-1}\left(1 - \frac{1}{4m\xi}\right)\right)^m\left(1 - \frac{m(4\xi-1)}{4m\xi-1}\right)\\ &= \frac{m^m(4m\xi-1)^{m-1}}{(m-1)^{m-1}(4m\xi)^m} = \frac{1}{4\xi}\left(\frac{4m\xi-1}{4m\xi-4\xi}\right)^{m-1}\\ &= \frac{1}{4\xi}\left(1 + \frac{4\xi-1}{4\xi(m-1)}\right)^{m-1} \leq \frac{1}{4\xi}e^{1-\frac{1}{4\xi}} \leq \frac{1}{4}e^{\frac{3}{4}}.\end{aligned}$$

The last inequality follows since the function te^{1-t} is an increasing function for $t \in [0,1]$, and from (32) we have $0 < \frac{1}{4\xi} \leq \frac{1}{4}$. Finally,

$$\frac{\text{vol}(E_{\hat{Q}})}{\text{vol}(E_Q)} = \frac{\sqrt{\det(\hat{Q})}}{\sqrt{\det(Q)}} \leq \frac{1}{2}e^{\frac{3}{8}}. \qquad \square$$

THEOREM 34. *Suppose $\mathcal{C}(d) < \infty$. Then Algorithm WLJ will terminate in at most*

$$O\left(m^2\sqrt{\vartheta^*}\ln^2\left(\frac{\bar{d}}{\rho(d)}\right)\ln\left(\frac{m\vartheta^*}{\tau^*}\right)\right) \tag{39}$$

iterations of the barrier method. It will return upon termination a preconditioner $\tilde{B}$ such that

$$\mu_{\mathcal{A}} \leq \mathcal{C}(\tilde{B}d) \leq \frac{4m\mu_{\mathcal{A}}}{\delta}.$$

Proof. First observe that the matrix $Q^0 = \bar{d}^2 I$ used to initialize the algorithm is a valid iterate, since for any point $y \in \mathcal{T}_d$, $\|y\| \le \|d\| \le \bar{d}$, and so $\mathcal{T}_d \subseteq E_{Q^0}$.

Suppose Algorithm WLJ has performed k iterations, and let Q^k be the current iterate. Since $\mathcal{T}_d \subseteq E_{Q^k}$, we conclude that

$$\operatorname{vol}(\mathcal{T}_d) \le \operatorname{vol}(E_{Q^k}) \le \left(\frac{1}{2}e^{\frac{3}{8}}\right)^k \operatorname{vol}(E_{Q^0}) = \left(\frac{1}{2}e^{\frac{3}{8}}\right)^k \bar{d}^m \operatorname{vol}(B(0,1)).$$

On the other hand, since $B(0,\rho(d)) \subseteq \mathcal{T}_d$, we have $\operatorname{vol}(\mathcal{T}_d) \ge \operatorname{vol}(B(0,\rho(d)) = \rho(d)^m \operatorname{vol}(B(0,1))$. Therefore, $\rho(d)^m \operatorname{vol}(B(0,1)) \le \bar{d}^m (\frac{1}{2}e^{\frac{3}{8}})^k \operatorname{vol}(B(0,1))$, and Algorithm WLJ will perform at most

$$K \le m \ln\left(\frac{\bar{d}}{\rho(d)}\right) \cdot \frac{1}{\ln 2 - .375} \le \frac{10}{3} m \ln\left(\frac{\bar{d}}{\rho(d)}\right) \tag{40}$$

iterations.

To bound the skewness of the ellipsoids generated by Algorithm WLJ, note that all such ellipsoids contain the set $\mathcal{T}_d$ and therefore contain $B(0,\rho(d))$. This implies that for any ellipsoid encountered by the algorithm, $\lambda_1 \ge \rho(d)^2$.

We now estimate the change in the largest eigenvalue of the ellipsoid matrix Q^k from one iteration of the algorithm to the next. Suppose Q and $\hat{Q}$ are two consecutive iterates of the algorithm. Then from (38) we conclude that

$$\hat{\lambda}_m = \|\hat{Q}\| \le \|Q\| \frac{m}{m-1}\left(1 - \frac{1}{4m\xi}\right) = \lambda_m \frac{m}{m-1}\left(1 - \frac{1}{4m\xi}\right) \le \lambda_m \frac{m}{m-1} \le \lambda_m e^{\frac{1}{m-1}}.$$

Hence, at any iteration k,

$$\lambda_m^k \le \lambda_m^0 e^{\frac{k}{m-1}} = \bar{d}^2 e^{\frac{k}{m-1}} \le \left(\frac{\bar{d}}{\rho(d)}\right)^{\frac{10m}{3(m-1)}} \bar{d}^2,$$

the last inequality following from (40). Therefore, throughout the algorithm, the skewness of all ellipsoids generated by the algorithm is bounded above by

$$\sqrt{\frac{\lambda_m}{\lambda_1}} \le \sqrt{\left(\frac{\bar{d}}{\rho(d)}\right)^{\frac{10m}{3(m-1)}+2}} \le \left(\frac{\bar{d}}{\rho(d)}\right)^5. \tag{41}$$

Using (41) we conclude from Lemma 32 that any call to subroutine *Weak Check* will perform at most $O(m\sqrt{\vartheta^*}\ln(\frac{m\vartheta^*}{\tau^*} \cdot \frac{\bar{d}}{\rho(d)}))$ iterations of the barrier method. Combining this with (40), we can bound the total number of iterations of the barrier method performed by Algorithm WLJ by

$$O\left(m^2\sqrt{\vartheta^*}\ln^2\left(\frac{\bar{d}}{\rho(d)}\right)\ln\left(\frac{m\vartheta^*}{\tau^*}\right)\right).$$

Finally, the inequalities $\mu_{\mathcal{A}} \le \mathcal{C}(\tilde{B}d) \le \frac{4m\mu_{\mathcal{A}}}{\delta}$ follow from Theorem 18, (28), and Lemma 29. ∎

REMARK 35. *Note that the skewness of the ellipsoids does not necessarily degrade at every iteration. In fact, the last ellipsoid of the algorithm has the nice property that* $\sqrt{\frac{\lambda_m}{\lambda_1}} \le 4\sqrt{m}\mathcal{C}(d)$.

To see why this remark is true, notice that the axes of any ellipsoid of the algorithm will satisfy $\|v_i\| \geq \rho(d)$ for all i, and so $\sqrt{\lambda_1} \geq \rho(d)$. Also, the last ellipsoid of the algorithm satisfies $\frac{1}{4\sqrt{m}} v_i \in \mathcal{T}_d \subset B(0, \|d\|)$ for all i, and so $\|v_i\| \leq 4\sqrt{m}\|d\|$, whereby $\sqrt{\lambda_m} \leq \sqrt{m}\|d\|$.

To further interpret the complexity result of Theorem 34, suppose for simplicity that $\bar{d} = \|d\|$, i.e., the size of the data $\|d\|$ is known. Then Algorithm WLJ will perform at most

$$O\left(m^2\sqrt{\vartheta^*}\ln^2\left(\mathcal{C}(d)\right)\ln\left(\frac{m\vartheta^*}{\tau^*}\right)\right)$$

iterations. We see that the condition number $\mathcal{C}(d)$ of the initial data instance d plays a crucial role in the complexity of Algorithm WLJ, which aims to find an equivalent data instance whose condition number is within a given factor of the best possible. In particular, if the original data instance d is badly conditioned, i.e., $\mathcal{C}(d)$ is large, it might take a large number of iterations to find a "good" preconditioner as above. Another interesting observation is that the complexity of Algorithm WLJ depends on $\mathcal{C}(d)$ rather then $\mu_{\mathcal{A}}$. This result, which may seem counterintuitive at first, is actually explained by the fact that in order to obtain a preconditioner, Algorithm WLJ has to work with the set $\mathcal{T}_d$, rather then $\mathcal{H}_d$, which is symmetric about 0 regardless of the geometry of $\mathcal{H}_d$.

6. Conclusions. In this paper we have addressed several issues related to measures of conditioning for convex feasibility problems. We have discussed some potential drawbacks of using the condition number $\mathcal{C}(d)$ as the sole measure of conditioning of a conic linear system, motivating the study of data-independent measures. We have introduced the symmetry measure $\mu_{\mathcal{A}}$ for feasible conic linear systems as one such data-independent measure, and we have studied many of its implications for problem geometry, conditioning, and algorithm complexity.

One research topic that is not addressed in this paper concerns the existence of data-independent measures of conditioning for (FP_d) that are useful when (FP_d) is infeasible and/or whether any such measures can be adapted to analyze the linear optimization version of (FP_d). Such measures might or might not be an extension of the symmetry measure discussed in this paper.

Another potential topic of research stems from the importance of the inherent conditioning of the problem data for certain properties of (FP_d) such as sensitivity to data perturbations and numerical precision required for accurate computation in algorithms. The complexity bound for computing the good preconditioner in Algorithm WLJ is only reassuring in theory, as it would be unthinkable to use this algorithm in practice. Instead, much as in the case for linear optimization, it would be interesting to explore heuristic methods for preconditioning (FP_d). The notion of a heuristic preconditioning/preprocessing stage in an algorithm is well-established; most optimization software packages include some type of preprocessing options, such as variable and constraint elimination or data scaling, for improving condition numbers and other numerical measures in matrix computations. We hope that the results in this paper may inspire future research on the analysis of heuristic preconditioning techniques for solving linear and conic optimization problems.

Appendix. Implementation of subroutine *Weak Check*. In this appendix we present an implementation of the subroutine *Weak Check*. Recall that each iteration of Algorithm WLJ calls the subroutine *Weak Check* with input being the axes

$v_1, \ldots, v_m$ of an ellipsoid $E_Q \supseteq \mathcal{T}_d$. The purpose of *Weak Check* is to verify whether the rescaled axes $\pm\frac{1}{4\sqrt{m}}v_i$ are elements of $\mathcal{T}_d$, for $i = 1, \ldots, m$, and if not, to produce a parallel cut vector s satisfying (33).

Consider the following $2m$ optimization problems (P_{ϕ_v}), where $v = \pm v_i,\ i = 1, \ldots, m$:

$$
\begin{array}{lllll}
(\mathrm{P}_{\phi_v}) & \phi_v = \max_{\phi} & \phi & = & \max_{\theta,x,\phi} \quad \phi \\
& \text{s.t.} & \phi v \in \mathcal{H}_d & & \text{s.t.} \quad b\theta - Ax = v\phi, \\
& & & & \quad |\theta| + \|x\| \le 1, \\
& & & & \quad \theta \ge 0,\ x \in C_X.
\end{array} \tag{42}
$$

It is easy to verify that $\pm\frac{1}{4\sqrt{m}}v_i \in \mathcal{T}_d$ for all $i = 1, \ldots, m$ precisely when

$$
\phi_Q \stackrel{\triangle}{=} \min_{\pm v_i} \phi_v \ge \frac{1}{4\sqrt{m}}. \tag{43}
$$

(Here $\min_{\pm v_i} \phi_v$ stands for $\min\{\phi_{v_1}, -\phi_{v_1}, \ldots, \phi_{v_m}, -\phi_{v_m}\}$.) We will therefore implement the subroutine *Weak Check* by means of approximately solving the $2m$ optimization problems (42) and checking whether condition (43) is satisfied.

The approach we use to solve the optimization problems (42) in the subroutine *Weak Check* relies on the barrier method described in section 3. Since no obvious starting point is available for (42), we solve (42) for all $2m$ values of $v = \pm v_i,\ i = 1, \ldots, m$, by considering its dual:

$$
\begin{array}{llll}
(\mathrm{P}_{\gamma_v}) & \gamma_v = & \min_{s,q,\gamma} & \gamma \\
& & \text{s.t.} & \|A^t s - q\| \le \gamma, \\
& & & b^t s \le \gamma, \\
& & & q \in C_X^*, \\
& & & v^t s = 1.
\end{array} \tag{44}
$$

It is straightforward to verify that strong duality holds for (P_{ϕ_v}) and (P_{γ_v}), and so

$$
\phi_Q = \min_{\pm v_i} \phi_v = \min_{\pm v_i} \gamma_v.
$$

In order to be able to apply the barrier method, we need the optimization problem at hand to have a bounded feasible region. To satisfy this condition, we consider the following modification of (44):

$$
\begin{array}{llll}
(\mathrm{P}_{\tilde{\gamma}_v}) & \tilde{\gamma}_v = & \min_{s,q,\gamma} & \gamma \\
& & \text{s.t.} & \|A^t s - q\| \le \gamma, \\
& & & b^t s \le \gamma, \\
& & & \|V^t s\| \le 2\sqrt{m}, \\
& & & \gamma \le \frac{7\sqrt{m}\bar{d}}{\sqrt{\lambda_1}}, \\
& & & q \in C_X^*, \\
& & & v^t s = 1,
\end{array} \tag{45}
$$

where $\bar{d}$ is the known upper bound on the norm of the data $\|d\|$, and $V = [v_1, \ldots, v_m] \in \Re^{m\times m}$. The following two simple facts are useful in the derivation of the forthcoming results. First, for all $i = 1, \ldots, m$, we have $\sqrt{\lambda_1} \le \|v_i\| \le \sqrt{\lambda_m}$. Second, for any vector $s \in Y^*$, $\|s\| \le \frac{\|V^t s\|}{\sqrt{\lambda_1}}$. In the next proposition we show that solving $(\mathrm{P}_{\tilde{\gamma}_v})$ instead of (P_{γ_v}) still yields a valid estimate of ϕ_Q.

PROPOSITION 36. *For any v, $\gamma_v \le \tilde{\gamma}_v$. Moreover,*

$$\phi_Q = \min_{\pm v_i} \gamma_v = \min_{\pm v_i} \tilde{\gamma}_v. \tag{46}$$

Proof. The first claim of the proposition is trivially true, since the feasible region of the program $(\mathrm{P}_{\tilde{\gamma}_v})$ is contained in the feasible region of the program (P_{γ_v}).

To establish the second claim, note that

$$\phi_Q = \min_{\pm v_i} \gamma_{\pm v_i} \le \min_{\pm v_i} \tilde{\gamma}_{\pm v_i}.$$

Suppose the minimum on the left is attained for $v = v_{i_0}$, and let $(\bar{s}, \bar{q}, \gamma_v)$ be an optimal solution of the corresponding program (P_{γ_v}). Then we have

$$\gamma_v = \max\{\|A^t\bar{s} - \bar{q}\|, b^t\bar{s}\}, \quad \bar{q} \in C_X^*, \quad v^t\bar{s} = 1.$$

We can further assume without loss of generality that $\|A^t\bar{s} - \bar{q}\| \le \|A^t\bar{s}\|$, since $\bar{q}$ can always be chosen to minimize the distance from $A^t\bar{s}$ to the cone C_X^*. If the point $(\bar{s}, \bar{q}, \gamma_v)$ is feasible for the corresponding program $(\mathrm{P}_{\tilde{\gamma}_v})$, then $\gamma_v = \tilde{\gamma}_v$ and (46) follows. Otherwise, let $\sigma = \max_i |v_i^t\bar{s}| \ge 1$. We can assume without loss of generality that $\sigma = v_j^t\bar{s}$ for some j. (If $v_j^t\bar{s} < 0$, we can redefine the jth axis of E_Q to be $-v_j$.) Define $(\tilde{s}, \tilde{q}, \tilde{\gamma}) = (\frac{1}{\sigma}\bar{s}, \frac{1}{\sigma}\bar{q}, \frac{1}{\sigma}\gamma_v)$. Note that $v_j^t\tilde{s} = 1$, $\tilde{q} \in C_X^*$, and

$$\|V^t\tilde{s}\| = \sqrt{\sum_{i=1}^{m}(v_i^t\tilde{s})^2} \le \sqrt{m} \le 2\sqrt{m}.$$

It remains to check whether the upper bound constraint on $\tilde{\gamma}$ is satisfied. Observe that $\|\tilde{s}\| \le \frac{\sqrt{m}}{\sqrt{\lambda_1}}$ (since $\|V^t\tilde{s}\| \le \sqrt{m}$). Therefore

$$\tilde{\gamma} = \max\{\|A^t\tilde{s} - \tilde{q}\|, b^t\tilde{s}\} \le \max\{\|A^t\tilde{s}\|, b^t\tilde{s}\} \le \bar{d} \cdot \frac{\sqrt{m}}{\sqrt{\lambda_1}} < \frac{7\sqrt{m}\bar{d}}{\sqrt{\lambda_1}}.$$

Hence the vector $(\tilde{s}, \tilde{q}, \tilde{\gamma})$ is feasible for $(\mathrm{P}_{\tilde{\gamma}_{v_j}})$, and $\tilde{\gamma}_{v_j} \le \tilde{\gamma} \le \gamma_v \le \gamma_{v_j} \le \tilde{\gamma}_{v_j}$, which implies that $\tilde{\gamma}_{v_j} = \gamma_v$, from which (46) follows. ∎

Now define

$$S \triangleq \left\{(s, q, \gamma) : \|A^t s - q\| \le \gamma,\ b^t s \le \gamma,\ \|V^t s\| \le 2\sqrt{m},\ \gamma \le \frac{7\sqrt{m}\bar{d}}{\sqrt{\lambda_1}},\ q \in C_X^*\right\}$$

and

$$L_v \triangleq \{(s, q, \gamma) : v^t s = 1\}.$$

Then L_v is a translate of an affine space, and S is a bounded convex set. Recall from the assumptions in section 5.2 that $f^*(\cdot)$ is a self-concordant barrier for the cone C_X^* with complexity parameter ϑ^*. Then the interior of the set S is the domain of the following self-concordant barrier $f(s, q, \gamma)$:

$$f(s, q, \gamma) \triangleq f^*(q) - \ln(\gamma^2 - \|A^t s - q\|^2) - \ln(\gamma - b^t s) - \ln(4m - \|V^t s\|^2) - \ln\left(\frac{7\sqrt{m}\bar{d}}{\sqrt{\lambda_1}} - \gamma\right),$$

whose complexity parameter is $\vartheta_f \le \vartheta^* + 5$.

In order to use the barrier method to solve $(\mathrm{P}_{\tilde{\gamma}_v})$, we need to have a point $(s', q', \gamma') \in \operatorname{int} S \cap L_v$ at which to initialize the method. The next proposition indicates that such point is readily available when the vector $u_* \in C_X^*$ of (7) is known; the second part of the proposition presents a lower bound on $\operatorname{sym}(S \cap L_v, (s', q', \gamma'))$, which is important in analyzing the complexity of the barrier method.

PROPOSITION 37.

$$(s', q', \gamma') \triangleq \left(\frac{v}{\|v\|^2}, \frac{2\bar{d}u_*}{\|v\|}, \frac{4\sqrt{m}\bar{d}}{\sqrt{\lambda_1}} \right) \in \operatorname{int} S \cap L_v,$$

and

$$\operatorname{sym}(S \cap L_v, (s', q', \gamma')) \geq \frac{\tau^*}{13\sqrt{m}} \cdot \sqrt{\frac{\lambda_1}{\lambda_m}}.$$

Proof. The first claim of the proposition is easily established by verifying directly that (s', q', γ') strictly satisfies the constraints of (45). The derivation of the bound on the symmetry in the second claim is fairly long and tedious, and is omitted. We refer the interested reader to [3] for details. □

We now present the formal statement of the implementation of the subroutine *Weak Check*.

Subroutine Weak Check.

- Input: Axes $v_i,\ i = 1, \ldots, m$, of an ellipsoid $E_Q \supseteq \mathcal{T}_d$.
- for $v = \pm v_i,\ i = 1, \ldots, m$,
  Step 1 Form the problem $(\mathrm{P}_{\tilde{\gamma}_v})$.
  Step 2 Run the barrier method on the problem $(\mathrm{P}_{\tilde{\gamma}_v})$ initialized at the point

$$(s', q', \gamma') = \left(\frac{v}{\|v\|^2}, \frac{2\bar{d}u_*}{\|v\|}, \frac{4\sqrt{m}\bar{d}}{\sqrt{\lambda_1}} \right)$$

  until the value of the barrier parameter η first exceeds $\bar{\eta} = \frac{24\sqrt{m}\vartheta_f}{5}$. Let (s, q, γ) be the last iterate of the barrier method.
  Step 3 If $\gamma < \frac{1}{2\sqrt{m}}$, terminate and return s. Otherwise, continue with the next value of v.
- Assert that $\frac{1}{4\sqrt{m}} v_i \in \mathcal{T}_d$ for all $i = 1, \ldots, m$.

Proof of Lemma 32. Subroutine *Weak Check* will apply the barrier method to at most $2m$ problems of the form $(\mathrm{P}_{\tilde{\gamma}_v})$. Note that

$$\min_{(s,q,\gamma) \in S \cap L_v} \gamma \geq 0 \quad \text{and} \quad \max_{(s,q,\gamma) \in S \cap L_v} \gamma \leq \frac{7\sqrt{m}\bar{d}}{\sqrt{\lambda_1}}.$$

Therefore, applying (15) and Proposition 37, we see that each of the (at most) $2m$ applications of the barrier method will terminate in at most

$$\begin{aligned} & O\left(\sqrt{\vartheta_f} \ln \left(\frac{7\sqrt{m}\bar{d}\vartheta_f}{\sqrt{\lambda_1}} \cdot \frac{\bar{\eta}}{\operatorname{sym}(S \cap L_v, (s', q', \gamma'))} \right) \right) \\ \leq & O\left(\sqrt{\vartheta^*} \ln \left(\frac{7\sqrt{m}\bar{d}\vartheta^*}{\sqrt{\lambda_1}} \cdot \frac{24\sqrt{m}\vartheta^*}{5} \cdot \frac{13\sqrt{m}}{\tau^*} \cdot \sqrt{\frac{\lambda_m}{\lambda_1}} \right) \right) \\ = & O\left(\sqrt{\vartheta^*} \ln \left(\frac{m\vartheta^*}{\tau^*} \cdot \frac{\bar{d}}{\sqrt{\lambda_1}} \cdot \sqrt{\frac{\lambda_m}{\lambda_1}} \right) \right) \end{aligned}$$

iterations, giving (36).

Suppose subroutine *Weak Check* has terminated in Step 3 of an iteration in which the barrier method is applied to the problem $(\mathrm{P}_{\tilde{\gamma}_{v_j}})$. (This is without loss of generality; if the termination occurs during the iteration that applies the barrier method to the problem $(\mathrm{P}_{\tilde{\gamma}_{-v_j}})$, we can redefine the jth axis of E_Q to be $-v_j$, to preserve the notation.) Then the last iterate (s, q, γ) of the barrier method satisfies

$$\begin{gathered}\|A^t s - q\| \leq \gamma < \tfrac{1}{2\sqrt{m}},\\ b^t s \leq \gamma < \tfrac{1}{2\sqrt{m}},\\ \|V^t s\| \leq 2\sqrt{m},\\ q \in C_X^*,\ v_j^t s = 1.\end{gathered}$$

The vector s above yields a parallel cut that separates $\pm\frac{v_j}{2\sqrt{m}}$ from $\mathcal{T}_d$. To see why this is true, let $h \in \mathcal{T}_d$. Then $h \in \mathcal{H}_d$, and hence $h = b\theta - Ax$ for some $(\theta, x) \in \Re_+ \times C_X$ such that $|\theta| + \|x\| \leq 1$. Therefore

$$\begin{aligned}s^t h = s^t(b\theta - Ax) &= \theta(b^t s) - x^t(A^t s) = \theta(b^t s) - x^t(A^t s - q) - x^t q\\ &\leq (|\theta| + \|x\|)\gamma \leq \gamma < \frac{1}{2\sqrt{m}} = \frac{s^t v_j}{2\sqrt{m}}.\end{aligned}$$

Applying the same argument for the point $-h \in \mathcal{H}_d$, we conclude that $s^t h > -\frac{s^t v_j}{2\sqrt{m}}$, and therefore the vector s returned by the subroutine *Weak Check* satisfies (37).

Next, suppose that the barrier method applied to $(\mathrm{P}_{\tilde{\gamma}_v})$ has not terminated in Step 3 of the subroutine *Weak Check*, i.e., we have $\gamma \geq \frac{1}{2\sqrt{m}}$. Then, using (14),

$$\tilde{\gamma}_v \geq \gamma - \frac{6\vartheta_f}{5\bar{\eta}} \geq \frac{1}{2\sqrt{m}} - \frac{6\vartheta_f}{5\bar{\eta}} \geq \frac{1}{4\sqrt{m}}.$$

Therefore, if the subroutine *Weak Check* has not terminated in Step 3 for any $v = \pm v_i,\ i = 1, \ldots, m$, we conclude that $\phi_Q = \min_{\pm v_i} \tilde{\gamma}_v \geq \frac{1}{4\sqrt{m}}$, and we correctly assert that $\pm\frac{1}{4\sqrt{m}} v_i \in \mathcal{T}_d$ for all $i = 1, \ldots, m$. ∎

Acknowledgments. The authors would like to thank James Renegar and two anonymous referees for their valuable comments and suggestions.

REFERENCES

[1] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming, Theory and Algorithms*, 2nd ed., John Wiley and Sons, New York, 1993.

[2] R. Bland, D. Goldfarb, and M. J. Todd, *The ellipsoid method: A survey*, Oper. Res., 29 (1981), pp. 1039–1091.

[3] M. Epelman, *Complexity, Condition Numbers, and Conic Linear Systems*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 1999.

[4] M. Epelman and R. M. Freund, *Condition number complexity of an elementary algorithm for computing a reliable solution of a conic linear system*, Math. Program., 88 (2000), pp. 451–486.

[5] S. Filipowski, *On the complexity of solving sparse symmetric linear programs specified with approximate data*, Math. Oper. Res., 22 (1997), pp. 769–792.

[6] S. Filipowski, *On the complexity of solving feasible linear programs specified with approximate data*, SIAM J. Optim., 9 (1999), pp. 1010–1040.

[7] R. M. Freund and J. R. Vera, *On the complexity of computing estimates of condition measures of a conic linear system*, Technical report, Operations Research Center, MIT, Cambridge, MA, 1999.

[8] R. M. Freund and J. R. Vera, *Some characterizations and properties of the "distance to ill-posedness" and the condition measure of a conic linear system*, Math. Program., 86 (1999), pp. 225–260.

[9] R. M. Freund and J. R. Vera, *Condition-based complexity of convex optimization in conic linear form via the ellipsoid algorithm*, SIAM J. Optim., 10 (2000), pp. 155–176.
[10] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, New York, 1994.
[11] J. C. K. Ho, *Structure of the central path and related complexity issues for linear programming*, Master's thesis, University of Waterloo, Waterloo, ON, Canada, 1999.
[12] F. John, *Extremum problems with inequalities as subsidiary conditions*, in Studies and Essays, Presented to R. Courant on His 60th Birthday, Interscience, New York, 1948, pp. 187–204.
[13] N. Karmarkar, *A new polynomial-time algorithm for linear programming*, Combinatorica, 4 (1984), pp. 373–395.
[14] L. G. Khachiyan, *A polynomial algorithm in linear programming*, Soviet Math. Dokl., 20 (1979), pp. 191–194.
[15] V. Klee and G. J. Minty, *How good is the simplex algorithm?*, in Inequalities, O. Shisha, ed., Academic Press, New York, 1972, pp. 159–175.
[16] N. Megiddo, S. Mizuno, and T. Tsuchiya, *A modified layered-step interior-point algorithm for linear programming*, Math. Programming, 82 (1998), pp. 339–355.
[17] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM, Philadelphia, 1994.
[18] M. A. Nunez, *A Characterization of Ill-Posed Data Instances for Convex Programming*, informal paper, 1998.
[19] M. A. Nunez and R. M. Freund, *Condition measures and properties of the central trajectory of a linear program*, Math. Programming, 83 (1998), pp. 1–28.
[20] J. Peña, *Computing the Distance to Infeasibility: Theoretical and Practical Issues*, Technical report, Cornell University, Ithaca, NY, 1997.
[21] J. Peña, *Understanding the geometry of infeasible perturbations of a conic linear system*, SIAM J. Optim., 10 (2000), pp. 534–550.
[22] J. Peña and J. Renegar, *Computing approximate solutions for conic systems of constraints*, Math. Program., 87 (2000), pp. 351–383.
[23] J. Renegar, *A polynomial-time algorithm, based on Newnton's method, for linear programming*, Math. Programming, 40 (1988), pp. 59–93.
[24] J. Renegar, *Some perturbation theory for linear programming*, Math. Programming, 65 (1994), pp. 73–91.
[25] J. Renegar, *Incorporating condition measures into the complexity theory of linear programming*, SIAM J. Optim., 5 (1995), pp. 506–524.
[26] J. Renegar, *Linear programming, complexity theory, and elementary functional analysis*, Math. Programming, 70 (1995), pp. 279–351.
[27] J. Renegar, *A Mathematical View of Interior-Point Methods for Convex Optimization*, SIAM, Philadelphia, 2001.
[28] S. M. Robinson, *Bounds for error in the solution set of a perturbed linear program*, Linear Algebra Appl., 6 (1973), pp. 69–81.
[29] M. J. Todd, L. Tunçel, and Y. Ye, *Characterizations, bounds, and probabilistic analysis of two complexity measures for linear programming problems*, Math. Program., 90 (2001), pp. 59–69.
[30] L. Tunçel, *Approximating the complexity measure of Vavasis-Ye algorithm is NP-hard*, Math. Program., 86 (1999), pp. 219–223.
[31] L. Tunçel, *On the condition numbers for polyhedra in Karmarkar's form*, Oper. Res. Lett., 24 (1999), pp. 149–155.
[32] S. A. Vavasis and Y. Ye, *Condition numbers for polyhedra with real number data*, Oper. Res. Lett., 17 (1995), pp. 209–214.
[33] S. A. Vavasis and Y. Ye, *A primal-dual interior point method whose running time depends only on the constraint matrix*, Math. Programming, 74 (1996), pp. 79–120.
[34] S. A. Vavasis and Y. Ye, *A simplification to "a primal-dual interior point method whose running time depends only on the constraint matrix,"* in High Performance Optimization, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000, pp. 233–243.
[35] J. R. Vera, *Ill-Posedness and the Computation of Solutions to Linear Programs with Approximate Data*, Technical report, Cornell University, Ithaca, NY, 1992.
[36] J. R. Vera, *Ill-Posedness in Mathematical Programming and Problem Solving with Approximate Data*, Ph.D. thesis, Cornell University, Ithaca, NY, 1992.
[37] J. R. Vera, *Ill-posedness and the complexity of deciding existence of solutions to linear programs*, SIAM J. Optim., 6 (1996), pp. 549–569.
[38] J. R. Vera, *On the complexity of linear programming under finite precision arithmetic*, Math. Programming, 80 (1998), pp. 91–123.
[39] S. J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, PA, 1997.

SIAM J. OPTIM.
Vol. 12, No. 3, pp. 656–668

A FUZZY NECESSARY OPTIMALITY CONDITION FOR NON-LIPSCHITZ OPTIMIZATION IN ASPLUND SPACES*

HUYNH VAN NGAI† AND MICHEL THÉRA‡

Abstract. In this paper, we consider a general optimization problem in the broad class of Asplund spaces. We derive a new necessary optimality condition in the so-called Lagrangian "fuzzy form" without standard Lipschitz conditions. We also give a chain rule for Fréchet subdifferentials and subdifferential representations of Fréchet normals to constrained sets.

Key words. constrained optimization problem, necessary optimality condition, fuzzy calculus, Fréchet subdifferential, normal cone, optimality condition

AMS subject classifications. 49K27, 49J52, 90C30

PII. S1052623400366656

1. Introduction. Throughout the paper, X denotes a Banach space, $C \subset X$ is a closed subset of X and $f_i : X \to \mathbb{R} \cup \{+\infty\}$, $i = 0, 1, \ldots, n$, are extended real valued functions. We consider a general constrained optimization problem with semicontinuous inequality and continuous equality data:

$$(\mathfrak{P}) \qquad \begin{array}{ll} \text{minimize} & f_0(x) \quad \text{subject to (s.t.)} \\ & f_i(x) \le 0, \quad i = 1, \ldots, m, \\ & f_i(x) = 0, \quad i = m+1, \ldots, n, \\ & x \in C. \end{array}$$

Although problem $(\mathfrak{P})$ for nonsmooth data has been studied by several authors including, for instance, [1, 2, 6, 7, 11, 15] and the references therein, the current research was motivated by the work of [1, 11, 18, 19, 22]. In these papers, necessary optimality conditions were established in different situations. In [1], Borwein, Treiman, and Zhu considered reflexive Banach spaces, and their necessary optimality condition was given in terms of (smooth) subderivatives and normal cones. In [11], Kruger and Mordukhovich gave a necessary condition in terms of limiting constructions in spaces with Fréchet renorms, while Mordukhovich [18] and Mordukhovich and Wang [19] considered Asplund spaces. Finally, this research continues work by Ngai and Théra, who proved in [22] a necessary condition using limiting Fréchet subdifferentials and limiting normals in Asplund spaces with compactness assumptions.

It is a purpose of this report to prove a fuzzy multiplier rule for the above problem in terms of Fréchet subdifferentials and Fréchet normals in Asplund spaces, without standard Lipschitz conditions.

In section 2, a "fuzzy calculus rule" for Fréchet subdifferentials of composite functions is established. Using this chain rule, we derive in section 3 a fuzzy multiplier rule for problem $(\mathfrak{P})$. In the final section we present in the Asplund setting a result on the relationship between the normal cone to a level set and the subdifferential of the

*Received by the editors January 24, 2000; accepted for publication (in revised form) July 9, 2001; published electronically January 9, 2002.

http://www.siam.org/journals/siopt/12-3/36665.html

†Ecole Normale Supérieure de Quy Nhon, Department of Mathematics and Informatics, DHSP, Quy Nhon, Vietnam (ngaivn@yahoo.com). This author was supported in part by the PICS-CNRS *"Formath Vietnam"* and by a *"bourse de co-tutelle."*

‡LACO, UMR 6090, Université de Limoges, 123 avenue Albert Thomas, 87060 Limoges Cedex, France (michel.thera@unilim.fr).

corresponding function. Such a relationship for smooth subderivatives and smooth normal cones in reflexive spaces plays a key role in the proof of the main result in [1].

2. Fuzzy calculus for Fréchet subdifferentials. Let X be a Banach space with closed unit ball B_X and with dual space X^*. Let $f : X \to \mathbb{R} \cup \{+\infty\}$ be a lower semicontinuous function. We denote as usual by $\mathrm{dom} f := \{x \in X \ : \ f(x) < +\infty\}$, $\mathrm{epi} f := \{(x, \alpha) \in X \times \mathbb{R} \ : \ \alpha \geq f(x)\}$, and $\mathrm{gph} f := \{(x, \alpha) \in X \times \mathbb{R} \ : \ \alpha = f(x)\}$ the *domain*, the *epigraph*, and the *graph* of f, respectively.

Recall that the *Fréchet subdifferential* of f at $x \in \mathrm{dom} f$ is defined by

$$\partial^F f(x) := \left\{ x^* \in X^* : \liminf_{h \to 0} \frac{f(x+h) - f(x) - \langle x^*, h \rangle}{\|h\|} \geq 0 \right\}. \tag{2.1}$$

If $x \notin \mathrm{dom} f$, we set $\partial^F f(x) := \emptyset$.

For a closed subset C of X, the *Fréchet normal cone* to C at $x \in C$ is the set $N^F(C, x) := \partial^F \delta_C(x)$, where $\delta_C(.)$ is the *indicator function* of the set C and is given by

$$\delta_C(x) := \begin{cases} 0 & \text{if } x \in C, \\ +\infty & \text{if } x \notin C. \end{cases}$$

Equivalently,

$$N^F(C, x) = \left\{ x^* \in X^* \ : \ \limsup_{u \to x, \ u \in C} \frac{\langle x^*, u - x \rangle}{\|u - x\|} \leq 0 \right\}.$$

The Fréchet subdifferential has a geometrical interpretation in terms of the Fréchet normal cone to the epigraph of the function under consideration:

$$\partial^F f(x) = \left\{ x^* \in X^* \ : \ (x^*, -1) \in N^F(\mathrm{epi}\, f, (x, f(x))) \right\}.$$

Recall that a Banach space is said to be *Asplund* if every convex continuous function is Fréchet differentiable on a dense G_δ-subset of the interior of its domain. This class of Banach spaces includes Banach spaces with Fréchet differentiable renorms or bump functions (hence, all reflexive spaces; see [23]). A important characterization of Asplund spaces is the *fuzzy sum rule* for Fréchet subdifferentials proved by Fabian [4, 5] (see also other characterizations in Modukhovich and Shao [16] and Jourani and Théra [9]).

PROPOSITION 2.1 (see Fabian [5]). *Let X be an Asplund space, let $f_i : X \to \mathbb{R} \cup \{+\infty\}$, $i = 1, \ldots, n$, be lower semicontinuous functions. Let $\bar{x} \in \mathrm{dom} f_1 \cap \cdots \cap \mathrm{dom} f_n$. Then, for any $\epsilon > 0$ and any weak* neighborhood V of 0 in X^*,*

$$\partial^F (f_1 + \cdots + f_n)(\bar{x}) \subseteq \bigcup \big\{ \partial^F f_1(x_1) + \cdots + \partial^F f_n(x_n) + V : \\ (x_i, f_i(x_i)) \in (\bar{x}, f_i(\bar{x})) + \epsilon B_{X \times \mathbb{R}}, \ i = 1, \ldots, n \big\}.$$

Moreover, in addition, if $f_2, \ldots, f_n$ are locally Lipschitz, then the following inclusion holds:

$$\partial^F (f_1 + \cdots + f_n)(\bar{x}) \subseteq \bigcup \big\{ \partial^F f_1(x_1) + \cdots + \partial^F f_n(x_n) + \epsilon B_{X^*} : \\ (x_i, f_i(x_i)) \in (\bar{x}, f_i(\bar{x})) + \epsilon B_{X \times \mathbb{R}}, \ i = 1, \ldots, n \big\}.$$

In what follows we shall make use of the following pair of lemmata from [8, 12, 17, 22].

LEMMA 2.2 (see Ioffe [8], Modukhovich and Shao [17]). *Let $f : X \to \mathbb{R} \cup \{+\infty\}$ be a lower semicontinuous function and let $(\bar{x}, \alpha) \in \mathrm{epi} f$.*

(i) *Let $\lambda \neq 0$; the equivalence*

$$(x^*, -\lambda) \in N^F(\mathrm{epi} f, (\bar{x}, \alpha)) \iff \lambda > 0,\ \alpha = f(\bar{x}),\ x^* \in \partial^F(\lambda f)(\bar{x})$$

is true in every Banach space X.

(ii) *Suppose that X is an Asplund space. If $(x^*, 0) \in N^F(\mathrm{epi} f, (\bar{x}, \alpha))$, then there exist sequences $\{x_n\}_{n\in\mathbb{N}}$, $\{x_n^*\}_{n\in\mathbb{N}}$, $\{\lambda_n\}_{n\in\mathbb{N}}$ such that*

$$\begin{gathered} x_n^* \in \lambda_n \partial^F f(x_n), \\ (x_n, f(x_n)) \to (\bar{x}, f(\bar{x})),\ \lambda_n \downarrow 0, \\ \text{and } \|x_n^* - x^*\| \to 0. \end{gathered}$$

LEMMA 2.3 (see [12, 22]). *Let $f : X \to \mathbb{R} \cup \{+\infty\}$ be a continuous function and let $\bar{x} \in \mathrm{dom} f$.*

(i) *Let $\lambda \neq 0$; the equivalence*

$$(x^*, -\lambda) \in N^F(\mathrm{gph} f, (\bar{x}, f(\bar{x}))) \iff x^* \in \partial^F(\lambda f)(\bar{x})$$

is true in any Banach space X.

(ii) *Suppose that X is an Asplund space. If $(x^*, 0) \in N^F(\mathrm{gph} f, (\bar{x}, f(\bar{x})))$, then there exist sequences $\{x_n\}_{n\in\mathbb{N}}$, $\{x_n^*\}_{n\in\mathbb{N}}$, $\{\lambda_n\}_{n\in\mathbb{N}}$ such that*

$$\begin{gathered} x_n^* \in \partial^F(\lambda_n f)(x_n) \cup \partial^F(-\lambda_n f)(x_n), \\ (x_n, f(x_n)) \to (\bar{x}, f(\bar{x})),\ \ \lambda_n \downarrow 0, \\ \text{and } \|x_n^* - x^*\| \to 0. \end{gathered}$$

Let $f_i : X \to \mathbb{R} \cup \{+\infty\}$, $i = 1, \ldots, n$, $g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$. Consider the composite function $g[f_1, \ldots, f_n] : X \to \mathbb{R} \cup \{+\infty\}$ defined by

$$g[f_1, \ldots, f_n](x) := \begin{cases} g(f_1(x), \ldots, f_n(x)) & \text{if} \quad x \in \mathrm{dom} f_1 \cap \cdots \cap \mathrm{dom} f_n, \\ +\infty & \text{otherwise.} \end{cases}$$

We next prove the following chain rule.

THEOREM 2.4. *Let X be an Asplund space. Let $f_i : X \to \mathbb{R} \cup \{+\infty\}$ be lower semicontinuous for $i = 1, \ldots, m$ and be continuous for $i = m+1, \ldots, n$. Let $g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be lower semicontinuous such that $\lim_{|t|\to+\infty} g(t) = +\infty$. Suppose that g is nondecreasing for each of its first m variables. Let $\bar{x} \in \mathrm{dom} f_1 \cap \cdots \cap \mathrm{dom} f_n$ and $(f_1(\bar{x}), \ldots, f_n(\bar{x})) \in \mathrm{dom} g$. Then, for any $\epsilon > 0$ and any weak* neighborhood V of 0 in X^*, we have*

$$\begin{aligned} \partial^F g[f_1, \ldots, f_n](\bar{x}) \subseteq \bigcup \Big\{ & \partial^F(\mu_1 f_1)(x_1) + \cdots + \partial^F(\mu_n f_n)(x_n) + V : \\ & (x_i, f_i(x_i)) \in (\bar{x}, f_i(\bar{x})) + \epsilon B_{X\times\mathbb{R}},\ i = 1, \ldots, n; \\ & (\alpha_1, \ldots, \alpha_n) \in (f_1(\bar{x}), \ldots, f_n(\bar{x})) + \epsilon B_{\mathbb{R}^n}; \\ & (\mu_1, \ldots, \mu_n) \in \partial^F g(\alpha_1, \ldots, \alpha_n) + \epsilon B_{\mathbb{R}^n}; \\ & \mu_i > 0,\ i = 1, \ldots, m;\ \mu_i \neq 0,\ i = m+1, \ldots, n \Big\}. \end{aligned}$$

Proof. First, we set

$$S_i := \Big\{(x, \alpha_1, \dots, \alpha_n) \in X \times \mathbb{R}^n : \alpha_i \geq f_i(x)\Big\}, \qquad i = 1, \dots, m,$$
$$S_i := \Big\{(x, \alpha_1, \dots, \alpha_n) \in X \times \mathbb{R}^n : \alpha_i = f_i(x)\Big\}, \quad i = m+1, \dots, n.$$

Clearly, for $i = 1, \dots, m$,

$$N^F(S_i, (x, \alpha_1, \dots, \alpha_n)) = \Big\{(x^*, \lambda_1, \dots, \lambda_n) \in X^* \times \mathbb{R}^n : \lambda_j = 0 \quad \text{if} \quad j \neq i,$$
$$(x^*, \lambda_i) \in N^F(\text{epi} f_i, (x, \alpha_i))\Big\},$$

while for $i = m+1, \dots, n$,

$$N^F(S_i, (x, \alpha_1, \dots, \alpha_n)) = \Big\{(x^*, \lambda_1, \dots, \lambda_n) \in X^* \times \mathbb{R}^n : \lambda_j = 0 \quad \text{if} \quad j \neq i,$$
$$(x^*, \lambda_i) \in N^F(\text{gph} f_i, (x, \alpha_i))\Big\}.$$

Fix $x^* \in \partial^F g[f_1, \dots, f_n](\bar{x})$. Observe that

$$g[f_1, \dots, f_n](x) = \min\left\{g(\alpha_1, \dots, \alpha_n) + \sum_{i=1}^n \delta_{S_i}(x, \alpha_1, \dots, \alpha_n) : (x, \alpha_1, \dots, \alpha_n) \in X \times \mathbb{R}^n\right\}.$$

Hence,

$$(x^*, 0, \dots, 0) \in \partial^F \left[g(.) + \sum_{i=1}^n \delta_{S_i}(.)\right](\bar{x}, f_1(\bar{x}), \dots, f_n(\bar{x})). \tag{2.2}$$

For any $\epsilon > 0$, any weak* neighborhood V of 0 in X^*, let U be a weak* neighborhood of 0 in X^* such that $U + n\epsilon B_{X^*} \subseteq V$. ($B_{X^*}$ is the closed unit ball in X^*.)

Since the functions f_i are lower semicontinuous, there exists $\eta \in (0, \frac{\epsilon}{2})$ such that

$$f_i(x) > f_i(\bar{x}) - \frac{\epsilon}{2} \qquad \text{for all } x \in \bar{x} + \eta B_X,\ i = 1, \dots, n. \tag{2.3}$$

Using the fuzzy sum rule for (2.2), there exist

$$(x_i, r_i) \in \big((\bar{x}, f_i(\bar{x})) + \eta B_{X \times \mathbb{R}}\big) \cap \text{epi} f_i,\ i = 1, \dots, m, \tag{2.4}$$

$$(x_i, f_i(x_i)) \in (\bar{x}, f_i(\bar{x})) + \eta B_{X \times \mathbb{R}},\ i = m+1, \dots, n, \tag{2.5}$$

$$(\alpha_1, \dots, \alpha_n) \in (f_1(\bar{x}), \dots, f_n(\bar{x})) + \eta B_{\mathbb{R}^n}, \tag{2.6}$$

$$(\lambda_1, \dots, \lambda_n) \in \partial^F g(\alpha_1, \dots, \alpha_n),$$

$$(\zeta_i, -\gamma_i) \in N^F(\text{epi} f_i, (x_i, r_i)),\ i = 1, \dots, m, \tag{2.7}$$

$$(\zeta_i, -\gamma_i) \in N^F(\mathrm{gph} f_i, (x_i, f_i(x_i))),\ i = m+1, \dots, n, \tag{2.8}$$

such that

$$(x^*, 0, \dots, 0) \in (0, \lambda_1, \dots, \lambda_n) + (\zeta_1 + \cdots + \zeta_n, -\gamma_1, \dots, -\gamma_n) + U \times \eta B_{\mathbb{R}^n}.$$

Thus,

$$x^* \in \zeta_1 + \cdots + \zeta_n + U, \tag{2.9}$$

$$(\gamma_1, \dots, \gamma_n) \in \partial^F g(\alpha_1, \dots, \alpha_n) + \eta B_{R^n}. \tag{2.10}$$

For every $i = 1, \dots, m$, by (2.7), using Lemma 2.2, we obtain the following:

• If $\gamma_i \neq 0$, then $r_i = f_i(x_i)$, $\gamma_i > 0$, and $\zeta_i \in \gamma_i \partial^F f_i(x_i)$. Set $\mu_i := \gamma_i$, $z_i := x_i$, and $\xi_i := \zeta_i$.

• Else, $\gamma_i = 0$. Then, there exist $\mu_i \in (0, \eta)$, $(z_i, f_i(z_i)) \in (x_i, f_i(x_i)) + \eta B_{X \times \mathbb{R}}$, $\xi_i \in \mu_i \partial^F f_i(z_i)$ such that $\|\xi_i - \zeta_i\| < \epsilon$.

Similarly, for every $i = m+1, \dots, n$, applying Lemma 2.3 to (2.8), one has the following:

• If $\gamma_i \neq 0$, then $\zeta_i \in \partial^F(\gamma_i f_i)(x_i)$. Set $\mu_i := \gamma_i$, $z_i := x_i$, and $\xi_i := \zeta_i$.

• Else, $\gamma_i = 0$. Then, there exist $\mu_i \in (-\eta, \eta)$, $\mu_i \neq 0$, $(z_i, f_i(z_i)) \in (x_i, f_i(x_i)) + \eta B_{X \times \mathbb{R}}$, $\xi_i \in \partial^F(\mu_i f_i)(z_i)$ such that $\|\xi_i - \zeta_i\| < \epsilon$. Hence, by (2.10) we have

$$(\mu_1, \dots, \mu_n) \in \partial^F g(\alpha_1, \dots, \alpha_n) + \epsilon B_{R^n},$$

and moreover, from (2.9) we derive that

$$x^* \in \zeta_1 + \cdots + \zeta_n + U \subseteq \xi_1 + \cdots + \xi_n + n\epsilon B_{X^*} + U \subseteq \xi_1 + \cdots + \xi_n + V.$$

On the other hand, combining (2.3), (2.4), (2.5) yields

$$(z_i, f_i(z_i)) \in (\bar{x}, f_i(\bar{x})) + \epsilon B_{X \times \mathbb{R}} \qquad \text{for all } i = 1, \dots, n.$$

The proof is complete. □

3. A necessary optimality condition. Let us again consider the constrained optimization problem:

$$(\mathfrak{P}) \qquad \begin{array}{lll} \min & f_0(x) & \text{s.t.} \\ & f_i(x) \leq 0, & i = 1, \dots, m, \\ & f_i(x) = 0, & i = m+1, \dots, n, \\ & x \in C. & \end{array}$$

Suppose that $f_i : X \to \mathbb{R} \cup \{+\infty\}$ is lower semicontinuous for each $i = 1, \dots, m$, continuous for each $i = m+1, \dots, n$, and that $C \subseteq X$ is a nonempty closed set.

We now use the chain rule established in section 2 to obtain a multiplier rule for problem $(\mathfrak{P})$. In the proof, following Treiman [26], we use the function $g : \mathbb{R}^{n+1} \to \mathbb{R}$ given by

$$g(\alpha_0, \dots, \alpha_n) := \begin{cases} \max\{\alpha_0, \dots, \alpha_n\} & \text{if} \quad \alpha_{m+1} = \cdots = \alpha_n = 0, \\ \max\{\alpha_0, \dots, \alpha_m, |\alpha_{m+1}|, \dots, |\alpha_n|\} & \text{otherwise.} \end{cases}$$

We first establish the following simple lemma.

LEMMA 3.1. *For any* $(\alpha_0, \dots, \alpha_n) \in \mathbb{R}^{n+1}$,

$$(\mu_0, \dots, \mu_n) \in \partial^F g(\alpha_0, \dots, \alpha_n) \implies \mu_i \geq 0,\ i = 0, \dots, m, \ \text{and} \ \sum_{i=0}^{n} |\mu_i| \geq 1.$$

Proof. Clearly, g is nondecreasing for each of its first $m+1$ variables. Hence, immediately, $\mu_i \geq 0$ for all $i = 0, \dots, m$. By definition, for each $\epsilon > 0$ there exists $\delta > 0$ such that

$$\sum_{i=0}^{n} \mu_i h_i \leq g(\alpha_0 + h_0, \dots, \alpha_n + h_n) - g(\alpha_0, \dots, \alpha_n) + \epsilon \max\{|h_0|, \dots, |h_n|\} \tag{3.1}$$

$$\text{for all } (h_0, \dots, h_n) \in \delta B_{R^{n+1}}.$$

Let $h \in (0, \delta)$. Take $(h_0, \dots, h_n)$ in (3.1) such that $h_0 = \cdots = h_m = -h$, and for $i = m+1, \dots, n$, $h_i = -h$ if $\alpha_i > 0$; $h_i = h$ if $\alpha_i < 0$; $h_i = 0$ if $\alpha_i = 0$. Then, when δ is small, we have

$$g(\alpha_0 + h_0, \dots, \alpha_n + h_n) - g(\alpha_0, \dots, \alpha_n) = -h.$$

Hence, from (3.1),

$$\sum_{i=0}^{n} |\mu_i| h \geq -\sum_{i=0}^{n} \mu_i h_i \geq h(1 - \epsilon).$$

Dividing the latter inequality by h and letting ϵ go to zero, we obtain $\sum_{i=0}^{n} |\mu_i| \geq 1$, establishing the proof of Lemma 3.1. □

We can now prove the main result, as follows.

THEOREM 3.2. *Let X be an Asplund space, let C be a closed subset of X. Suppose the functions f_i are lower semicontinuous for $i = 0, \dots, m$ and continuous for $i = m+1, \dots, n$. Assume that $\bar{x}$ is a local solution of $(\mathfrak{P})$. Then for any $\epsilon > 0$, any weak* neighborhood V of 0 in X^*, there exist*

$$\begin{aligned} (x_i, f_i(x_i)) &\in (\bar{x}, f_i(\bar{x})) + \epsilon B_{X \times \mathbb{R}}, \quad i = 0, \dots, n, \\ x_{n+1} &\in \bar{x} + \epsilon B_X, \\ \mu_i &> 0, \qquad i = 0, \dots, m, \\ \mu_i &\neq 0, \qquad i = m+1, \dots, n, \end{aligned}$$

such that

$$|\mu_0| + \cdots + |\mu_n| = 1,$$

$$0 \in \partial^F(\mu_0 f_0)(x_0) + \cdots + \partial^F(\mu_n f_n)(x_n) + N^F(C, x_{n+1}) + V.$$

Proof. Observe that if $\bar{x}$ is a local solution of $(\mathfrak{P})$, then $\bar{x}$ is a local minimum point of function $g \circ F + \delta_C$, where $g : \mathbb{R}^{n+1} \to \mathbb{R}$,

$$g(\alpha_0, \dots, \alpha_n) := \begin{cases} \max\{\alpha_0, \dots, \alpha_n\} & \text{if} \quad \alpha_{m+1} = \cdots = \alpha_n = 0, \\ \max\{\alpha_0, \dots, \alpha_m, |\alpha_{m+1}|, \dots, |\alpha_n|\} & \text{otherwise,} \end{cases}$$

and

$$F(x) := (f_0(x) - f_0(\bar{x}), \dots, f_m(x) - f_m(\bar{x}), f_{m+1}(x), \dots, f_n(x)).$$

Therefore,

$$0 \in \partial^F(g \circ F + \delta_C)(\bar{x}). \tag{3.2}$$

Of course, we only need to consider $\epsilon \in (0,1)$. Since the functions f_i are lower semicontinuous, there exists $\eta \in (0, \frac{\epsilon}{2})$ such that

$$f_i(x) > f_i(\bar{x}) - \frac{\epsilon}{2} \qquad \text{for all} \quad x \in \bar{x} + \eta B_X,\ i = 0, \dots, n. \tag{3.3}$$

Let U be a convex weak* neighborhood of 0 in X^* such that $2U \subseteq V$. Applying the fuzzy sum rule to (3.2) yields

$$(y, g \circ F(y)) \in (\bar{x}, g \circ F(\bar{x})) + \eta B_{X \times \mathbb{R}}, \tag{3.4}$$

$$x_{n+1} \in \bar{x} + \eta B_X, \quad \zeta \in \partial^F(g \circ F)(y), \quad \zeta_{n+1} \in N^F(C, x_{n+1})$$

such that

$$0 \in \zeta + \zeta_{n+1} + \frac{U}{2}. \tag{3.5}$$

Combining (3.3) and (3.4), we derive that

$$(y, f_i(y)) \in (\bar{x}, f_i(\bar{x})) + \frac{\epsilon}{2} B_{X \times \mathbb{R}}, \quad i = 0, \dots, n. \tag{3.6}$$

On the other hand, since $\zeta \in \partial^F(g \circ F)(y)$, by virtue of Theorem 2.4, there exist

$$(x_i, f_i(x_i)) \in (y, f_i(y)) + \frac{\epsilon}{2} B_{X \times \mathbb{R}}, \quad i = 0, \dots, n, \tag{3.7}$$

$$(\alpha_0, \dots, \alpha_n) \in (f_0(y), \dots, f_n(y)) + \frac{\epsilon}{2} B_{\mathbb{R}^{n+1}}, \tag{3.8}$$

$$(\lambda_0, \dots, \lambda_n) \in \partial^F g(\alpha_0, \dots, \alpha_n) + \frac{\epsilon}{2} B_{\mathbb{R}^{n+1}}, \tag{3.9}$$

and $\zeta_i \in \partial^F(\lambda_i f_i)(x_i),\ i = 0, \dots, n$, such that

$$\lambda_i > 0 \ \text{ for } \ i = 0, \dots, m, \quad \lambda_i \neq 0 \ \text{ for } \ i = m+1, \dots, n,$$

and

$$\zeta \in \zeta_0 + \cdots + \zeta_n + \frac{U}{2}. \tag{3.10}$$

From (3.5), (3.10), we have

$$0 \in \zeta_0 + \cdots + \zeta_n + \zeta_{n+1} + U. \tag{3.11}$$

By (3.6), (3.7), we have

$$(x_i, f_i(x_i)) \in (\bar{x}, f_i(\bar{x})) + \epsilon B_{X\times\mathbb{R}}. \tag{3.12}$$

By Lemma 3.1 and (3.9), we derive that $|\lambda_0| + \cdots + |\lambda_n| \geq 1 - \frac{\epsilon}{2} > \frac{1}{2}$. Dividing the inclusion (3.11) by $\lambda := \sum_{i=0}^{n} |\lambda_i| > \frac{1}{2}$, and setting

$$\begin{gathered}\mu_i := \lambda_i/\lambda,\\ \xi_i := \zeta_i/\lambda \in \partial^F(\mu_i f_i)(x_i),\ i = 0, \dots, n,\\ \xi_{n+1} = \zeta_{n+1}/\lambda,\end{gathered}$$

we obtain

$$\begin{gathered}\mu_i > 0 \ \text{ for } \ i = 0, \dots, m,\\ \mu_i \neq 0 \ \text{ for } \ i = m+1, \dots, n,\\ \sum_{i=0}^{n} |\mu_i| = 1,\end{gathered}$$

and

$$0 \in \sum_{i=0}^{n} \xi_i + \xi_{n+1} + U/\lambda \subseteq \sum_{i=0}^{n} \partial^F(\mu_i f_i)(x_i) + N^F(C, x_{n+1}) + V.$$

The proof is complete. □

Let us mention that optimality necessary conditions in the "fuzzy" form for optimization problems with non-Lipschitzian data were first obtained by Kruger and Mordukhovich [11] in terms of ϵ-Fréchet normals in Banach spaces with Fréchet differentiable renorms. The above necessary condition was established in reflexive Banach spaces by Borwein, Treiman, and Zhu [1]. Our proof differs from the proof given in [1]. Recently, Mordukhovich [18, Theorem 5.1(i)] established the general version of fuzzy necessary optimality conditions in terms of Fréchet normals in Asplund spaces. Finally, a result close to Theorem 3.2 was derived by a different method in [19, Theorem 6.3].

4. The normal cone to a level set. The relationship between the normal cone to a level set and the subderivative of the corresponding function was established in the reflexive setting [1]. This relation was the key ingredient of the proof of the main result in [1]. This relationship is of some independent interest; however, in the final section, we prove that it is also valid for Fréchet subdifferentials and Fréchet normals in an Asplund setting.

THEOREM 4.1. *Let X be an Asplund space and let $f : X \to \mathbb{R} \cup \{+\infty\}$ be a lower semicontinuous function. Let $S := \{x \in X : f(x) \leq 0\}$, and let $\bar{x} \in \operatorname{bdry} S$ (the boundary of S). Then, either*

(A1) *for any $\epsilon, \eta > 0$, there exists $(x, f(x)) \in (\bar{x}, f(\bar{x})) + \eta B_{X\times\mathbb{R}}$ such that $\partial^F f(x) \cap \epsilon B_{X^*} \neq \emptyset$,*

or

(A2) *for any $\xi \in N^F(S, \bar{x})$, any $\epsilon > 0$, there exist $(x, f(x)) \in (\bar{x}, f(\bar{x})) + \epsilon B_{X\times\mathbb{R}}$, $\zeta \in \partial^F f(x)$, and $\lambda > 0$ such that $\|\lambda\zeta - \xi\| < \epsilon$.*

We first recall the following two lemmata from [10, 22]. Denote by $d_C(.)$ the distance function to a set C.

LEMMA 4.2 (see Jourani and Thibault [10]). *Let C be a nonempty closed subset of X and let $x_0 \notin C$. Then the implication*

$$x^* \in \partial^F d_C(x_0) \Longrightarrow \|x^*\| = 1$$

holds for any Banach space X.

The following lemma is proved in [22]. It follows the lines of Thibault's paper [25].

LEMMA 4.3. *Let $C \subset X$ be a nonempty closed set.*

(i) *Let $\bar{x} \in C$ and let $x^* \in N^F(C, \bar{x})$. Then $x^* \in \lambda \partial^F d_C(\bar{x})$ holds for any $\lambda \geq \|x^*\| + 1$ and any Banach space X.*

(ii) *Let X be an Asplund space and let $x \in X$. If $x^* \in \partial^F d_C(\bar{x})$, then for any $\epsilon \in (0,1)$ there exist $x_\epsilon \in C$ and $x^*_\epsilon \in N^F(C, x_\epsilon)$ such that*

$$\|x_\epsilon - \bar{x}\| < d_C(\bar{x}) + \epsilon \quad \text{and} \quad \|x^*_\epsilon - x^*\| \leq \epsilon.$$

Proof of Theorem 4.1. We set $S_1 := \big\{(x,\alpha) \in X \times \mathbb{R} : \alpha \leq 0\big\}$, $S_2 := \text{epi} f$. Clearly,

$$\partial^F d_{S_1}(x,\alpha) = \begin{cases} \{(0,0)\} & \text{if} \quad \alpha < 0, \\ \{0\} \times [0,1] & \text{if} \quad \alpha = 0, \\ \{(0,1)\} & \text{if} \quad \alpha > 0. \end{cases}$$

We consider the following two cases.

Case 1. There exists a sequence $(x_n, \alpha_n) \to (\bar{x}, f(\bar{x}))$ such that

$$r_n := d_{S_1 \cap S_2}(x_n, \alpha_n) > n\big(d_{S_1}(x_n, \alpha_n) + d_{S_2}(x_n, \alpha_n)\big), \ n = 1, 2, \ldots .$$

Thus,

$$d_{S_1}(x_n, \alpha_n) + d_{S_2}(x_n, \alpha_n) < \frac{r_n}{n} \leq \min_{X \times \mathbb{R}}\big(d_{S_1}(.) + d_{S_2}(.)\big) + \frac{r_n}{n}.$$

By the Ekeland variational principle, for every $n = 1, 2, \ldots$ there exists a point $(z_n, \beta_n) \in X \times \mathbb{R}$ such that $\|(z_n, \beta_n) - (x_n, \alpha_n)\| < r_n$ and (z_n, β_n) is a minimizer of the function

$$g_n(x,\alpha) := d_{S_1}(x,\alpha) + d_{S_2}(x,\alpha) + \frac{1}{n}\|(x,\alpha) - (z_n, \beta_n)\|.$$

Therefore,

$$(0,0) \in \partial^F g_n(z_n, \beta_n), \quad n = 1, 2, \ldots . \tag{4.1}$$

Observe that $(z_n, \beta_n) \notin S_1 \cap S_2$; indeed, otherwise we would have

$$\|(x_n, \alpha_n) - (z_n, \beta_n)\| \geq d_{S_1 \cap S_2}(x_n, \alpha_n) = r_n,$$

a contradiction. Since S_1 and S_2 are closed, there exists $\delta_n > 0$ such that either $((z_n, \beta_n) + \delta_n B_{X \times \mathbb{R}}) \cap S_1 = \emptyset$ or $((z_n, \beta_n) + \delta_n B_{X \times \mathbb{R}}) \cap S_2 = \emptyset$, and $\delta_n \to 0$ as $n \to \infty$.

Now, using the fuzzy sum rule in (4.1) (note that d_{S_1} and d_{S_2} are Lipschitz), there exist

$$\begin{gathered} (u_n, a_n) \in (z_n, \beta_n) + \delta_n B_{X \times \mathbb{R}}, \\ (v_n, b_n) \in (z_n, \beta_n) + \delta_n B_{X \times \mathbb{R}}, \\ (0, a^*_n) \in \partial^F d_{S_1}(u_n, a_n), \\ (\zeta_n, -b^*_n) \in \partial^F d_{S_2}(v_n, b_n) \end{gathered}$$

such that

$$(0,0) \in (0,a_n^*) + (\zeta_n, -b_n^*) + \frac{2}{n} B_{X^* \times \mathbb{R}}. \tag{4.2}$$

Thus, either $(u_n, a_n) \notin S_1$ or $(v_n, b_n) \notin S_2$. By Lemma 4.2, then either $|a_n^*| = 1$ (since $a_n^* \geq 0$, this means that $a_n^* = 1$) or $\|(\zeta_n, -b_n^*)\| = 1$. Combining this observation and (4.2), we derive that

$$\zeta_n \to 0 \quad \text{and} \quad b_n^* \to 1 \quad \text{as} \quad n \to \infty. \tag{4.3}$$

On the other hand, $(\zeta_n, -b_n^*) \in \partial^F d_{S_2}(v_n, b_n)$. Lemma 4.3 gives the existence of

$$\begin{gathered}(y_n, \lambda_n) \in S_2,\\ (\xi_n, -\mu_n) \in N^F(S_2, (y_n, \lambda_n))\end{gathered}$$

such that

$$\|(y_n, \lambda_n) - (v_n, b_n)\| < d_{S_2}(v_n, b_n) + \frac{1}{n}, \quad \|(\xi_n, -\mu_n) - (\zeta_n, -b_n^*)\| < \frac{2}{n}.$$

Hence, $(y_n, f(y_n)) \to (\bar{x}, f(\bar{x}))$, and by (4.3), $\mu_n \to 1$, $\xi_n \to 0$. When n is large, then $\mu_n > 0$. Since $(\xi_n, -\mu_n) \in N^F(\text{epi} f, (y_n, \lambda_n))$, using Lemma 2.2, we obtain

$$\xi_n/\mu_n \in \partial^F f(y_n).$$

From the above, we derive that, for any $\epsilon > 0$, $\eta > 0$, when n is large enough, we have

$$(y_n, f(y_n)) \in (\bar{x}, f(\bar{x})) + \eta B_{X \times \mathbb{R}} \quad \text{and} \quad \xi_n/\mu_n \in \partial^F f(y_n) \cap \epsilon B_{X^*}.$$

We obtain (A1).

Case 2. There are $a > 0$, $r > 0$ such that

$$d_{S_1 \cap S_2}(x, \alpha) \leq a\big(d_{S_1}(x, \alpha) + d_{S_2}(x, \alpha)\big) \quad \text{for all } (x, \alpha) \in (\bar{x}, f(\bar{x})) + r B_{X \times \mathbb{R}}.$$

Fix $\xi \in N^F(S, \bar{x})$. Clearly, $(\xi, 0) \in N^F(S_1 \cap S_2, (\bar{x}, f(\bar{x})))$. By Lemma 4.3,

$$(\xi, 0) \in b\partial^F d_{S_1 \cap S_2}(\bar{x}, f(\bar{x})) \quad \text{for some} \quad b > \|\xi\| + 1.$$

Therefore,

$$(\xi, 0) \in \kappa \partial^F \big(d_{S_1}(.) + d_{S_2}(.)\big)(\bar{x}, f(\bar{x})), \quad \text{where} \quad \kappa := ab. \tag{4.4}$$

Since f is lower semicontinuous, there exists $\eta \in (0, \frac{\epsilon}{2})$ such that

$$f(x) > f(\bar{x}) - \frac{\epsilon}{2} \quad \text{for all} \quad x \in \bar{x} + \eta B_X. \tag{4.5}$$

Now, apply the fuzzy sum rule to (4.4), to obtain the existence of

$$\begin{gathered}(u, \alpha),\ (v, \beta) \in (\bar{x}, f(\bar{x})) + \frac{\eta}{4} B_{X \times \mathbb{R}},\\ (0, \gamma) \in \partial^F d_{S_1}(u, \alpha),\\ (v^*, \lambda) \in \partial^F d_{S_2}(v, \beta)\end{gathered}$$

such that

$$(\xi,0)\in\kappa\big((0,\gamma)+(v^*,\lambda)\big)+\frac{\epsilon}{4}B_{X^*\times\mathbb{R}}.$$

Thus,

$$\xi\in\kappa v^*+\frac{\epsilon}{4}B_{X^*}. \tag{4.6}$$

Since $(v^*,\lambda)\in\partial^F d_{S_2}(v,\beta)$, by Lemma 4.3, there exist $(z,\nu)\in S_2:=\mathrm{epi}f$ and $(z^*,-\nu^*)\in N^F(S_2,(z,\nu))$ such that

$$\|(z,\nu)-(v,\beta)\|\le d_{S_2}(v,\beta)+\frac{\eta}{2} \tag{4.7}$$

and

$$\|(v^*,\lambda)-(z^*,-\nu^*)\|<\frac{\epsilon}{4\kappa}. \tag{4.8}$$

Combining (4.5) and (4.7), we obtain $(z,f(z))\in(\bar{x},f(\bar{x}))+\frac{\epsilon}{2}B_{X\times\mathbb{R}}$. From (4.6) and (4.8) we derive that $\|\kappa z^*-\xi\|<\frac{\epsilon}{2}$. Next, we apply Lemma 2.2 to $(z^*,-\nu^*)\in N^F(S_2,(z,\nu))$, to obtain the following:

• If $\nu^*\neq 0$, then $\zeta:=z^*/\nu^*\in\partial^F f(z)$ and $\|\lambda\zeta-\xi\|<\epsilon$. Here, $\lambda=\kappa\nu^*$. We thus obtain (A2).

• Else, there exist

$$(y,f(y))\in(z,f(z))+\frac{\epsilon}{2}B_{X\times\mathbb{R}}\subseteq(\bar{x},f(\bar{x}))+\epsilon B_{X\times\mathbb{R}},$$

$t>0$, and $\zeta:=y^*/t\in\partial^F f(y)$ such that $\|y^*-z^*\|<\frac{\epsilon}{2\kappa}$. Therefore, $\|\lambda\zeta-\xi\|<\epsilon$, where $\lambda=\kappa t$. We again obtain (A2). The proof is complete. □

THEOREM 4.4. *Let X be an Asplund space, and let $f:X\to\mathbb{R}\cup\{+\infty\}$ be a continuous function. Let $S:=\{x\in X:f(x)=0\}$, and let $\bar{x}\in S$. Then, either*

(B1) *for any $\epsilon,\eta>0$, there exists $(x,f(x))\in(\bar{x},f(\bar{x}))+\eta B_{X\times\mathbb{R}}$ such that $\big[\partial^F f(x)\cup\partial^F(-f)(x)\big]\cap\epsilon B_{X^*}\neq\emptyset$,*

or

(B2) *for any $\xi\in N^F(S,\bar{x})$, any $\epsilon>0$, there exist $(x,f(x))\in(\bar{x},f(\bar{x}))+\epsilon B_{X\times\mathbb{R}}$, $\zeta\in\big[\partial^F f(x)\cup\partial^F(-f)(x)\big]$, and $\lambda>0$ such that $\|\lambda\zeta-\xi\|<\epsilon$.*

Proof. Set

$$S=\big\{x\in X:f(x)=0\big\},\ S_1=\big\{x\in X:f(x)\le 0\big\},\ \text{ and }\ S_2=\big\{x\in X:f(x)\ge 0\big\}.$$

We first prove that

$$N^F(S,\bar{x})\subseteq N^F(S_1,\bar{x})\cup N^F(S_2,\bar{x}). \tag{4.9}$$

Let $x^*\in N^F(S,\bar{x})$. Assume to the contrary that $x^*\notin N^F(S_1,\bar{x})\cup N^F(S_2,\bar{x})$. It follows that there exist $\epsilon_0>0$, sequences $\{x_n\}_{n\in\mathbb{N}},\{y_n\}_{n\in\mathbb{N}}$ converging, respectively, to $\bar{x}$ such that $x_n\in S_1,y_n\in S_2$, and

$$\langle x^*,x_n-\bar{x}\rangle\ge\epsilon_0\|x_n-\bar{x}\|,$$

$$\langle x^*,y_n-\bar{x}\rangle\ge\epsilon_0\|y_n-\bar{x}\|.$$

Since f is continuous, there exists $z_n = \lambda_n x_n + (1-\lambda_n)y_n$ with $\lambda_n \in [0,1]$ such that $z_n \in S$. Obviously, $z_n \to \bar{x}$.

From the above inequalities, we obtain

$$\begin{aligned}\langle x^*, z_n - \bar{x}\rangle &= \lambda_n\langle x^*, x_n - \bar{x}\rangle + (1-\lambda_n)\langle x^*, y_n - \bar{x}\rangle \\ &\geq \epsilon_0(\lambda_n\|x_n - \bar{x}\| + (1-\lambda_n)\|y_n - \bar{x}\|) \\ &\geq \epsilon_0\|z_n - \bar{x}\|.\end{aligned}$$

This implies that $x^* \notin N^F(S,\bar{x})$, a contradiction. Then, Theorem 4.4 follows immediately from Theorem 4.1 and inclusion (4.9). □

Observe that we can use Theorems 4.1 and 4.4 and the method in [1] to establish a fuzzy multiplier rule for problem $(\mathfrak{P})$ in an Asplund space setting.

Acknowledgment. The first author would like to thank the LACO from the University of Limoges for hospitality and support.

REFERENCES

[1] J. M. Borwein, J. S. Treiman, and Q. S. Zhu, *Necessary conditions for constrained optimization problems with semicontinuous and continuous data,* Trans. Amer. Math. Soc., 350 (1998), pp. 2409–2429.
[2] F. H. Clarke, *Optimization and Nonsmooth Analysis,* Wiley and Sons, New York, 1983.
[3] I. Ekeland, *On the variational principle,* J. Math. Anal. Appl., 47 (1970), pp. 324–353.
[4] M. Fabian, *Subdifferentials, local ϵ-supports and Asplund spaces,* J. London Math. Soc., 34 (1986), pp. 569–576.
[5] M. Fabian, *Subdifferentiability and trustworthiness in the light of a new variational principle of Borwein and Preiss,* Acta. Univ. Carolin., 30 (1989), pp. 51–56.
[6] J. B. Hiriart-Urruty, *Refinements of necessary optimality conditions in nondifferentiable programming,* Appl. Math. Optim., 5 (1979), pp. 63–82.
[7] A. Ioffe, *Necessary conditions for nonsmooth optimization,* Math. Oper. Res., 9 (1984), pp. 59–189.
[8] A. Ioffe, *Proximal analysis and approximate subdifferentials,* J. London Math. Soc., 41 (1990), pp. 175–192.
[9] A. Jourani and M. Théra, *On limiting Fréchet ϵ-subdifferentials, generalized convexity, generalized monotonicity: Recent results,* in Proceedings of an Advanced Research Workshop, Marseille, Luminy, France, 1996, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998, pp. 185–198.
[10] A. Jourani and L. Thibault, *Metric inequality and subdifferential calculus in Banach spaces,* Set-Valued Anal., 3 (1995), pp. 87–100.
[11] A. Y. Kruger and B. S. Mordukhovich, *Extremal points and the Euler equation in nonsmooth optimization,* Dokl. Akad. Nauk BSSR, 24 (1980), pp. 684–687.
[12] A. Y. Kruger, *Properties of generalized subdifferentials,* Siberian Math. J., 26 (1985), pp. 822–832.
[13] B. S. Modukhovich, *Maximum principle in problems of time optimal control with nonsmooth constraints,* Appl. Math. Mech., 40 (1976), pp. 960–969.
[14] B. S. Modukhovich, *Metric approximations and necessary optimality conditions for general class of nonsmooth extremal problems,* Soviet Math. Dokl., 22 (1980), pp. 526–530.
[15] B. S. Mordukhovich, *Approximation Methods in Problems of Optimization and Control,* Nauka, Moscow, Russia, 1988.
[16] B. S. Modukhovich and Y. Shao, *Extremal characterization of Asplund spaces,* Proc. Amer. Math. Soc., 124 (1996), pp. 197–205.
[17] B. S. Modukhovich and Y. Shao, *Nonsmooth sequential analysis in Asplund spaces,* Trans. Amer. Math. Soc., 348 (1996), pp. 1235–1280.
[18] B. S. Mordukhovich, *The extremal principle and its applications to optimization and economics,* in Optimization and Related Topics, A. Rubinov and B. M. Glover, eds., Appl. Optim. 47, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2001, pp. 343–369.
[19] B. S. Mordukhovich and B. Wang, *Necessary suboptimality and optimality conditions via variational principle,* SIAM J. Control Optim., to appear.

[20] H. V. Ngai, D. T. Luc, and M. Théra, *On ϵ-convexity and ϵ-monotonicity,* in Calculus of Variations and Differential Equations, Proceedings of an Advanced Research Workshop, Haifa, Israel, 1998, Res. Notes Math. 1402, Chapman and Hall, London, UK, 1999, pp. 82–100.

[21] H. V. Ngai, D. T. Luc, and M. Théra, *Extensions of Fréchet ϵ-subdifferential calculus and applications,* J. Math. Anal. Appl., to appear.

[22] H. V. Ngai and M. Théra, *On metric inequality, subdifferential calculus and application,* Set-Valued Anal., 9 (2001), pp. 187–216.

[23] R. R. Phelps, *Convex Functions, Monotone Operator and Differentiability,* 2nd ed., Lecture Notes in Math. 1364, Spinger-Verlag, New York, 1993.

[24] R. T. Rockafellar, *Proximal subgradients, marginal values and augmented Lagrangians in nonconvex optimization,* Math. Oper. Res., 6 (1981), pp. 424–436.

[25] L. Thibault, *On subdifferentials of optimal value functions,* SIAM J. Control Optim., 29 (1991), pp. 1019–1036.

[26] J. S. Treiman, *The linear nonconvex generalized gradient and Lagrange multipliers,* SIAM J. Optim., 5 (1995), pp. 670–680.

[27] D. E. Ward and J. M. Borwein, *Nonsmooth calculus in finite dimensions,* SIAM J. Control Optim., 25 (1987), pp. 1312–1340.

SIAM J. OPTIM.
Vol. 12, No. 3, pp. 669–691

SOLVING SOME LARGE SCALE SEMIDEFINITE PROGRAMS VIA THE CONJUGATE RESIDUAL METHOD*

KIM-CHUAN TOH† AND MASAKAZU KOJIMA‡

Abstract. Most current implementations of interior-point methods for semidefinite programming use a direct method to solve the Schur complement equation (SCE) $M\Delta y = h$ in computing the search direction. When the number of constraints is large, the problem of having insufficient memory to store M can be avoided if an iterative method is used instead. Numerical experiments have shown that the conjugate residual (CR) method typically takes a huge number of steps to generate a high accuracy solution. On the other hand, it is difficult to incorporate traditional preconditioners into the SCE, except for block diagonal preconditioners. We decompose the SCE into a 2×2 block system by decomposing Δy (similarly for h) into two orthogonal components with one lying in a certain subspace that is determined from the structure of M. Numerical experiments on semidefinite programming problems arising from the Lovász θ-function of graphs and MAXCUT problems show that high accuracy solutions can be obtained with a moderate number of CR steps using the proposed equation.

Key words. large scale semidefinite programming, interior-point methods, inexact search directions, preconditioned conjugate residual method, deflated conjugate gradient method

AMS subject classification. 90C05

PII. S1052623400376378

1. Introduction.

Let $\mathcal{S}^n$ be the vector space of $n \times n$ real symmetric matrices endowed with the inner product $A \bullet B = \text{Trace}(AB)$. Given an integer n, we let $\bar{n} = n(n+1)/2$. Let **svec** be an isometry identifying $\mathcal{S}^n$ with $\mathbb{R}^{\bar{n}}$ such that $K \bullet L = \mathbf{svec}(K)^T\mathbf{svec}(L)$, and let **smat** be the inverse of **svec**. Given $k \times l$ matrices G, H, we define $G \circledast H : \mathcal{S}^l \longrightarrow \mathcal{S}^k$ by $G \circledast H(M) = (HMG^T + GMH^T)/2$, for $M \in \mathcal{S}^l$.

Consider the standard semidefinite program (SDP)

$$\begin{array}{lrcl} \min_X & C \bullet X & & \\ & A_k \bullet X & = & b_k, \quad k = 1, \ldots, m, \\ & X & \succeq & 0, \end{array} \tag{1}$$

where $b \in \mathbb{R}^m$, $A_k, C \in \mathcal{S}^n$, and $X \succeq 0$ means that X is positive semidefinite. The dual of (1) is

$$\begin{array}{ll} \max_{y,Z} & b^T y \\ & \sum_{k=1}^m y_k A_k + Z = C \\ & Z \succeq 0. \end{array}$$

For later discussion, let us introduce the linear map $\mathcal{A} : \mathcal{S}^n \longrightarrow \mathbb{R}^m$, defined by

$$\mathcal{A}(X) = [A_1 \bullet X \quad \cdots \quad A_m \bullet X]^T.$$

*Received by the editors August 4, 2000; accepted for publication (in revised form) May 31, 2001; published electronically January 9, 2002.

http://www.siam.org/journals/siopt/12-3/37637.html

†Department of Mathematics, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119620, Singapore (mattohkc@math.nus.edu.sg). This author's research was supported in part by National University of Singapore Academic Research grant RP972685.

‡Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, 2-12-1 Oh-Okayama, Meguro-ku, Tokyo 152, Japan (kojima@is.titech.ac.jp).

The adjoint of $\mathcal{A}$ with respect to the standard inner product in $\mathcal{S}^n$ and $\mathbb{R}^m$ is the linear map $\mathcal{A}^T : \mathbb{R}^m \longrightarrow \mathcal{S}^n$ defined by

$$\mathcal{A}^T y \;=\; \sum_{k=1}^{m} y_k A_k.$$

We consider primal-dual path-following methods [21, 23] for SDP using the Nesterov–Todd (NT) direction in which the general framework in each iteration is as follows. Given a current iterate (X, y, Z) and a centering parameter $\sigma \in [0, 1)$, where X, Z are symmetric positive definite, the methods find a search direction $(\Delta X, \Delta y, \Delta Z)$ so as to generate the next iterate by solving the following linear system of equations:

$$\mathcal{A}\Delta X = R_p \;:=\; b - \mathcal{A}X, \tag{2a}$$

$$\mathcal{A}^T\Delta y \;+\; \Delta Z = R_d \;:=\; C - Z - \mathcal{A}^T y, \tag{2b}$$

$$\mathcal{E}\Delta X \;+\; \mathcal{F}\Delta Z = R_c \;:=\; \sigma\mu I - \Sigma^2, \tag{2c}$$

where

$$\mu = X \bullet Z/n, \qquad \mathcal{E} = G^{-T} \circledast GZ, \qquad \mathcal{F} = G^{-T}X \circledast G,$$

and G is a matrix such that $\Sigma := GZG^T = G^{-T}XG^{-1}$ is a positive definite diagonal matrix. Note that $W := G^TG$ is the NT scaling matrix with $WZW = X$; see [21].

As in the case of linear programming, we can avoid solving the linear system of $n^2 + m$ equations (2a)–(2c) directly by first solving a Schur complement equation (SCE) involving only Δy and then computing $\Delta X, dZ$ in terms of Δy as follows:

$$M\Delta y = h \;:=\; R_p + \mathcal{A}\mathcal{E}^{-1}\mathcal{F}R_d - \mathcal{A}\mathcal{E}^{-1}R_c, \tag{3a}$$

$$\Delta Z = R_d - \mathcal{A}^T\Delta y, \tag{3b}$$

$$\Delta X = \mathcal{E}^{-1}R_c - \mathcal{E}^{-1}\mathcal{F}(R_d - \mathcal{A}^T\Delta y), \tag{3c}$$

where $M = \mathcal{A}\mathcal{E}^{-1}\mathcal{F}\mathcal{A}^T$ is the Schur complement matrix whose (i, j) element is given by

$$M_{ij} \;=\; A_i \bullet WA_jW, \quad i, j = 1, \ldots, m. \tag{3d}$$

Generally, (3a) is solved by a direct method through the following steps:

(i) Compute the $m \times m$ matrix M and store it in the computer memory;

(ii) Compute the Cholesky factorization of M, and obtain Δy by solving two triangular systems of linear equations.

The work required for step (i) is $2mn^3 + m^2n^2/2$ flops if the SDP data is dense. (A flop is one addition or one multiplication.) But when the data is sparse, substantial reduction in the computational cost of M is possible by exploiting the sparsity; see [7] for details. However, M is generally fully dense even when the SDP data is sparse. Thus when m is large, say more than a few thousands, it is impossible to store M in the memory of most current workstations. Furthermore, the $m^3/3$ flops required to compute the Cholesky factorization of M also become prohibitively expensive.

In this paper, we will mainly focus on SDPs where m is large but n is moderate (say, less than 1000). It is well known that one can overcome the memory problem

just mentioned by using an iterative method to solve (3a). The reason is that an iterative method only needs to access matrix-vector products of the form Mv, and these can be easily computed based on the operator description of M for any given vector v. As a result, M need not be formed explicitly, and hence the problem of having insufficient computer memory to store M does not arise in this case.

Of course, memory problems can also occur when n is large, since the primal variable X is typically dense even if the SDP data and the resulting dual variable Z are sparse. However, the root cause of this problem lies in the primal-dual framework used to solve the SDP and it cannot be easily overcome by simply using an iterative method to compute the search direction. For such a problem, it is more appropriate to use methods that avoid the need to form X explicitly. One such method is the dual scaling interior-point method proposed in [3], which is able to solve large SDPs (with $m, n \approx 2000$) arising from MAXCUT problems. Another method that avoids the explicit formation of X is the spectral bundle method proposed in [10]. (This is a first order method that is designed for SDPs in which Trace(X) is a constant. The authors succeeded in solving some large scale SDPs arising from MAXCUT problems and the Lovász θ-function on graphs, but only with low accuracies in the duality gaps.)

The idea of using an iterative method to solve (3a) in order to avoid excessive memory requirement is well known. In [16], a preconditioned conjugate gradient (CG) method was proposed to approximately solve the SCE in each iteration of a primal-dual interior-point method. Besides reporting some impressive computational results on SDPs arising from the Lovász θ-function on graphs and graph partitioning problems, that paper also gave a detailed discussion on how to incorporate inexact search directions into a primal-dual interior-point method. Building on the earlier work, a preconditioned conjugate residual (CR) method was proposed to solve the SCE in [17]. Recently, Choi and Ye [5] also reported computational results for large SDPs arising from MAXCUT problems (with n up to 14000). They used the dual scaling interior-point method described in [3], but solved the associated SCE in each iteration by a preconditioned CG method. Earlier works on using preconditioned CG methods to solve the SCE in interior-point methods for SDP include [14] and [24].

As far as we are aware, all the earlier works mentioned above used diagonal or block diagonal preconditioners. These preconditioners are ineffective when the Schur complement matrix becomes more and more ill-conditioned as the interior-point iterates approach an optimal solution. Although attempts were made in [14] to construct more effective preconditioners based on incomplete Cholesky factors, none really succeeded in overcoming the ill-conditioning problem of the SCE. As a result, in all these works, only low accuracies in the duality gaps could be obtained at reasonable costs.

In this paper, we propose a method to overcome the ill-conditioning problem of the Schur complement matrix as interior iterates approach an optimal solution. By analyzing the structure of the Schur complement matrix, we decompose the SCE into a 2×2 block system by decomposing Δy (similarly for h) into two orthogonal components with one lying in a certain subspace of $\mathbb{R}^m$ that is determined from the structure of M. We call the resulting 2×2 block system the projected SCE. Using the combination of applying the CR method to the SCE when interior-point iterates are not close to an optimal solution and switching to the projected SCE when they are, we are able to solve some large SDPs arising from the Lovász θ-function of graphs and MAXCUT problems to moderately high accuracies, but at reasonable costs.

Although we focus only on the NT direction in this paper, we should mention that the method presented here also works for the dual scaling direction considered in [3], where $\mathcal{E} = I \circledast I$, $\mathcal{F} = Z^{-1} \circledast Z^{-1}/\mu$. In fact, if the dual scaling method is used, our idea can in principle be used to solve sparse SDPs where m and n are both large. On the negative side, it seems that our proposed method cannot be adapted for the HRVW/KSH/M direction [11, 12, 15], for reasons that we will explain in section 5.

Now we introduce some notations. The MATLAB notation $[x\,;\,y]$ is used to denote the column vector formed by appending a column vector y to x. We let $\mathcal{I}$ be the set of indices of nonzero elements of the matrix $\sum_{k=1}^m |A_k|$ (where $|A_k|$ is the matrix whose (i,j) element is the magnitude of the corresponding element of A_k), and

$$\rho_s = (\text{number of nonzero elements of the matrix } \textstyle\sum_{k=1}^m |A_k|)/n^2,$$

$$\rho_t = (\text{total number of nonzero elements of } A_1, A_2, \ldots, A_m)/(mn^2).$$

We use $\|\cdot\|$ to denote the vector and matrix 2-norms, and $\|\cdot\|_F$ to denote the Frobenius norm.

An iterative method can solve (2a)–(2c) only approximately. In section 2, we discuss how to incorporate an approximately computed search direction (or inexact search direction) into a primal-dual interior-point method. In section 3, some basic results on the convergence of the CR method are given. The behavior of the CR method on SDPs arising from the Lovász θ-function of graphs is given in section 4. The derivation of the projected Schur complement equation and discussions on related issues are given in section 5. This is followed by numerical results showing the effectiveness of the projected Schur complement approach on two classes of SDPs, namely, those arising from the Lovász θ-function of graphs, and MAXCUT problems.

2. Inexact search directions. The use of iterative methods to solve the SCE (3a) requires less computer memory than does the use of a direct method. It also has the advantage that one can terminate the iterative solver whenever an approximate solution of (3a) is deemed sufficiently accurate. This can lead to a significant savings in the CPU time required in each interior-point iteration, especially during the initial phase where accurate computation of the search direction is not necessary. In [13], Kojima, Shida, and Shindoh proposed inexact search directions for which (2a) and (2b) are satisfied exactly but (2c) is relaxed. Given a fixed parameter $\kappa \in [0, 1)$, they said that $(\Delta X, \Delta y, \Delta Z)$ is an admissible direction if (2a) and (2b) hold, and

$$\|(\mathcal{E}\Delta X + \mathcal{F}\Delta Z) - R_c\|_F \;\le\; \kappa\, \|R_c\|_F\,. \tag{4}$$

Note that the Frobenius norm is used above out of convenience since it can usually be computed cheaply. However, the norms in (4) need not be computed exactly, and it is sufficient to have some reasonably accurate estimates. As a result, we should keep in mind that sometimes it is cheaper to use the 2-norm instead since it can be estimated via the Lanczos method [18, Chapter 4] by calculating dominant eigenvalues.

Under certain mild assumptions, it is shown in [13] that primal-dual interior-point methods using inexact admissible directions maintain the same polynomial complexity enjoyed by their counterparts that use exact search directions. In this paper, we follow the framework laid out in [13] for the computation of inexact directions. Suppose Δy satisfies (3a) only approximately, i.e.,

$$M\Delta y \;\approx\; h.$$

Let

$$r = h - M\Delta y.$$

Given such a Δy, we compute ΔZ and $\widehat{\Delta X}$ as in (3b) and (3c), respectively. Thus $(\widehat{\Delta X}, \Delta y, \Delta Z)$ satisfies (2b) and (2c) exactly, but not (2a), since

$$\mathcal{A}\widehat{\Delta X} \;=\; \mathcal{A}\mathcal{E}^{-1}R_c - \mathcal{A}\mathcal{E}^{-1}\mathcal{F}\left(R_d - \mathcal{A}^T\Delta y\right) \;=\; R_p - r.$$

However, (2a) can be made to hold exactly by replacing $\widehat{\Delta X}$ by the minimizer of the following linear least squares problem:

$$\min_{\Delta X} \left\|\Delta X - \widehat{\Delta X}\right\|_F$$
$$\text{such that} \quad \mathcal{A}\Delta X \;=\; R_p,$$

whose solution is given by

$$\Delta X \;=\; \widehat{\Delta X} \;+\; \mathcal{A}^T(\mathcal{A}\mathcal{A}^T)^{-1}\, r. \tag{5}$$

By using $(\Delta X, \Delta y, \Delta Z)$ as the search direction, (2a) and (2b) are satisfied exactly, but not (2c), where

$$\|(\mathcal{E}\Delta X + \mathcal{F}\Delta Z) - R_c\|_F = \left\|\mathcal{E}[\mathcal{A}^T(\mathcal{A}\mathcal{A}^T)^{-1}\, r]\right\|_F \;=:\; \|\mathcal{R}\|_F\,. \tag{6}$$

Thus $(\Delta X, \Delta y, \Delta Z)$ is an admissible direction as soon as $\|\mathcal{R}\|_F \le \kappa\,\|R_c\|_F$.

Note that in (6), the quantity $\|\mathcal{R}\|_F$ can be computed in $3n^3$ flops, since for a given $U \in \mathcal{S}^n$,

$$\mathcal{E}(U) \;=\; \left[(G^{-T}UG^{-1})\Sigma + \Sigma(G^{-T}UG^{-1})\right]/2$$

can be computed in $3n^3$ flops, to leading order.

Notice that in (5), one has to solve a system of linear equations in computing $\mathcal{A}^T(\mathcal{A}\mathcal{A}^T)^{-1}r$. Fortunately, for many large scale SDPs such as those arising from the Lovász θ-function of graphs and MAXCUT problems, the matrix $\mathcal{A}\mathcal{A}^T$ is diagonal or sparse, and hence $(\mathcal{A}\mathcal{A}^T)^{-1}r$ can be computed at reasonable cost. But for problems (such as those arising from control theory) where $\mathcal{A}\mathcal{A}^T$ is dense, the computation of $(\mathcal{A}\mathcal{A}^T)^{-1}r$ can be very expensive and should be done by using an iterative solver in order to avoid incurring a huge memory requirement in storing $\mathcal{A}\mathcal{A}^T$.

3. The conjugate residual method. The CR method [18] is a Krylov subspace method that is analogous to the CG method, except that the former minimizes the norm of the residual vector over the underlying Krylov subspace in each iteration while the latter orthogonalizes the residual vector against the underlying Krylov subspace. The advantage of using the CR method is that the residual norm decreases monotonically as the iteration progresses. In general, the CR and CG methods exhibit similar convergence behavior.

In our application of the CR method to the SCE (3a), the stopping criterion is based on the quantity $\mathcal{R}$ in (6) that is a function of the standard residual vector r. We feel that for our case it is even more desirable to have a method that exhibits a monotone convergence in the residual norm, and thus we choose the CR method over the more commonly used CG method as our iterative solver for solving the SCE.

Given an $N \times N$ symmetric linear system of equations $Bx = d$, the standard implementation of the CR method is as follows [18].

CR METHOD.
Compute $r_0 = d - Bx_0$, and Br_0. Set $p_0 = r_0$ and $Bp_0 = Br_0$.
For $j = 0, 1, 2, \ldots,$

$$\begin{aligned}
\alpha_j &= r_j^T Br_j / \|Bp_j\|^2 \\
x_{j+1} &= x_j + \alpha_j p_j \\
r_{j+1} &= r_j - \alpha_j Bp_j \\
&\text{Compute } Br_{j+1} \\
\beta_j &= r_{j+1}^T Br_{j+1} / r_j^T Br_j \\
p_{j+1} &= r_{j+1} + \beta_j p_j \\
&\text{Compute } Bp_{j+1} = Br_{j+1} + \beta_j Bp_j
\end{aligned}$$

end

Each iteration of the algorithm requires $12N$ flops, plus the cost of computing Br_{j+1}.

It is well known that the CR method will suffer from slow convergence when the coefficient matrix B has an unfavorable eigenvalue distribution. For readers who are not familiar with the convergence theory of the CR method, we will now briefly review how its convergence is related to the spectrum for the case where B is positive definite. Suppose the eigenvalues of B are distributed almost uniformly in an interval $[a, b]$ on the positive real line. Then, in exact arithmetic, the asymptotic convergence rate of the CR method is determined by the ratio $\tau := b/a$; specifically, the convergence rate is given by

$$\rho = \frac{\sqrt{\tau} - 1}{\sqrt{\tau} + 1}.$$

If, in addition, B has t isolated eigenvalues lying on the right of $[a, b]$, then these outliers may cause a stagnation (that is, very little reduction in the residual norm) of at most t steps during the initial phase of the CR method, but they do not affect the asymptotic convergence rate. Similarly, if B has isolated eigenvalues that lie on the left of $[a, b]$, then these outliers may cause a stagnation during the initial phase of the CR method, but the stagnation can last for many steps, depending on how close the outliers are to the origin. However, we should note that stagnation due to outliers will not occur if the initial residual vector r_0 is orthogonal to the eigenspace associated with these isolated eigenvalues. For a more detailed account of the convergence behavior of the CR method and other Krylov subspace iterative methods, we refer the reader to [6].

From the above discussion, we see that when the eigenvalues of B are distributed over a large interval, or when there are eigenvalues close to the origin, the convergence of the CR method can be exceedingly slow. In such a situation, it is necessary to apply a preconditioner to B to improve the convergence rate. That is, instead of applying the CR method to the original linear system of equations, one applies it to a transformed linear system, say $(L^{-1}BL^{-T})(L^T x) = L^{-1}d$, where L is a matrix chosen such that $Lz = f$ is easy to solve, and the spectrum of the preconditioned matrix $L^{-1}BL^{-T}$ is better behaved than that of the original matrix B in either having fewer outliers that are close to the origin or having eigenvalues that are distributed over a smaller interval.

So far, our discussion on the convergence behavior of the CR method is based on exact arithmetic. In the presence of rounding errors, there are further complications

in the behavior of the CR method. Roughly speaking, there are two main effects caused by rounding errors.

1. The residual norm will stop decreasing beyond a certain accuracy level (known as the final attainable level of accuracy) even if one continues the CR iteration. This happens because the residual vectors r_j generated by the CR method differ from the true residual vectors $d - Bx_j$ in the presence of rounding errors. In [9], it is shown that the CR method will stop making further progress when j is such that

$$\|d - Bx_j - r_j\| \;\approx\; u\,\|B\|\,\|x\|\,\mathcal{O}(j)\,(1 + \max\{\|x_i\| / \|x\| : i \le j\})\,.$$

   Here u denotes the machine precision.
2. Rounding errors deteriorate the convergence rate by causing a loss of orthogonality among the computed vectors (in exact arithmetic, they are orthogonal). For example, in exact arithmetic, the CR method will converge to the exact solution in no more than N steps. But due to rounding errors, it may take more than N steps to reduce the residual norm to the final attainable level of accuracy, and such a delay can apparently be arbitrarily long; see [20].

4. Behavior of the CR method on the Schur complement equation. In this section, we will present some numerical experiments showing the convergence behavior of the CR method in solving the following preconditioned version of (3a):

$$\underbrace{L^{-1}ML^{-T}}_{\widehat{M}}\quad(L^T\Delta y) \quad=\quad \underbrace{L^{-1}h}_{\hat{h}}, \tag{7}$$

where L is a nonsingular lower triangular matrix chosen to precondition M. Our immediate task is to construct a suitable preconditioner for M. In the current literature, most preconditioning techniques are proposed for a sparse linear system of equations where the sparse coefficient matrix is known explicitly, and preconditioners such as incomplete Cholesky factors are generally quite effective [18]. However, as the reader may recall, our matrix M is dense and is not formed explicitly. Thus, a lot of the current preconditioning techniques [18, Chapter 10] are not applicable to our linear system. The only obvious and easily implementable choices for our system are block diagonal preconditioners. That is, L is chosen to be the Cholesky factor of a matrix of the form $\mathrm{diag}(M_1, M_2, \ldots, M_k)$, where the M_i's are diagonal blocks of M. Note that if each block is just a scalar, then the diagonal elements of L are simply the square roots of the diagonal elements of M.

In [14], Lin and Saigal used incomplete Cholesky factors as preconditioners for the SCE arising from SDP relaxation of quadratic assignment problems. But their numerical results showed that these sophisticated preconditioners are not clearly better than the diagonal preconditioner.

We will first analyze the behavior of the CR method on the preconditioned SCE (7) associated with the SDP problem `theta2` taken from the SDPLIB collection [2]. For this problem, $n = 50$, $m = 498$, and we take LL^T to be the diagonal preconditioner of M. In each CR iteration, we compute $\widehat{M}v$ for a given $v \in \mathbb{R}^m$ via the procedure described in Table 1, where the cost is also estimated.

All the numerical results presented in this paper are computed using MATLAB 5.3 on a 400MHz Pentium II PC with 256M of memory. The parameter κ in (4) is set to $\kappa = 0.01$. The interior-point method we used is the primal-dual path-following

TABLE 1
Computational cost of a matrix-vector product for (7).

Computing	Number of flops required
$w := L^{-T}v$	m
$U := \mathcal{A}^T w$	$\rho_t\, mn^2$
$V := U\,W$	$2\rho_s\, n^3$
$\{T_{ij} \mid (i,j) \in \mathcal{I}\}$, where $T := W\,V$	$\rho_s\, n^3$
$u := \mathcal{A}(T)$	$\rho_t\, mn^2$
$\widehat{M}v = L^{-1}u$	$3\rho_s\, n^3 + 2\rho_t\, mn^2$

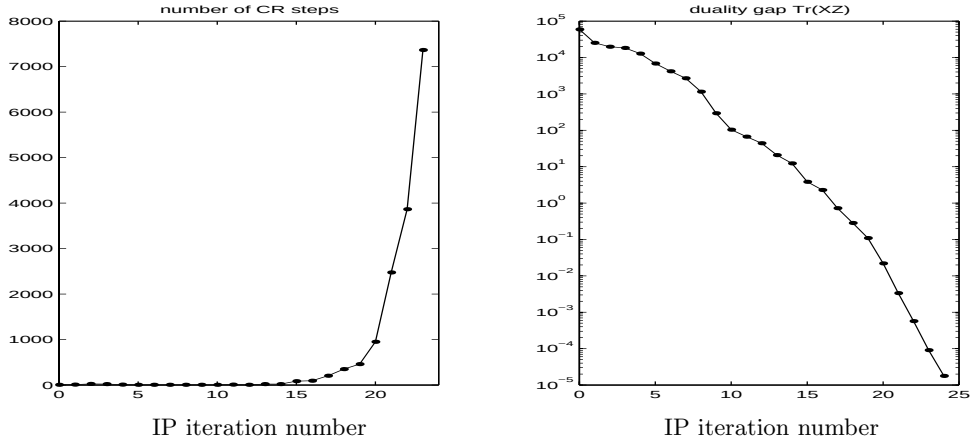


FIG. 1. *Behavior of the CR method on the SDP problem* `theta2`. *The left-hand plot is the number of CR steps taken in each interior-point iteration to solve the preconditioned Schur complement equation* (7) *approximately to satisfy the admissible condition* (4). *The right-hand plot is the corresponding duality gap* $X \bullet Z$. *The residual in the infeasibilities,* $\max(\|r_p\|, \|R_d\|_F)$, *is less than* 10^{-12} *after the* 9*th interior-point iteration.*

method (without corrector) described in [22]. For easy reference, we will refer to the interior-point method in [22] as Algorithm PFchol, and the corresponding method that uses the CR method to solve preconditioned SCE (7) as Algorithm PFCR. The default starting iterates described in [22] are used throughout.

Let

$$NCR(k) \;=\; \left\{ \begin{array}{l} \text{the number of CR steps required at the } k\text{th interior-point iteration} \\ \text{to solve (7) so that the admissible condition (4) is satisfied.} \end{array} \right.$$

In Figure 1, we plot $NCR(k)$ against k. The corresponding duality gap $X^k \bullet Z^k$ is shown in the same figure. We use the superscript "k" to denote dependence on the kth interior-point iterate (X^k, y^k, Z^k). Notice that the duality gap decreases at almost a linear rate. This implies that the norm $\|R_c^k\|_F$ also decreases at almost a linear rate as the interior-point iteration progresses. As a result, the admissible condition (4) becomes more and more stringent as k increases, and more CR steps are required for this condition to be fulfilled. Ideally, we would hope that $NCR(k)$ increases at most linearly with k. However, it is evident from Figure 1 that $NCR(k)$ increases far more rapidly than the reduction in $X^k \bullet Z^k$. As the latter becomes smaller, the convergence rate of the CR method deteriorates rapidly, resulting in a huge increase in $NCR(k)$. Furthermore, the effect of rounding errors also becomes more prominent, as can be

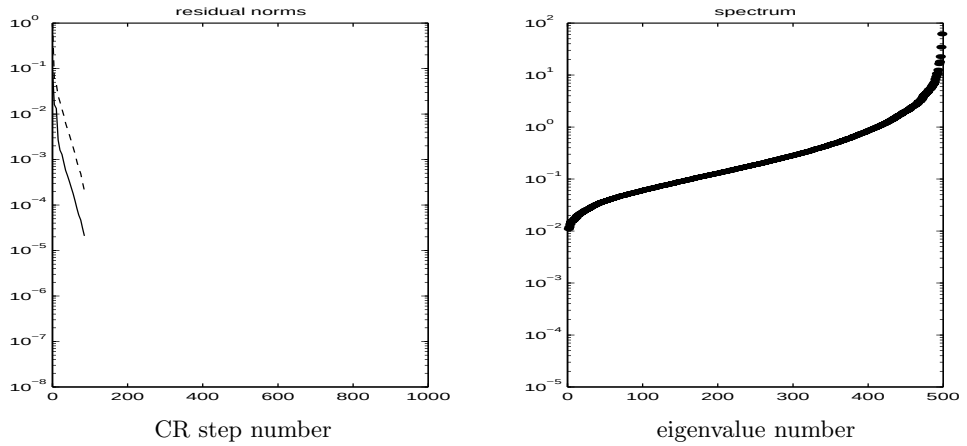


FIG. 2. *The dashed line in the left-hand plot is the residual norm* $\|h - Mv_j\| / \|h\|$ *generated by the CR method, where* M *and* h *correspond to the* 15*th interior-point iterate generated from the run in Figure* 1. *The solid line corresponds to the quantity* $\|\mathcal{R}\|_F$ *defined in* (6). *The right-hand plot is the spectrum of the preconditioned matrix* $\widehat{M}$.

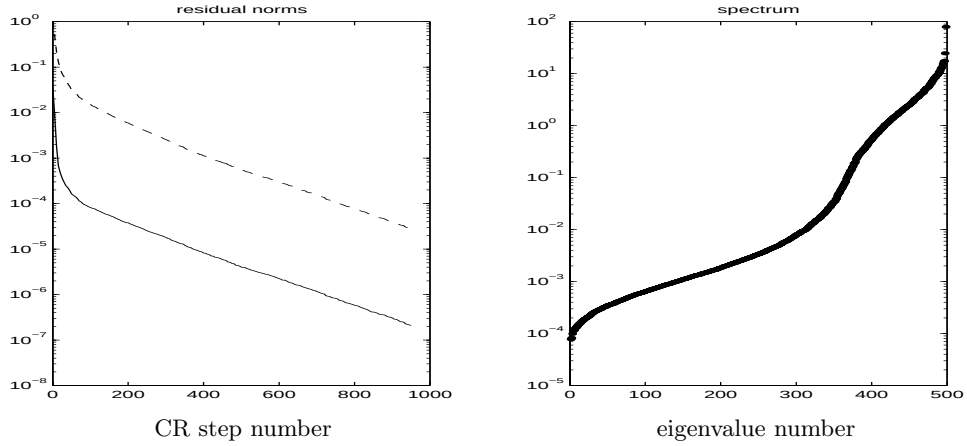


FIG. 3. *Same as Figure* 2, *but for the* 20*th interior-point iterate.*

seen from the fact that $NCR(22) > 5m$, when in fact this number should be at most m in exact arithmetic.

The deterioration in the convergence rate occurs because the spectrum of $\widehat{M}_k$ becomes "worse" as k increases. In Figures 2 and 3, the spectra of $\widehat{M}_k$ are plotted for $k = 15$ and $k = 20$, respectively. Based on these spectra, the convergence rates are at best equal to 0.94 and 0.993, respectively. This phenomenon of worsening convergence rate as the duality gap decreases is typical for the SCE. It is to be expected since the eigenvalues of $\widehat{M}_k$ are spread uniformly in an interval of the form $[O(\mu^k), O(1)]$.

In Table 2, we show the cumulative CPU time (in the format hours: minutes: seconds) taken by Algorithms PFchol and PFCR to solve a number of SDP problems arising from the Lovász θ-function of random graphs to an accuracy of 10^{-6} in the relative duality gaps. (The graphs were generated using a program provided by K. Nakata, whose help we gratefully acknowledge.) For the machine we used, the CPU

time required to compute the Cholesky factor of a given dense $m \times m$ positive definite matrix follows the empirical formula below:

$$\text{chol\_time}(m) \;\approx\; 4.446 \left(\frac{m}{100}\right)^3 \text{ secs.} \tag{8}$$

Those numbers which appear in typewriter style in Table 2 are the cumulative CPU times estimated from (8) to factorize M, and the memories needed to store M alone.

Observe that for the theta problems with large m, the CPU time taken by the CR method to solve these problems to an accuracy level of roughly 0.1 in the duality gaps can be one hundred times faster than that using the Cholesky factorization. This success can be attributed to the following facts when the duality gap is at a level above 0.1:

- the accuracy needed in computing an admissible direction is quite low;
- the matrix $\widehat{M}$ is well-conditioned, leading to a fast convergence of the CR method where it takes only a relatively small number of steps to obtain an admissible direction.

But once the interior-point iteration progresses to the stage where the duality gaps are smaller than 0.01, the CR method becomes highly inefficient, resulting in a huge increase in the CPU time needed to compute an admissible direction. For example, for the problem with $m = 23872$, the CR method spent only about 1 hour in the first 15 interior-point iterations, but almost 99 hours in the last 6 iterations. On the other hand, for the direct method using Cholesky factorization, the time taken is still the same as in all the previous iterations. The result is that, when the duality gaps are less than 0.01, an interior-point iteration using a direct method to solve (3a) is far more efficient than that using the CR method. But it is worthy to note from Table 2 that when m is very large ($m = 13390$ and $m = 23872$), the cumulative CPU time taken by PFCR to compute an approximate optimal solution with a relative duality gap of 10^{-6} or smaller is still less than that required by PFchol even if there is enough computer memory to store M.

We note that [16] also reported the same observations on the behavior of the CG method in solving a similar collection of SDPs.

Based on the above observations, one may want to combine the advantages of the CR and direct methods by using a hybrid method that employs the CR method to compute the search direction when the duality gap is above a certain level (say, 0.01) and then switch to the direct method when the duality gap falls below that level. But as our main purpose in this paper is to address the memory problem when m is large, we shall not distract ourselves with such a hybrid method.

Finally, observe that from Figures 2 and 3 the ratio at step j,

$$\frac{\|\mathcal{R}_j\|_F}{\|r_j\|} \;=\; \frac{\left\|\mathcal{E}\mathcal{A}^T(\mathcal{A}\mathcal{A}^T)^{-1}r_j\right\|_F}{\|r_j\|},$$

stays at an almost constant value for all j. Thus we can first estimate this ratio, say c, at the beginning of the CR method, then stop the CR method when we have

$$\|\mathcal{R}_j\|_F \;\approx\; c\,\|r_j\| \;\le\; \kappa\,\|R_c\|_F\,.$$

In this way we avoid the need to compute the quantity $\|\mathcal{R}_j\|$ for each j. This can sometimes lead to a substantial savings in the computation time in each CR step, especially when computing $(\mathcal{A}\mathcal{A}^T)^{-1}r_j$ or $\mathcal{E}(U)$ for a given U is expensive. The empirical fact just mentioned can be very useful in cutting down computation time in practice.

TABLE 2

Comparison of Algorithms PFchol and PFCR on a number of SDP problems arising from Lovász θ-function on graphs. CPU times are cumulative. The numbers in typewriter style are estimated figures needed to factorize and store M alone.

		Solving (3a) via Cholesky factorization of M				Solving (3a) via the CR method on (7) with diagonal preconditioner				
n	m	Iter. no.	CPU time	$X \bullet Z$	Mem. MB	Iter. no.	CPU time	$X \bullet Z$	Mem. MB	$NCR(k)$
200	1949	18	12:47	$3.6e-1$	80	18	2:57	$3.6e-1$	18	335
		21	14:55	$3.9e-3$	80	21	11:11	$3.9e-3$	18	2500
		24	17:02	$3.6e-5$	80	24	57:48	$3.6e-5$	18	9745
200	5986	15	3:58:30		287	15	5:10	$1.7e-1$	20	245
		18	4:46:12		287	18	33:41	$1.3e-3$	20	3015
		21	5:33:54		287	21	6:03:50	$2.2e-5$	20	29930
300	4375	20	2:35:05	$2.1e-1$	219	20	14:14	$2.4e-1$	27	445
		23	2:57:18	$1.2e-3$	219	23	1:08:16	$1.3e-3$	27	6120
		26	3:19:23	$1.2e-5$	219	26	8:36:40	$1.3e-5$	27	31315
300	13390	15	44:28:15		1434	15	22:18	$2.3e-1$	44	290
		18	53:21:54		1434	18	2:33:16	$2.1e-3$	44	3450
		21	62:15:33		1434	21	35:02:47	$2.5e-5$	44	52320
400	7905	20	12:12:00		500	20	34:06	$2.9e-1$	39	475
		23	14:01:48		500	23	2:59:19	$1.9e-3$	39	5435
		25	15:15:00		500	25	15:19:15	$6.6e-5$	39	27420
400	23872	15	251:56:15		4560	15	1:02:39	$2.8e-1$	89	290
		18	302:19:30		4560	18	6:44:35	$3.2e-3$	89	2860
		21	352:42:45		4560	21	99:57:54	$1.2e-5$	89	57200

5. The projected SCE. The numerical results in the last section show that the CR method is very efficient in computing an admissible search direction when $X \bullet Z$ is greater than 0.1, but becomes exceedingly slow when $X \bullet Z$ is smaller than 0.01 for the theta problems. In this section, we propose a method to overcome this difficulty based on the structure of the Schur complement matrix when $X \bullet Z$ is small.

Given an interior-point iterate (X, y, Z), let $\mu := X \bullet Z/n$, and let W be the associated NT scaling matrix. Suppose that (X, y, Z) is close to some optimal solution (X^*, y^*, Z^*) of the primal and dual SDP. If (X^*, Z^*) satisfies the strict complementarity, as well as the primal and dual nondegeneracy conditions defined in [1], then as (X, Z) approaches this optimal solution (i.e., when μ is sufficiently small), the eigenvalues of W will separate into two groups, one with large magnitudes of the order $\mathcal{O}(1/\sqrt{\mu})$ and the other with small magnitudes of the order $\mathcal{O}(\sqrt{\mu})$.

Now suppose that W has a group of p large eigenvalues and a group of $q := n - p$ small eigenvalues. Let $W = QDQ^T$ be the eigenvalue decomposition of W. We can rewrite W as

$$W \;=\; W_1 + W_2, \tag{9}$$

where $W_1 = Q_1 D_1 Q_1^T$ and $W_2 = Q_2 D_2 Q_2^T$, according to the partition $D = [D_1\ 0\,;\ 0\ D_2]$ and $Q = [Q_1\ \ Q_2]$, with $D_1 \in \mathbb{R}^{p\times p}$, $Q_1 \in \mathbb{R}^{n\times p}$ corresponding to the large eigenvalues, and $D_2 \in \mathbb{R}^{q\times q}$, $Q_2 \in \mathbb{R}^{n\times q}$ corresponding to the small eigenvalues.

Note that in practice it is only necessary to compute a partial eigenvalue decomposition of W. All we need is D_1 and Q_1, and then W_2 will be completely determined as $W - W_1$. If p is an integer that is much smaller than n, then D_1 and Q_1 can be

computed cheaply by using variants of the Lanczos method for computing dominant eigenvalues. For the dual scaling direction used in [3], instead of W, it is the large eigenvalues of $Z^{-1}/\sqrt{\mu}$ that should be computed. When Z is sparse and a sparse Cholesky factorization of it is readily available, computing the partial eigenvalue decomposition D_1 and Q_1 via Lanczos methods (applied to Z^{-1}) can sometimes be done efficiently even if n is large. Note that each step of a Lanczos method applied to Z^{-1} requires the solution of a system of linear equations with Z as the coefficient matrix, and Z^{-1} is not required explicitly.

When μ is sufficiently small, the number of eigenvalues of W with magnitudes $\mathcal{O}(1/\sqrt{\mu})$ is equal to the rank of X^*. Thus p is usually equal to the rank of X^*. In actual computation, however, we can set p to be any integer such that $\bar{p} \le m$, and it is not necessary to know the rank of X^*.

With the partition in (9), the Schur complement matrix M can be rewritten as

$$\begin{aligned} M &= \mathcal{A}\,(Q_1 \circledast Q_1)\,(D_1 \circledast D_1)\,(Q_1^T \circledast Q_1^T)\,\mathcal{A}^T + \mathcal{A}\,[(2W_1 + W_2) \circledast W_2]\,\mathcal{A}^T \\ &= \mathcal{A}_1 \mathcal{D}_1^2 \mathcal{A}_1^T + \mathcal{B}, \end{aligned} \tag{10}$$

where

$$\mathcal{A}_1 = \mathcal{A}(Q_1 \circledast Q_1), \qquad \mathcal{D}_1 = D_1^{1/2} \circledast D_1^{1/2}, \qquad \mathcal{B} = \mathcal{A}\,[(2W_1 + W_2) \circledast W_2]\,\mathcal{A}^T.$$

Note that $\mathcal{A}_1$ is a linear map from $\mathcal{S}^p$ into $\mathbb{R}^m$. But we will sometimes view $\mathcal{A}_1$ also as the matrix representation of the linear map with respect to the standard bases in $\mathcal{S}^p$ and $\mathbb{R}^m$. Under the assumption that (X^*, Z^*) satisfies the strict complementarity as well as the primal and dual nondegeneracy conditions defined in [1], the matrix $\mathcal{A}_1$ will have full column rank [1, Theorem 3] when μ is small and p is the rank of X^*.

Notice that the decomposition (10) depends on our ability to find the eigenvalue decomposition of $W \circledast W$. For the HRVW/KSH/M direction direction described in [11, 12, 15], $W \circledast W$ is replaced by $X \circledast Z^{-1}$. Unlike that of the former, the eigenvalue decomposition of the latter is not readily available even if those of X and Z are known. For this reason, the Schur complement matrix cannot be easily decomposed into the form in (10) for the HRVW/KSH/M direction.

We are now ready to describe our method to alleviate the problem of slow convergence when the CR method is applied to SCE (3a).

THEOREM 5.1. *Suppose $\mathcal{A}_1 \in \mathbb{R}^{m \times \bar{p}}$ has full column rank. Let $H = \mathcal{A}_1^T \mathcal{A}_1$. Then $H \in \mathbb{R}^{\bar{p} \times \bar{p}}$ is nonsingular, and solving the SCE, $Mv = h$, for v is equivalent to solving for v_1 and v_2 from the following linear system of equations:*

$$\underbrace{\begin{bmatrix} I + \mathcal{D}_1^{-1} H^{-1} \mathcal{A}_1^T \mathcal{B} \mathcal{A}_1 H^{-1} \mathcal{D}_1^{-1} & \mathcal{D}_1^{-1} H^{-1} \mathcal{A}_1^T \mathcal{B} \mathcal{Q} \\ \mathcal{Q} \mathcal{B} \mathcal{A}_1 H^{-1} \mathcal{D}_1^{-1} & \mathcal{Q} \mathcal{B} \mathcal{Q} \end{bmatrix}}_{K} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \mathcal{D}_1^{-1} H^{-1} \mathcal{A}_1^T h \\ \mathcal{Q} h \end{bmatrix}}_{g}, \tag{11}$$

where $\mathcal{Q} = I - \mathcal{A}_1 H^{-1} \mathcal{A}_1^T$. The solution v of the SCE can be recovered from

$$v \;=\; \mathcal{A}_1 H^{-1} \mathcal{D}_1^{-1} v_1 + v_2. \tag{12}$$

In addition, if we let $[r_1\,;\, r_2] = g - K[v_1\,;\, v_2]$, then

$$h - Mv \;=\; \mathcal{A}_1 \mathcal{D}_1 r_1 + r_2.$$

Proof. In this proof, we will view $\mathcal{A}_1$ as a matrix in $\mathbb{R}^{m\times\bar{p}}$ instead of as a linear operator from $\mathcal{S}^p$ to $\mathbb{R}^m$. Consider the decomposition

$$\mathcal{A}_1 \;=\; PR, \tag{13}$$

where $P \in \mathbb{R}^{m\times\bar{p}}$ is a matrix whose columns form an orthonormal set, and $R \in \mathbb{R}^{\bar{p}\times\bar{p}}$ is an upper triangular matrix. Let $\mathcal{Q} = I - PP^T$ be the orthogonal projection of $\mathbb{R}^m$ onto the orthogonal complement of Range($\mathcal{A}_1$). Then $v \in \mathbb{R}^m$ can be decomposed as

$$v \;=\; P\tilde{v}_1 + v_2, \tag{14}$$

where $\tilde{v}_1 = P^T v$ and $v_2 = \mathcal{Q}v$. Substituting (13) and (14) into the SCE, $Mv = h$, and using (10), we have

$$PR\,\mathcal{D}_1^2\,R^T\tilde{v}_1 + \mathcal{B}P\,\tilde{v}_1 + \mathcal{B}\,v_2 \;=\; h,$$

which implies that

$$(R\,\mathcal{D}_1^2\,R^T + P^T\mathcal{B}P)\tilde{v}_1 + P^T\mathcal{B}\,v_2 = P^T h,$$

$$\mathcal{Q}\mathcal{B}P\,\tilde{v}_1 + \mathcal{Q}\mathcal{B}\,v_2 = \mathcal{Q}\,h.$$

Thus

$$\left.\begin{array}{rcl} \left(I + \mathcal{D}_1^{-1}R^{-1}P^T\mathcal{B}PR^{-T}\mathcal{D}_1^{-1}\right)v_1 + \mathcal{D}_1^{-1}R^{-1}P^T\mathcal{B}\,v_2 &=& \mathcal{D}_1^{-1}R^{-1}P^T h, \\ \mathcal{Q}\mathcal{B}PR^{-T}\mathcal{D}_1^{-1}v_1 + \mathcal{Q}\mathcal{B}\,v_2 &=& \mathcal{Q}\,h, \end{array}\right\} \tag{15}$$

where $v_1 = \mathcal{D}_1 R^T\tilde{v}_1$. We can avoid the explicit formation of the $m \times \bar{p}$ matrix P by observing that $P = \mathcal{A}_1 R^{-1}$, and R is the Cholesky factor of the matrix $H = \mathcal{A}_1^T\mathcal{A}_1$. With this observation, it is readily verified that (15) can be rewritten as (11), and $\mathcal{Q} = I - \mathcal{A}_1 H^{-1}\mathcal{A}_1^T$. Finally, from (14), it is readily shown that (12) holds. ∎

We will call the linear system of equations in (11) the *projected Schur complement equation* since its derivation is based on the orthogonal projection of $\mathbb{R}^m$ onto Range($\mathcal{A}_1$). From the above theorem, we see that instead of applying the CR method to compute Δy approximately from (3a), we can compute it from (11). We may view (11) as a preconditioned version of (3a), but not in the conventional sense since an explicit description of an approximate inverse of M is not available.

The matrix K in (11) is a singular positive semidefinite matrix, but system (11) is consistent. In fact, K has $\bar{p}$ zero eigenvalues. But these zero eigenvalues do not impede the convergence of the CR method since the right-hand side vector is orthogonal to the eigenspace associated with the zero eigenvalues. By a result in [4], the convergence rate of the CR method for (11) is determined solely by the positive eigenvalues of K when the initial guess of the CR method is chosen to be the zero vector. To be more quantitative, the next theorem states a result on the positive eigenvalues of K.

THEOREM 5.2. *Let Λ_+ be the set of positive eigenvalues of the matrix $\mathcal{Q}\mathcal{B}\mathcal{Q}$. Assume that a partition in* (10) *is chosen such that $\|D_1\| = \mathcal{O}(1/\sqrt{\mu})$, $\|D_2\| = \mathcal{O}(\sqrt{\mu})$, and* $\mathrm{dist}(0,\Lambda_+) \gg \mathcal{O}(\sqrt{\mu})$. *Suppose λ is a positive eigenvalue of the matrix K in* (11). *Then*

$$\mathrm{dist}(\lambda, \Lambda_+ \cup \{1\}) = \|\mathcal{Q}\mathcal{B}\mathcal{A}_1\| \;\left\|H^{-1}\right\|\; \mathcal{O}(\sqrt{\mu}).$$

Note that $\|\mathcal{B}\|$ can be bounded by a constant that is independent of μ.

Proof. Observe that K can be written as

$$K \;=\; \begin{bmatrix} I & \\ & \mathcal{Q}\mathcal{B}\mathcal{Q} \end{bmatrix} + \begin{bmatrix} \mathcal{D}_1^{-1}H^{-1}\mathcal{A}_1^T\mathcal{B}\mathcal{A}_1H^{-1}\mathcal{D}_1^{-1} & \mathcal{D}_1^{-1}H^{-1}\mathcal{A}_1^T\mathcal{B}\mathcal{Q} \\ \mathcal{Q}\mathcal{B}\mathcal{A}_1H^{-1}\mathcal{D}_1^{-1} & \end{bmatrix},$$

where the second matrix in the right-hand side is considered as a perturbation to the first matrix. By the Bauer–Fike theorem, we have

$$\operatorname{dist}(\lambda, \Lambda_+ \cup \{1\}) \;\le\; \|\mathcal{Q}\mathcal{B}\mathcal{A}_1\| \left\|H^{-1}\right\| \left\|D_1^{-1}\right\| + \left\|H^{-1}\mathcal{A}_1^T\mathcal{B}\mathcal{A}_1H^{-1}\right\| \left\|\mathcal{D}_1^{-1}\right\|^2.$$

Now, the required result follows readily from the above inequality.

To complete the proof, we need to show that $\|\mathcal{B}\|$ can be bounded by a constant that is independent of μ. Note that

$$\begin{aligned} \|\mathcal{B}\| &\le \|\mathcal{A}\|^2 \left(2\,\|W_1 \circledast W_1\| + \|W_2 \circledast W_2\|\right) \\ &= \|\mathcal{A}\|^2 \left(2\left\|\begin{bmatrix} D_1 & 0 \\ 0 & 0 \end{bmatrix} \circledast \begin{bmatrix} 0 & 0 \\ 0 & D_2 \end{bmatrix}\right\| + \left\|\begin{bmatrix} 0 & 0 \\ 0 & D_2 \end{bmatrix} \circledast \begin{bmatrix} 0 & 0 \\ 0 & D_2 \end{bmatrix}\right\|\right) \\ &= \|\mathcal{A}\|^2 \left(\max_{1\le i\le p}(D_1)_{ii} \max_{1\le j\le q}(D_2)_{jj} + \max_{1\le j\le q}(D_2)_{jj}^2\right) \\ &= \mathcal{O}(1)\,\|\mathcal{A}\|^2. \end{aligned}$$

The last equality above follows from the assumptions made on D_1 and D_2. ∎

The above theorem implies that the positive eigenvalues of K are roughly contained in the set $\Lambda_+ \cup \{1\}$ when μ is small. Therefore, the convergence rate of the CR method applied to (11) is governed by the positive eigenvalues of $\mathcal{Q}\mathcal{B}\mathcal{Q}$. Since $\mathcal{B}$ is better conditioned than $\widehat{M}$, the effective condition number (the ratio of the largest to the smallest positive eigenvalues) of $\mathcal{Q}\mathcal{B}\mathcal{Q}$ is usually smaller than the condition number of $\widehat{M}$. Thus, we would expect the CR method to converge faster in this case than when it is applied to (7).

5.1. Connection to the deflated CG method. The projection step used in the derivation of the projected SCE bears some resemblance to the deflated CG method described in [19]. The basic process involved in the deflated CG method for solving a linear system of equation $Bx = d$ is as follows. Suppose a matrix U of appropriate dimension is given and U^TBU is nonsingular. Given an initial guess x_0 such that $U^Tr_0 = 0$, where $r_0 = d - Bx_0$, the deflated CG constructs successively for each j an approximate solution x_{j+1} such that

$$r_{j+1} := d - Bx_{j+1} \;\perp\; \text{Range}(U) + \langle r_0, r_1, \ldots, r_j\rangle.$$

In doing so, the convergence rate of the deflated CG method is governed by the eigenvalues of the matrix $\tilde{B} := B - BU(U^TBU)^{-1}U^TB$ instead of by those of the original matrix B.

Now, adapting the deflated CG to the Schur complement matrix M and taking $U = \mathcal{A}_1H^{-1}D_1^{-1}$, we have

$$\tilde{M} \;:=\; M - MU(U^TMU)^{-1}U^TM \;=\; \mathcal{Q}\mathcal{B}\mathcal{Q} - (I-\mathcal{Q})\mathcal{B}(I-\mathcal{Q}) + E,$$

where $\|E\| = \mathcal{O}(\|D_1^{-1}\|^2)$, and $\mathcal{B}, \mathcal{Q}$ are the matrices which appeared in Theorem 5.1. Thus if we apply the deflated CG to the SCE, the convergence rate is basically

governed by the matrix $\mathcal{Q}\mathcal{B}\mathcal{Q}-(I-\mathcal{Q})\mathcal{B}(I-\mathcal{Q})$ when $\|D_1^{-1}\|$ is small. This convergence rate is very similar to that for the projected SCE where it is governed by the matrix $\mathcal{Q}\mathcal{B}\mathcal{Q}$.

Computationally, each step j of the deflated CG method requires the solution of the linear system $U^TBU\xi = U^TBr_j$, which for the SCE is given by

$$(I + D_1^{-1}H^{-1}\mathcal{A}_1^T\mathcal{B}\mathcal{A}_1H^{-1}D_1^{-1})\,\xi \;=\; (D_1\mathcal{A}_1^T + D_1^{-1}H^{-1}\mathcal{A}_1^T\mathcal{B})\,r_j.$$

Notice that the coefficient matrix is the $(1,1)$ block of the matrix K in (11). To apply the CR method to (11) this matrix need not be formed explicitly, but for the deflated CG, we need to compute it explicitly. However, computing this coefficient matrix is very expensive. Due to this serious disadvantage, we will not explore any further in this paper the use of the deflated CG method to solve the SCE.

5.2. Cost per CR step. To minimize the computation time of the CR method when applied to (11), we need to compute efficiently a matrix-vector product for (11). To begin with, we state in Table 3 the cost for computing some basic steps required in the matrix-vector product.

TABLE 3
Computational cost of some basic steps needed in a matrix-vector product for (11).

Computing		Number of flops required
$\mathcal{A}^Tv$,	given $v \in \mathbb{R}^m$	$\rho_t\, mn^2$
$\mathcal{A}(T)$,	given $T \in \mathcal{S}^n$	$\rho_t\, mn^2$
$\mathcal{A}_1^Tv$,	given $v \in \mathbb{R}^m$	$p^2n + 2\rho_s\, pn^2 + \rho_t\, mn^2$
$\mathcal{A}_1(T)$,	given $T \in \mathcal{S}^p$	$2p^2n + \rho_s\, pn^2 + \rho_t\, mn^2$

From (11), it is easy to confirm that if we let

$$z \;=\; \mathcal{B}\left(\mathcal{A}_1H^{-1}\mathcal{D}_1^{-1}v_1 + \mathcal{Q}v_2\right) \;=\; \mathcal{B}\left(\mathcal{A}_1H^{-1}\left[\mathcal{D}_1^{-1}v_1 - \mathcal{A}_1^Tv_2\right] + v_2\right),$$

then

$$K\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \;=\; \begin{bmatrix} v_1 + \mathcal{D}_1^{-1}(H^{-1}\mathcal{A}_1^Tz) \\ z - \mathcal{A}_1(H^{-1}\mathcal{A}_1^Tz) \end{bmatrix}.$$

Let

(16) $\quad \omega_H \;=$ the number of flops required to compute $H^{-1}f$ for a given f.

If H is precomputed with Cholesky factorization $H = R^TR$, then $\omega_H = 2\bar{p}^2$. For the numerical experiments in this paper, we compute H explicitly where the cost is estimated to be at most $p(4\rho_t mn^2 + 2p^2n + (1+\rho_s)pn^2)$ flops. For easy reference, we summarize in Table 4 the cost for computing quantities such as W_1 and H that are needed in the projected SCE.

In our implementation of the CR method for (11), we compute $K[v_1\,;\,v_2]$ via the procedure described in Table 5, where the cost is also estimated. Compared to the cost of computing $\widehat{M}v$ in section 4, we see that computing $K[v_1\,;\,v_2]$ can be much more expensive. Exactly how much more expensive is difficult to estimate since it depends on the sparsity factors ρ_s, ρ_t, the cost ω_H in computing $H^{-1}f$, and the integers m, n, p. But hopefully, the reduction in the number of CR steps needed for computing an admissible direction will outweigh the higher cost required.

TABLE 4
Computational cost of quantities that are needed in the projected SCE (11).

Computing	Number of flops required
Q_1, D_1 via QR algorithm	$\mathcal{O}(n^3)$
W_1	$2pn^2$
$W_2 = W - W_1$	n^2
$H = \mathcal{A}_1^T \mathcal{A}_1$	at most $p(4\rho_t mn^2 + 2p^2 n + (1+\rho_s)pn^2)$
Cholesky factor R of H	$\frac{1}{3}\bar{p}^3$

TABLE 5
Computational cost required in the matrix-vector product for (11).

Computing	Number of flops required
$w := \mathcal{A}_1 H^{-1}(\mathcal{D}_1^{-1} v_1 - \mathcal{A}_1^T v_2) + v_2$	$3p^2 n + 3\rho_s\, pn^2 + 2\rho_t\, mn^2 + \omega_H$
$z := \mathcal{B} w$	$4\rho_s\, n^3 + 2\rho_t\, mn^2$
$H^{-1}\mathcal{A}_1^T z$	$p^2 n + 2\rho_s\, pn^2 + \rho_t\, mn^2 + \omega_H$
$\mathcal{A}_1(H^{-1}\mathcal{A}_1^T z)$	$2p^2 n + \rho_s\, pn^2 + \rho_t\, mn^2$
$K[v_1\,;\, v_2]$	$6p^2 n + 6\rho_s\, pn^2 + 4\rho_s\, n^3 + 6\rho_t\, mn^2 + 2\omega_H$

5.3. A preconditioner for K. The coefficient matrix K has an eigenvalue distribution that is much more favorable than that of the preconditioned Schur complement matrix $\widehat{M}$. But we can further improve the eigenvalue distribution by applying a preconditioner to K.

The most obvious preconditioner for K is the diagonal preconditioner. However, it turns out that computing the diagonal elements of K is very expensive; the cost of computing all the diagonal elements is at least $m(2p^2 n + pn^2 + 3\rho_s\, n^3 + 2\rho_t\, mn^2 + \omega_H)$. Thus we are forced to look for a cheaper alternative. We find that the diagonal preconditioner of $\mathcal{B}$ is such an alternative where the cost of computing all the diagonal elements is only about $3\rho_t mn^3$. We do not claim that the diagonal preconditioner of $\mathcal{B}$ is more effective than the diagonal preconditioner of K itself. But it is a much cheaper alternative that in practice does improve the convergence rate associated with K.

Suppose that Φ is the diagonal preconditioner of $\mathcal{B}$, and $\Phi = LL^T$ is its Cholesky factorization. We first rewrite (3a) using (10) as follows:

$$\mathcal{A}_1 \mathcal{D}_1^2 \mathcal{A}_1^T\, \Delta y + \mathcal{B}\, \Delta y \;=\; h. \tag{17}$$

Then we precondition the above system by L to get

$$\widehat{\mathcal{A}}_1 \mathcal{D}_1^2 \widehat{\mathcal{A}}_1^T\, \Delta \hat{y} + \widehat{\mathcal{B}}\, \Delta \hat{y} \;=\; \hat{h}, \tag{18}$$

where

$$\widehat{\mathcal{A}}_1 = L^{-1}\mathcal{A}_1, \quad \widehat{\mathcal{B}} = L^{-1}\mathcal{B}L^{-T}, \quad \Delta\hat{y} = L^T \Delta y, \quad \hat{h} = L^{-1}h. \tag{19}$$

Thus instead of using Theorem 5.1 to solve (17), we apply it to (18), and the associated equation has the form:

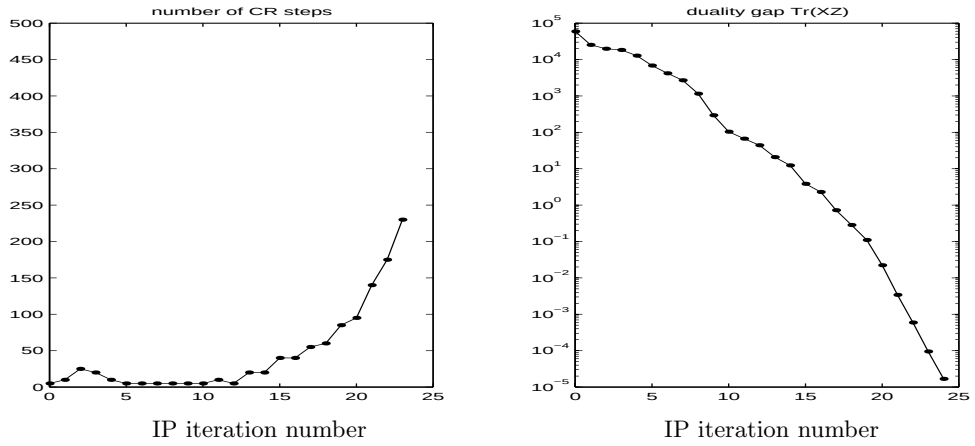


FIG. 4. *Behavior of the CR method on the SDP problem* `theta2`. *The left-hand plot is the number of CR steps taken in each iteration of Algorithm PFCR1 to solve either* (7) *or* (20) *in order to approximately satisfy the admissible condition* (4). *Note that from the* 15*th iteration onwards,* (20) *is used. The right-hand plot is the corresponding duality gap* $X \bullet Z$. *Note the difference in scale between this figure and Figure* 1.

(20)
$$\underbrace{\begin{bmatrix} I + \mathcal{D}_1^{-1}\widehat{H}^{-1}\widehat{\mathcal{A}}_1^T\widehat{\mathcal{B}}\widehat{\mathcal{A}}_1\widehat{H}^{-1}\mathcal{D}_1^{-1} & \mathcal{D}_1^{-1}\widehat{H}^{-1}\widehat{\mathcal{A}}_1^T\widehat{\mathcal{B}}\widehat{\mathcal{Q}} \\ \widehat{\mathcal{Q}}\widehat{\mathcal{B}}\widehat{\mathcal{A}}_1\widehat{H}^{-1}\mathcal{D}_1^{-1} & \widehat{\mathcal{Q}}\widehat{\mathcal{B}}\widehat{\mathcal{Q}} \end{bmatrix}}_{\widehat{K}} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \mathcal{D}_1^{-1}\widehat{H}^{-1}\widehat{\mathcal{A}}_1^T\hat{h} \\ \widehat{\mathcal{Q}}\hat{h} \end{bmatrix}}_{\hat{g}},$$

where $\widehat{H} = \widehat{\mathcal{A}}_1^T\widehat{\mathcal{A}}_1$ and $\widehat{\mathcal{Q}} = I - \widehat{\mathcal{A}}_1\widehat{H}^{-1}\widehat{\mathcal{A}}_1^T$.

Let $\widehat{K}$ be the coefficient matrix in (20). We observe that the eigenvalue distribution of $\widehat{K}$ is better than that of K for the SDPs considered in this paper.

6. Behavior of the CR method on the projected Schur complement equation. In view of the efficiency of the CR method in computing an admissible search direction via (7) when the duality gap $X \bullet Z$ is not too small, we propose in this section a hybrid method that combines the advantages of applying the CR method to (7) and (20) for computing the search direction in each interior-point iteration. The details of the hybrid method are given in Algorithm PFCR1. For the numerical experiments in this section, when step 3(b) in Algorithm PFCR1 is invoked, the $\bar{p} \times \bar{p}$ matrix $\widehat{H}$ is formed explicitly and its Cholesky factorization $\widehat{H} = R^T R$ is computed before the CR method is applied to (20).

Figures 4, 5, and 6 are the analogues of Figures 1, 2, and 3, respectively, for the SDP problem `theta2`, but using Algorithm PFCR1 instead of Algorithm PFCR. Comparing Figures 1 and 4, we see that the CR method method takes far fewer steps to solve (20) than are needed to solve (7) when $X \bullet Z$ is small. This indicates that the matrix $\widehat{K}$ has a much more favorable eigenvalue distribution than does to $\widehat{M}$. This is indeed confirmed in Figures 5 and 6, where the eigenvalues of $\widehat{K}$ are plotted. From the spectra shown in these figures, the convergence rates of the CR method applied to (20) are roughly equal to 0.87 and 0.93 at the 15th and 20th interior-point iteration, respectively.

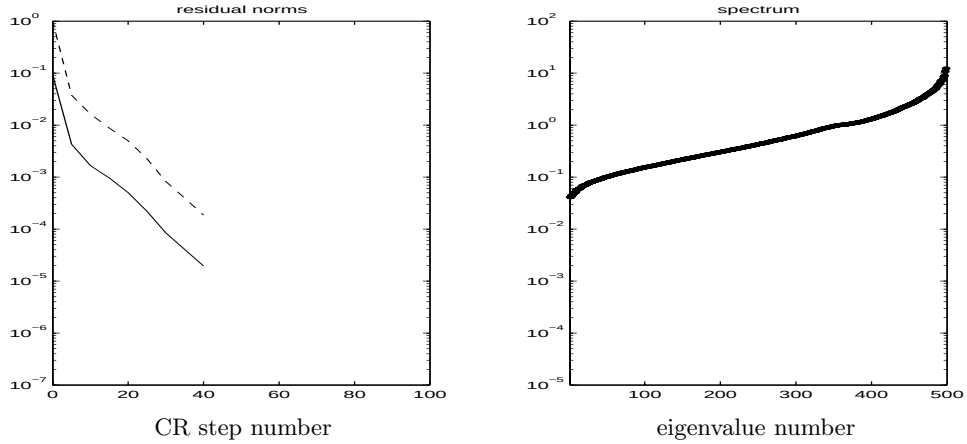


FIG. 5. *The dashed line in the left-hand plot is the residual norm* $\|h - Mv_j\| / \|h\|$ *generated by the CR method, where* M *and* h *correspond to the* 15*th interior-point iterate generated from the run in Figure* 4. *The solid line corresponds to the quantity* $\|\mathcal{R}\|_F$ *defined in* (6). *The right-hand plot is the spectrum of* $\widehat{K}$. *Note that there are* 45 *zero eigenvalues not shown in the plot. Also note the difference in scale between this figure and Figure* 2.

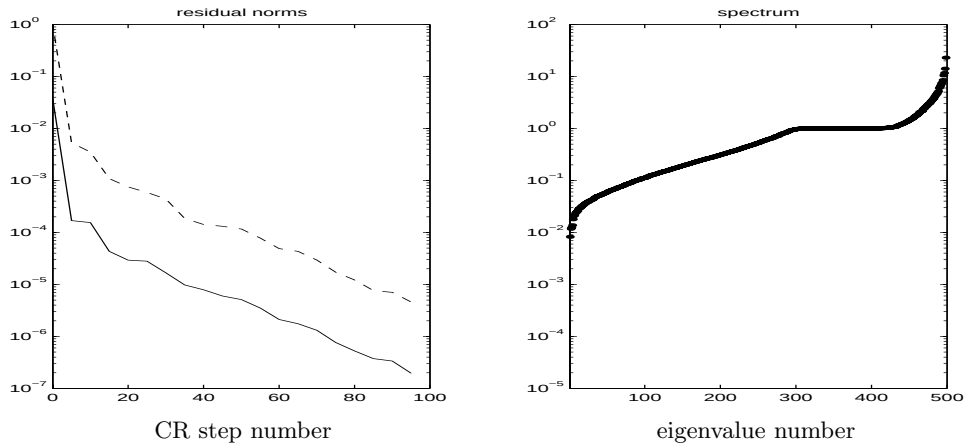


FIG. 6. *Same as Figure* 5*, but for the* 20*th interior-point iterate.*

Table 6 compares the cumulative CPU time taken by Algorithms PFchol and PFCR1 at various interior-point iterations in order to achieve an accuracy of 10^{-6} in the relative duality gaps. It shows that Algorithm PFCR1 performs very much better than Algorithm PFchol, especially for problems with large m. For example, consider the problem with $m = 13390$; Algorithm PFCR1 is at least ten times faster than Algorithm PFchol at achieving the required accuracy in the relative duality gap, even if there is enough memory to store M. Comparing Tables 2 and 6, we see that the number of CR steps needed to solve (20) is far less than that required by (7). But because computing a matrix-vector product for (20) is much more expensive, the savings in CPU time is not as impressive as the reduction in the number of CR steps.

Observe that although we need to store the dense $\bar{p} \times \bar{p}$ matrix $\widehat{H}$, we are able to solve SDPs with m much larger than that allowed by the direct method. This is especially true when $\bar{p}$ is much smaller than m. But we should note that, although storing

ALGORITHM PFCR1. Suppose we are given an initial iterate (X^0, y^0, Z^0) with X^0, Z^0 positive definite. Select $\kappa = 0.01$. Set $\gamma^0 = 0.9$ and $\sigma^0 = 0.5$.

For $k = 0, 1, \ldots$

Let the current and the next iterate be (X, y, Z) and (X^+, y^+, Z^+), respectively. Also, let the current and the next step-length (centering) parameter be denoted by γ and γ^+ (σ and σ^+), respectively.

1. Set $\mu = X \bullet Z/n$ and
$$\phi = \max\left(\frac{\|r_p\|}{\max(1, \|b\|)}, \frac{\|R_d\|_F}{\max(1, \|C\|_F)}\right).$$
Stop the iteration if the infeasibility measure ϕ and the duality gap $X \bullet Z$ are sufficiently small.
2. Compute the Nesterov–Todd scaling matrix W and its eigenvalue decomposition $W = QDQ^T$. Let $d = \mathrm{diag}(D)$, where d is sorted in ascending order.
If $\max(d)/\min(d) > 10^3$
choose p to be the integer such that d_{p+1}/d_p is the maximum,
else
set $p = 0$,
end
3. (a) If $p = 0$,
compute an admissible search direction $(\Delta X, \Delta y, \Delta Z)$ using the CR method via (7).
(b) If $p > 0$,
compute an admissible search direction $(\Delta X, \Delta y, \Delta Z)$ using the CR method via the projected SCE (20).
4. Update (X, y, Z) to (X^+, y^+, Z^+) by
$$X^+ = X + \alpha\,\Delta X, \quad y^+ = y + \beta\,\Delta y, \quad Z^+ = Z + \beta\,\Delta Z,$$
where
$$\alpha = \min\left(1, \frac{-\gamma}{\lambda_{\min}(X^{-1}\Delta X)}\right), \quad \beta = \min\left(1, \frac{-\gamma}{\lambda_{\min}(Z^{-1}\Delta Z)}\right).$$
(Here $\lambda_{\min}(U)$ denotes the minimum eigenvalue of U; if the minimum eigenvalue in either expression is positive, we ignore the corresponding term.)
5. Update the step-length parameter by
$$\gamma^+ = 0.9 + 0.08\min(\alpha, \beta)$$
and the centering parameter by $\sigma^+ = 1 - 0.9\min(\alpha, \beta)$.

$\widehat{H}$ requires significantly less memory than storing M for the SDPs we considered here, we may again face the problem of having insufficient memory when $\bar{p}$ is large, as can be seen from the last problem in Table 6. Based on the information obtained from the experiment in Table 2 for the problem with $m = 23872$, $\bar{p}$ is approximately equal to 9453, and storing $\widehat{H}$ would require about 715MB of memory. Although this is much smaller than the 4560MB of memory needed to store M, it still exceeds the limit of 256MB that our computer has.

In a future paper, we will present a method in which $\widehat{H}$ is not formed explicitly to avoid possible memory problems. More specifically, we will solve the linear system $\widehat{H}z = f$ using the CR method instead of using the Cholesky factorization.

In the context of this paper, we are primarily concerned with SDPs where $m \gg n$. However, for the sake of gaining more computational experience in solving the SCE by the CR method and by the method proposed in this section, we will also consider problems where $m \approx n$ are both large. As we have noted before, primal-dual interior-point methods are not the most appropriate methods to use when $m \approx n$ are both large, because the memory required by the primal variable is as acute as the Schur

TABLE 6

Comparison between Algorithm PFchol and PFCR1. The latter uses the CR method to solve either (7) or (20), depending on the condition number of W in each iteration. A "0" in the last column means that (7) is used; otherwise (20) is used. Note that for the last problem the information on p was obtained from the experiment conducted in Table 2.

		Solving (3a) via Cholesky factorization of M				Solving (3a) via the CR method on (7) / (20)					
n	m	Iter. no.	CPU time	$X \bullet Z$	Mem. MB	Iter. no.	CPU time	$X \bullet Z$	Mem. MB	$NCR(k)$	p
200	1949	18	12:47	$3.6e-1$	80	18	2:43	$3.3e-1$	20	110	20
		21	14:55	$3.9e-3$	80	21	4:33	$3.8e-3$	22	125	32
		24	17:02	$3.6e-5$	80	24	7:31	$2.4e-5$	22	215	32
200	5986	15	3:58:30		287	15	4:48	$1.7e-1$	22	257	0
		18	4:46:12		287	18	20:13	$1.5e-3$	63	195	67
		21	5:33:54		287	21	56:50	$1.2e-5$	65	530	69
300	4375	20	2:35:05	$2.1e-1$	219	20	11:28	$2.3e-1$	38	75	41
		23	2:57:18	$1.2e-3$	219	23	21:05	$1.3e-3$	40	240	47
		26	3:19:23	$1.2e-5$	219	26	45:23	$1.3e-5$	41	735	48
300	13390	15	44:28:15		1434	15	20:32	$2.3e-1$	45	297	0
		18	53:21:54		1434	18	1:52:10	$2.9e-3$	242	165	100
		21	62:15:33		1434	21	4:18:52	$2.7e-5$	258	185	106
400	7905	20	12:12:00		500	20	32:58	$2.3e-1$	65	65	58
		23	14:01:48		500	23	1:03:25	$1.5e-3$	82	235	65
		25	15:15:00		500	25	1:38:52	$4.4e-5$	82	505	66
400	23872					15					0
						18	insufficient memory to store the				131
						21	matrix $\widehat{H}$ in (20)				137

complement matrix. (The appropriate method for this class of SDPs is the dual scaling method described in [3].) Thus our main concern for these problems now is not alleviating the memory problem but increasing the speed at which the SCE can be solved by the CR method.

Table 7 compares the performance of PFchol, PFCR, and PFCR1 on a number of SDPs arising from MAXCUT problems for random graphs. In the table, we report the cumulative CPU time taken to solve the SCE alone (including the time to form M if it is needed explicitly) via the Cholesky factorization, the CR method on (7), and the hybrid combination of the CR method on (7) and (20). We stop the interior-point iterations when the relative duality gaps are less than 10^{-6}.

The most important observation we want to make from Table 7 is that the number of CR steps needed to solve (20) is far less than that needed to solve (7), although each step of the former is more expensive.

Notice that for PFchol, when there is enough memory to store the Schur complement matrix M, solving (3a) by the direct method takes far less CPU time than using an iterative method for the SDPs we tested. This can be attributed to two facts. First, the SDPs arising from the MAXCUT problems are highly sparse and the formation of M can be done efficiently. Second, the Cholesky factorization routine we used is a routine highly optimized for computer execution (about twice as fast as MATLAB's `chol`) compared to the iterative routines we implemented. However, we would expect these advantages of the direct method to diminish when the dimensions m, n of the SDPs increase, since computing the Cholesky factorization of M then becomes prohibitively expensive. But due to the huge memory requirement of the

primal variable, we are unable to test problems where n is much larger than 1000 in a primal-dual interior-point framework.

As mentioned earlier, the projected SCE proposed in this paper can be adapted to solve the SCE associated with a dual scaling interior-point method. Based on the computational experiences that have been reported in [3] and [5] for SDPs arising from MAXCUT problems, our guess is that it is very likely that the method we propose here can efficiently solve the SCE associated with the dual scaling interior-point method for this class of problems.

TABLE 7

*Comparison of Algorithms PFchol, PFCR, and PFCR*1 *on a number of SDPs arising from MAXCUT problems (values are shown for* (3a)*).*

			Solving via Cholesky factorization of M		Solving via the CR method on (7)			Solving via the CR method on (7) / (20)			
n	m	Iter. no.	CPU time	$X \bullet Z$	CPU time	$X \bullet Z$	$NCR(k)$	CPU time	$X \bullet Z$	$NCR(k)$	p
300	300	11	9	$4.0e+1$	9	$2.5e+1$	27	9	$2.5e+1$	27	0
		14	12	$1.0e+0$	26	$3.6e-1$	88	25	$3.7e-1$	16	8
		17	15	$1.0e-3$	73	$4.6e-4$	296	46	$4.3e-4$	26	8
500	500	14	74	$5.1e+1$	56	$1.3e+1$	75	63	$1.3e+1$	75	0
		17	91	$1.2e+0$	194	$2.3e-1$	380	217	$2.3e-1$	380	0
		20	108	$1.3e-2$	718	$4.4e-3$	1726	355	$4.3e-3$	37	11
1000	1000	12	65	$3.4e+2$	94	$2.9e+2$	26	93	$2.9e+2$	26	0
		15	81	$7.0e+0$	281	$8.5e+0$	114	278	$8.5e+0$	114	0
		18	98	$1.4e-2$	1176	$1.1e-2$	569	740	$9.2e-3$	41	14

7. Conclusion and discussion. We have introduced a decomposition of the SCE and converted it into what we call the projected SCE. Numerical experiments on SDPs arising from theta and MAXCUT problems were conducted to show that when the CR method is applied to this equation, moderately accurate solutions can be obtained at reasonable cost. The proposed method is well suited for a primal and dual nondegenerate problem that satisfies the strict complementarity condition and that has an optimal primal solution whose rank is small.

The research work in this paper has generated a number of problems that deserve further investigation in the future. They include the following.

First, the applicability of the projected Schur complement approach remains to be tested for wider classes of SDPs. For example, in a recent work [8] it was shown that a large sparse SDP can sometimes be converted into one with a block diagonal structure by increasing the number of linear constraints. It will be interesting to adapt the method developed in this paper to solve that type of SDPs.

Another area that requires further work is a careful implementation of the projected SCE to fully exploit any sparsity and special structures present in the SDP data. This is especially important for large SDPs such as those arising from graph partition problems in which the data involves dense matrices of small ranks.

We would also want to investigate the performance of the projected SCE on large SDPs arising from MAXCUT problems in a dual scaling interior-point method. Based on extensive computational experiences reported in [3] and [5] for SDPs arising from MAXCUT problems, our guess is that the method we proposed in this paper could solve such problems efficiently. But that remains to be verified.

As the reader may recall, in each step of the CR method for the projected SCE we need to solve a dense $\bar{p} \times \bar{p}$ linear system of equations $\widehat{H}z = f$. Though $\bar{p}$ is generally much smaller than m, it can still be large, and memory problems may again arise. One possible way to overcome such problems is to solve the system by an iterative method. Our preliminary work has shown that this linear system can be solved without much difficulty by the CR method.

Lastly, the most difficult problem we have to face is perhaps to extend the ideas in this paper in order to solve degenerate SDPs.

Acknowledgments. Part of this research was done while the first author was visiting the Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Tokyo, Japan. He thanks his colleagues there for their hospitality and for providing an excellent research environment during his visit.

REFERENCES

[1] F. Alizadeh, J. A. Haeberly, and M. Overton, *Complementarity and nondegeneracy in semidefinite programming*, Math. Programming, 77 (1997), pp. 111–128.

[2] B. Borchers, *SDPLIB* 1.2*, A library of semidefinite programming test problems. Interior point methods*, Optim. Methods Softw., 11/12 (1999), pp. 683–690; also available online from http://www.nmt.edu/˜borchers/sdplib.html.

[3] S. J. Benson, Y. Ye, and X. Zhang, *Solving large-scale sparse semidefinite programs for combinatorial optimization*, SIAM J. Optim., 10 (2000), pp. 443–461.

[4] P. N. Brown and H. F. Walker, *GMRES on (nearly) singular systems*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 37–51.

[5] C. Choi and Y. Ye, *Solving Sparse Semidefinite Programs Using the Dual Scaling Algorithm with an Iterative Solver*, working paper, Computational Optimization Laboratory, University of Iowa, Iowa City, IA, 2000.

[6] T. A. Driscoll, K. C. Toh, and L. N. Trefethen, *From potential theory to matrix iterations in six steps*, SIAM Rev., 40 (1998), pp. 547–578.

[7] K. Fujisawa, M. Kojima, and K. Nakata, *Exploiting sparsity in primal-dual interior-point methods for semidefinite programming*, Math. Programming, 79 (1997), pp. 235–253.

[8] M. Fukuda, M. Kojima, K. Murota, and K. Nakata, *Exploiting sparsity in semidefinite programming via matrix completion* I: *General framework*, SIAM J. Optim., 11 (2000), pp. 647–674.

[9] A. Greenbaum, *Estimating the attainable accuracy of recursively computed residual methods*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 535–551.

[10] C. Helmberg and K. C. Kiwiel, *A spectral bundle method with bounds*, Math. Program., to appear.

[11] C. Helmberg, F. Rendl, R. J. Vanderbei, and H. Wolkowicz, *An interior-point method for semidefinite programming*, SIAM J. Optim., 6 (1996), pp. 342–361.

[12] M. Kojima, S. Shindoh, and S. Hara, *Interior-point methods for the monotone semidefinite linear complementarity problem in symmetric matrices*, SIAM J. Optim., 7 (1997), pp. 86–125.

[13] M. Kojima, M. Shida, and S. Shindoh, *Search directions in the SDP and the monotone SDLCP: Generalization and inexact computation*, Math. Program., 85 (1999), pp. 51–80.

[14] C.-J. Lin and R. Saigal, *On Solving Large-Scale Semidefinite Programming Problems—A Case Study of Quadratic Assignment Problem*, Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI, manuscript.

[15] R. D. C. Monteiro, *Primal–dual path-following algorithms for semidefinite programming*, SIAM J. Optim., 7 (1997), pp. 663–678.

[16] K. Nakata, K. Fujisawa, and M. Kojima, *Using the conjugate gradient method in interior-points methods for semidefinite programs* (in Japanese), Proc. Inst. Statist. Math., 46 (1998), pp. 297–316.

[17] K. Nakata, S.-L. Zhang, and M. Kojima, *Preconditioned Conjugate Gradient Methods for Large Scale and Dense Linear Systems in Semidefinite Programming*, abstract based on talks delivered at INFORMS Meeting, Philadelphia, 1999.

[18] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, Boston, 1996.

[19] Y. Saad, M. Yeung, J. Erhel, and F. Guvomarc'h, *A deflated version of the conjugate gradient algorithm*, SIAM J. Sci. Comput., 21 (2000), pp. 1909–1926.

[20] Z. Strakos, *On the real convergence rate of the conjugate gradient method*, Linear Algebra Appl., 154–156 (1991), pp. 535–549.

[21] M. J. Todd, K. C. Toh, and R. H. Tütüncü, *On the Nesterov–Todd direction in semidefinite programming*, SIAM J. Optim., 8 (1998), pp. 769–796.

[22] K. C. Toh, M. J. Todd, and R. H. Tütüncü, *SDPT3—a* Matlab *software package for semidefinite programming, version* 1.3, Optim. Methods Softw., 11 (1999), pp. 545–581; also available online from http://www.math.nus.sg/~mattohkc.

[23] Y. Zhang, *On extending some primal-dual interior-point algorithms from linear programming to semidefinite programming*, SIAM J. Optim., 8 (1998), pp. 365–386.

[24] Q. Zhao, S. E. Karisch, F. Rendl, and H. Wolkowicz, *Semidefinite Programming Relaxations for the Quadratic Assignment Problem*, J. Comb. Optim., 2 (1998), pp. 71–109.

SIAM J. OPTIM.
Vol. 12, No. 3, pp. 692–714

AN INTERIOR-POINT APPROACH TO SENSITIVITY ANALYSIS IN DEGENERATE LINEAR PROGRAMS*

E. ALPER YILDIRIM† AND MICHAEL J. TODD‡

Abstract. We consider an interior-point approach to sensitivity analysis in linear programming developed by the authors. We investigate the quality of the interior-point bounds under degeneracy. In the case of a special type of degeneracy, we show that these bounds have the same nice asymptotic relationship with the optimal partition bounds as in the nondegenerate case. We prove a weaker relationship for general degenerate linear programs.

Key words. sensitivity analysis, degeneracy, interior-point methods, linear programming

AMS subject classifications. 90C31, 90C51, 90C05

PII. S1052623400382455

1. Introduction.

Sensitivity analysis (or postoptimality analysis) is the study of how the optimal solution of an optimization problem changes with respect to changes in the problem data. The possible presence of errors in the problem data often makes sensitivity analysis as important as solving the original problem itself.

In the context of linear programming, sensitivity analysis can be performed using an optimal basis approach (as in the simplex method) or an optimal partition approach, where the optimal partition refers to knowing, for each index, whether the corresponding component of an optimal primal solution or of an optimal dual slack vector can be positive. The latter approach has close connections with interior-point methods since such methods, when properly terminated, provide an optimal solution in the relative interior of the optimal face, from which the optimal partition is readily available. In fact, as will shortly be discussed in more detail, the optimal partition approach has been developed by Adler and Monteiro [1] and Jansen, de Jong, Roos, and Terlaky [7] as a promising alternative, in order to circumvent the drawbacks of the classical optimal basis approach in the presence of degeneracy. Later, Monteiro and Mehrotra [9] extended this approach by relaxing the requirement that the optimal partition be known. They also provided two methods for estimating the range of perturbations, each of which can be performed at any optimal solution, regardless of where it lies on the optimal face. More recently, Greenberg, Holder, Roos, and Terlaky [5] related the dimension of the optimal set to the dimension of the set of objective perturbations for which the optimal partition is invariant. Greenberg [4] considered the simultaneous perturbations of the right-hand side and the cost vectors in linear programming from an optimal partition perspective.

In [13], the authors studied perturbations of the right-hand side and the cost parameters in linear programming, motivated by the way in which interior-point methods from a near-optimal pair of strictly feasible solutions for a problem and its dual would

*Received by the editors December 13, 2000; accepted for publication (in revised form) August 14, 2001; published electronically February 8, 2002. This research was supported in part by NSF grant DMS-9805602 and ONR grant N00014-96-1-0050.

http://www.siam.org/journals/siopt/12-3/38245.html

†Department of Applied Mathematics and Statistics, State University of New York at Stony Brook, Stony Brook, NY 11794 (yildirim@ams.sunysb.edu). This research was performed as part of this author's Ph.D. study at Cornell University.

‡School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY 14853 (miketodd@cs.cornell.edu).

compare with the optimal basis approach obtained from a nondegenerate optimal basic solution for such perturbations. The proposed interior-point perspective results from the objectives of regaining feasibility and maintaining near-optimality in a *single* iteration of the interior-point method. This requires the setup of the "right" Newton system among many possible choices in order to achieve both objectives simultaneously. Such a perspective provides a basis for the comparison of the interior-point and the simplex approaches to sensitivity analysis.

Under the assumption of a unique, nondegenerate optimal solution, the authors show that the Newton system proposed in [13] is the "right" one in the sense that it yields asymptotically the same bounds on perturbations as those that keep the current basis optimal (after symmetrization with respect to the origin). Similar results, changing only one of the primal or dual near-optimal solutions, were obtained by Kim, Park, and Park [8].

However, most linear programs (LPs) arising from real-life problems are degenerate. Our goal in this paper is to investigate the quality of the bounds from the interior-point perspective in the absence of the strong assumption of nondegeneracy. This will lead to a complete analysis of the interior-point perspective proposed in [13]. In such an analysis, we need something to compare our interior-point bounds with. In contrast to the nondegenerate case, the presence of multiple optimal bases makes a simplex-based approach unsuitable, as will be explained shortly. We therefore compare our bounds to those obtained from consideration of how much the right-hand side or the cost vector can change while maintaining the same optimal partition. Consequently, we use different tools for our analysis in this paper.

The next section is devoted to preliminaries, including introduction of the tools relevant for the analysis as well as a restatement of our interior-point approach. Section 3 discusses the equivalence between the primal and dual formulations and shows that it suffices to consider perturbations of the right-hand side only. We analyze the interior-point bounds under a special case of degeneracy in section 4 and extend the analysis to the general degenerate case in section 5. We apply our interior-point approach to a well-documented, degenerate transportation example in section 6 and conclude the paper in section 7.

2. Preliminaries. We consider LP in the following standard form:

$$\text{(P)} \quad \min_x \; c^T x \quad \text{subject to } Ax = b, \quad x \geq 0.$$

The associated dual LP is given by

$$\text{(D)} \quad \max_{y,s} \; b^T y \quad \text{subject to } A^T y + s = c, \quad s \geq 0.$$

Here, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, and $c \in \mathbb{R}^n$ constitute the data, and $(x, y, s) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ are the decision variables. Throughout this paper, the coefficient matrix A will be fixed and we will consider one-dimensional perturbations of the right-hand side vector b and the cost vector c; i.e., b will be replaced by $b + t\Delta b$ and c by $c + t\Delta c$, where Δb and Δc will be fixed in $\mathbb{R}^m$ and $\mathbb{R}^n$, respectively, and $t \in \mathbb{R}$ will be the parameter. This is also called *parametric analysis* in the literature.

We will make the following assumptions.

Assumption 2.1. The coefficient matrix A has full row rank.

Assumption 2.2. Both (P) and (D) have strictly feasible solutions; i.e., there exist $x > 0$, $s > 0$, and y such that $Ax = b$ and $A^T y + s = c$.

While Assumption 2.1 is without loss of generality, Assumption 2.2 is clearly restrictive. However, we will discuss ways in which our approach can be extended to LPs that do not satisfy Assumption 2.2 (but do have optimal solutions) at the end of section 2.2.

The classical approach to sensitivity analysis has been based on the simplex method. Assuming that an optimal solution exists, the simplex method terminates with a basic optimal solution along with a corresponding basis. A natural criterion for the allowable perturbations in the data is then given by the range of such perturbations for which the current basis remains optimal for the resulting family of LPs.

Let us consider the parametric right-hand side (RHS) problem, i.e., let b be replaced by $b + t\Delta b$. Define $v(t) = \min\{c^T x : Ax = b + t\Delta b,\, x \geq 0\}$. It is well known that v is a convex, piecewise linear, continuous function of t. The parametric RHS problem includes finding all the "breakpoints" of $v(t)$.

Fixing a value of t, say at 0 for the purposes of this paper, the classical approach to sensitivity analysis then provides the set of values of t for which an optimal basis for $t = 0$ remains optimal for the resulting LPs parametrized by t. This is called the *optimality interval* associated with an optimal basis. Note that the optimal basis approach indeed yields the breakpoints of $v(t)$ around 0 under primal and dual nondegeneracy (which holds only if 0 itself is not a breakpoint of $v(t)$). However, the presence of primal and/or dual degeneracies is a shortcoming for this approach since, for example, multiple optimal bases might yield different optimality intervals. This shortcoming has been observed by several researchers. Adler and Monteiro [1], and Jansen, de Jong, Roos, and Terlaky [7] developed an optimal partition approach to sensitivity analysis and showed that the optimality intervals associated with the optimal partitions uniquely and unambiguously identify the breakpoints of $v(t)$ and the intervals between the consecutive breakpoints. By the symmetry between (P) and (D), which will be treated in more detail in section 3, the same conclusions also hold for the parametric analysis of the cost vector c.

The idea of the optimal partition is based on a well-known result of Goldman and Tucker [2]. The optimality conditions for (P) and (D) are given by primal and dual feasibility and complementary slackness; that is, a triple (x, y, s) is optimal for (P) and (D) if and only if it satisfies

$$Ax = b, \quad A^T y + s = c, \quad x_i s_i = 0, \qquad i = 1, \ldots, n, \quad x \geq 0,\ s \geq 0, \tag{2.1}$$

where x_i and s_i denote the ith components of x and s, respectively. Let Ω_P and Ω_D denote the set of optimal solutions for (P) and (D), respectively. Then, we can define two index sets as

$$\begin{aligned} \mathcal{B} &= \{j \in \{1, \ldots, n\} : x_j > 0 \text{ for some } x \in \Omega_P\}, \\ \mathcal{N} &= \{j \in \{1, \ldots, n\} : s_j > 0 \text{ for some } (y, s) \in \Omega_D\}. \end{aligned} \tag{2.2}$$

The optimality conditions (2.1) imply that $\mathcal{B} \cap \mathcal{N} = \emptyset$. The Goldman–Tucker result indicates that $\mathcal{B}$ and $\mathcal{N}$ actually partition the index set $\{1, \ldots, n\}$, i.e., $\mathcal{B} \cup \mathcal{N} = \{1, \ldots, n\}$. Therefore, there exist at least one primal optimal solution $x \in \Omega_P$ and one dual optimal solution $(y, s) \in \Omega_D$ such that $x + s > 0$. Such a solution will be called *strictly complementary*, and $(\mathcal{B}, \mathcal{N})$ will be called the *optimal partition*. In contrast to the possibility of multiple optimal bases, the optimal partition is unique for a given LP instance.

We will denote by B and N the columns of A corresponding to the indices in $\mathcal{B}$ and $\mathcal{N}$, respectively, and we will also partition the cost vector c as c_B and c_N, and the variables x and s as x_B and x_N, and s_B and s_N, accordingly. Note that if (x, y, s) is a strictly complementary solution, then we have $x_B > 0$, $x_N = 0$, $s_B = 0$, and $s_N > 0$.

Let us again restrict our attention to one-dimensional perturbations of the right-hand side vector b. The optimal partition approach is based on maintaining the whole dual optimal set invariant rather than an optimal basis as in the classical simplex approach. Note that perturbations of b do not affect the dual feasible region. Consequently, the range of t is given by solving two auxiliary LPs. More precisely, if b is replaced by $b + t\Delta b$, and if Ω_D denotes the dual optimal set for (D) (i.e., $t = 0$), then the lower and upper bounds on t are given by the optimal values of

$$\begin{array}{lll} \text{(AUX1)} & \min_{x,\lambda}\ (\max_{x,\lambda}) & \lambda \\ & \text{subject to} & \begin{aligned} Ax &= b + \lambda\Delta b, \\ x &\geq 0, \\ (s^*)^T x &= 0 \quad \forall\, (y^*, s^*) \in \Omega_D. \end{aligned} \end{array}$$

We will call the resulting bounds the *optimal partition bounds*. Note that both problems are always feasible, since $\lambda = 0$ together with any $x \in \Omega_P$ satisfy all the constraints. Fixing the dual optimal set Ω_D is equivalent to fixing the optimal partition $(\mathcal{B}, \mathcal{N})$ by the Goldman–Tucker result. Therefore, the (possibly infinite) last constraint set in (AUX1) can be replaced by the equivalent single constraint $x^T s^* = 0$, where s^* is any point in the relative interior of Ω_D (hence $s^*_N > 0$). This condition, in turn, is the same as setting $x_N = 0$. Consequently, (AUX1) can be written in the following simplified form:

$$\begin{array}{lll} \text{(AUX1}') & \min_{x_B,\lambda}\ (\max_{x_B,\lambda}) & \lambda \\ & \text{subject to} & \begin{aligned} Bx_B &= b + \lambda\Delta b, \\ x_B &\geq 0. \end{aligned} \end{array}$$

The analogous derivation for the one-dimensional perturbations of the cost vector c leads to the following auxiliary problems, whose optimal values give the optimal partition bounds for t when c is replaced by $c + t\Delta c$:

$$\begin{array}{lll} \text{(AUX2)} & \min_{y,s_N,\lambda}\ (\max_{y,s_N,\lambda}) & \lambda \\ & \text{subject to} & \begin{aligned} B^T y &= c_B + \lambda\Delta c_B, \\ N^T y + s_N &= c_N + \lambda\Delta c_N, \\ s_N &\geq 0. \end{aligned} \end{array}$$

Here, Δc_B and Δc_N constitute the corresponding partition of Δc.

Before getting into the symmetrized bounds, we would like to recall an important result about the dimensions of the optimal solution sets Ω_P and Ω_D. In what follows, $\dim(\cdot)$ denotes the dimension and $|\cdot|$ denotes the cardinality of a set. The reader is referred to Lemma IV.44 in [10] for a proof.

PROPOSITION 2.1. *The following hold:* $\dim(\Omega_P) = |\mathcal{B}| - \operatorname{rank}(B)$; $\dim(\Omega_D) = m - \operatorname{rank}(B)$. □

2.1. Symmetrized bounds. The auxiliary problems (AUX1) and (AUX2) can be reformulated in the following way. Let us consider (AUX1′) and let $x^* \in \Omega_P$. Then, the equality constraint can be rewritten as

$$Bx_B = Bx^*_B + \lambda\Delta b \quad \text{or} \quad B(x_B - x^*_B) = \lambda\Delta b.$$

Therefore, by a change of variable, if we let $u = x_B - x_B^*$, then (AUX1) is equivalent to

$$\begin{array}{lll} (\text{AUX1}'') & \min_{u,\lambda}\ (\max_{u,\lambda}) & \lambda \\ & \text{subject to} & Bu = \lambda \Delta b, \\ & & u \geq -x_B^*. \end{array}$$

Next, we will tighten the constraints in the above formulation by putting upper bounds on u as well, and our choice for the upper bound will be x_B^*, which will give the largest symmetric L_∞-box around the origin which is contained in the feasible region:

$$\begin{array}{lll} (\text{SA1}) & \min_{u,\lambda}\ (\max_{u,\lambda}) & \lambda \\ & \text{subject to} & Bu = \lambda \Delta b, \\ & & -x_B^* \leq u \leq x_B^*. \end{array}$$

We will call (SA1) the *symmetrized LP* and the resulting optimal solutions the *symmetrized bounds*. The formulation of (SA1) reveals that if (u^*, λ^*) solves the maximization problem, then $(-u^*, -\lambda^*)$ solves the minimization problem. Therefore, it suffices to solve one LP as opposed to solving two LPs to obtain the optimal partition bounds from (AUX1). A similar treatment of (AUX2) gives rise to the following symmetrized LP:

$$\begin{array}{lll} (\text{SA2}) & \min_{v,w,\lambda}\ (\max_{v,w,\lambda}) & \lambda \\ & \text{subject to} & B^T v = \lambda \Delta c_B, \\ & & N^T v + w = \lambda \Delta c_N, \\ & & -s_N^* \leq w \leq s_N^*, \end{array}$$

which is obtained by replacing $y - y^*$ by v, and $s_N - s_N^*$ by w, where $(y^*, s^*) \in \Omega_D$. Finally, a similar symmetrization has been applied to w.

Next, we would like to discuss the relationship between the auxiliary and the symmetrized LPs. First of all, let us assume that both (P) and (D) have unique and nondegenerate optimal solutions. Then, Proposition 2.1 implies that B is actually a square and nonsingular matrix, hence invertible. In fact, B is the optimal basis. Consequently, (AUX1) and (AUX2) are trivial to solve and their optimal solutions coincide with the optimal basis bounds arising from the simplex method. With this observation, the constraints of (AUX1$''$) reduce to

$$\lambda B^{-1} \Delta b \geq -x_B^* \quad \text{or} \quad \lambda (X_B^*)^{-1} B^{-1} \Delta b \geq -e, \tag{2.3}$$

where X_B^* is the diagonal matrix whose components are given by x_B^* and e denotes the vector of ones in the appropriate dimension. Similarly, the constraints of (SA1) can be rewritten as

$$-e \leq \lambda (X_B^*)^{-1} B^{-1} \Delta b \leq e \quad \text{or} \quad |\lambda| \|(X_B^*)^{-1} B^{-1} \Delta b\|_\infty \leq 1, \tag{2.4}$$

where $\|\cdot\|_\infty$ is the L_∞-norm. A similar treatment reveals that the constraints of (AUX2) are equivalent to

$$\lambda (S_N^*)^{-1} (\Delta c_N - N^T B^{-T} \Delta c_B) \geq -e, \tag{2.5}$$

where S_N^* is defined similarly, and that those of (SA2) are equivalent to

$$|\lambda| \left\| (S_N^*)^{-1} (\Delta c_N - N^T B^{-T} \Delta c_B) \right\|_\infty \leq 1. \tag{2.6}$$

The derivations (2.3)–(2.6) imply the following relationship between the auxiliary and the symmetrized LPs. Let (λ^-, λ^+) denote the optimal partition bounds given by the optimal solutions of the auxiliary LPs (possibly including $\pm\infty$). Then, the symmetrized bounds for t are $(-\lambda^s, \lambda^s)$, where

$$\lambda^s = \min(|\lambda^-|, \lambda^+). \tag{2.7}$$

Therefore, the symmetrized bounds are indeed equal to the "symmetrization" of the optimal partition bounds.

Note that we used an open interval for the optimal partition bounds above. The reason for this is that the optimal partition remains the same in this open interval, whereas it does change at the left and right limit points (assuming they are finite).

Next, let us assume that (P) has a unique but degenerate optimal solution. Then, by Proposition 2.1, B is nonsquare but has full column rank. Therefore, (AUX1″) is still easy to solve. If Δb does not lie in the range space of B, then the optimal solutions of (AUX1″) and (SA1) are all zero (which implies that $t = 0$ is a breakpoint of $v(t)$). Otherwise, there exists a unique vector v such that $Bv = \Delta b$, and hence, the constraints of (AUX1″) are equivalent to

$$\lambda(X_B^*)^{-1}v \geq -e. \tag{2.8}$$

Similarly, the constraints of (SA1) can be stated as

$$|\lambda|\|(X_B^*)^{-1}v\|_\infty \leq 1. \tag{2.9}$$

Once again, we conclude that a similar symmetry as in (2.7) continues to hold between (SA1) and (AUX1″). In a similar manner, one can show that such a relationship holds between (SA2) and (AUX2) if (D) has a unique but degenerate optimal solution.

The preceding discussion shows that the optimal solutions of the auxiliary and the symmetrized LPs have the nice relationship (2.7) as long as there is a unique optimal solution that one can use to symmetrize the constraints of the auxiliary LPs to obtain the symmetrized LPs. An interesting question then is whether the same relationship continues to hold between the optimal partition bounds and the symmetrized bounds if there are multiple optimal solutions; that is, whether the symmetrized bounds are independent of the choice of the optimal solution used to symmetrize the constraints. Unfortunately, the answer is no, as is shown by the following example. Let (P) be given by $\min\{x_2 - x_1 : x_1 - x_2 = 0,\ x_2 + x_3 = 1,\ x \geq 0\}$. Then (P) has multiple optimal solutions given by $(x_1, x_2, x_3) = (\beta, \beta, 1-\beta)$, where $\beta \in [0, 1]$, with an optimal value of 0. If the right-hand side is perturbed to $(0, 1)^T + t\,(2, 1)^T$, then the reader can easily verify that (AUX1) yields $(-1/3, +\infty)$ as the optimal partition bounds, whereas the symmetrized bounds are $(-\beta, +\beta)$ if one uses the optimal solutions with $\beta < 1/3$ to symmetrize the constraints, and $(-1/3, 1/3)$ if those with $\beta \geq 1/3$ are used. This example illustrates that in the case of multiple optimal solutions, the symmetrized bounds are dependent on the optimal solution used in the formulation of the symmetrized LPs. Therefore, the relationship (2.7) no longer holds between the optimal partition bounds and the symmetrized bounds.

However, we will keep using the symmetrized LPs for two reasons. First, at least in the unique solution case, they bear a nice relationship to the auxiliary LPs. For our analysis, we will always choose an optimal solution in the relative interior of the optimal set; therefore the symmetrization will hopefully allow more room for the decision variables of the symmetrized LPs. Second, the symmetrized LPs are easier

to deal with than the auxiliary LPs, and the symmetrized bounds will provide a good comparison basis for our interior-point approach proposed in [13], as will be analyzed in the subsequent sections.

2.2. Interior-point approach and central path neighborhoods. We will start with a brief review of the primal-dual path-following interior-point methods. The reader is referred to [11] for an extensive treatment. The central path is a path of strictly feasible points $(x(\nu), y(\nu), s(\nu))$ parametrized by a positive scalar ν. Each point on the central path satisfies the following system for some $\nu > 0$:

$$\begin{aligned} Ax &= b, \\ A^T y + s &= c, \\ XSe &= \nu e, \end{aligned} \tag{2.10}$$

with $x > 0$ and $s > 0$. Under Assumptions 2.1 and 2.2, such a solution exists and is unique for each positive ν. Interior-point methods are iterative algorithms that generate iterates which "follow" the central path in the direction of decreasing ν towards the primal-dual optimal set $\Omega_P \times \Omega_D$. The iterates generated typically lie in some neighborhood of the central path. For any given feasible iterate (x, y, s), the duality gap is given by $c^T x - b^T y = x^T s \geq 0$, and we define the duality measure μ as $\mu := \mu(x, s) := x^T s/n$. Let $\mathcal{S}$ and $\mathcal{S}^0$ denote the set of feasible and strictly feasible primal-dual points, respectively; that is,

$$\mathcal{S} = \{(x, y, s) \ : \ Ax = b, \ A^T y + s = c, \ (x, s) \geq 0\}, \tag{2.11}$$

$$\mathcal{S}^0 = \{(x, y, s) \in \mathcal{S} \ : \ (x, s) > 0\}. \tag{2.12}$$

One of the commonly used neighborhoods in interior-point methods is the so-called *wide neighborhood*, denoted by $\mathcal{N}_{-\infty}(\gamma)$:

$$\mathcal{N}_{-\infty}(\gamma) = \{(x, y, s) \in \mathcal{S}^0 \ : \ x_i s_i \geq \gamma\mu \ \forall \ i = 1, 2, \ldots, n\}, \tag{2.13}$$

where $\gamma \in (0, 1]$.

At each iteration, given $(x, y, s) \in \mathcal{N}_{-\infty}(\gamma)$, the algorithm determines a search direction $(\Delta x, \Delta y, \Delta s)$. This direction is usually obtained by seeking an approximation to the point on the central path corresponding to some parameter $\nu \leq \mu$ and then applying Newton's method to the nonlinear system of equations (2.10). Finally, a (damped) step is taken in this direction in such a way that the resulting iterate still lies in $\mathcal{N}_{-\infty}(\gamma)$.

However, as in the context of target-following methods [10, Part III], one might seek an approximation to a point other than the one on the central path. It suffices to redefine (2.10) by replacing νe in the right-hand side of the third equality by any *target vector* $v > 0$. In this case, the Newton step at (x, y, s) for the target vector v is given by

$$\begin{aligned} A\Delta x &= b - Ax, \\ A^T \Delta y + \Delta s &= c - A^T y, \\ S\Delta x + X\Delta s &= v - XSe. \end{aligned} \tag{2.14}$$

Next, we describe the interior-point approach proposed by the authors in [13]. Given a primal-dual pair of LPs (P) and (D), let us assume that b or c is perturbed in some fixed direction. Assuming that (x, y, s) is strictly primal-dual feasible for (P) and (D), a full Newton step is taken from (x, y, s) for the target vector $v := XSe$

for the perturbed LP pair, thereby aiming to maintain the current $x_i s_i$ products. If (x, y, s) is near-optimal for (P) and (D), then this particular choice is likely to result in a near-optimal solution for the perturbed LP pair, since $XSe \approx 0$. We state the results formally, referring the reader to [13] for the proofs. Note, in particular, that the duality gap of the resulting feasible iterate for the perturbed LP pair is no greater than that of the original iterate.

PROPOSITION 2.2. *Assume that (x, y, s) is a strictly feasible point for* (P) *and* (D) *and that the right-hand side vector b is replaced by $b + t\Delta b$, where $t \in \mathbb{R}$ and $\Delta b \in \mathbb{R}^m$. Suppose also that a Newton step is taken from (x, y, s) for the target vector $v := XSe$ for the perturbed problem. Then a full Newton step will yield a feasible iterate for the new problem if and only if*

$$| t | \leq \frac{1}{\|S^{-1}A^T(AD^2A^T)^{-1}\Delta b\|_\infty}, \tag{2.15}$$

where $D = X^{\frac{1}{2}}S^{-\frac{1}{2}}$. Moreover, in this case the new iterate will have duality gap at most $x^T s$.

PROPOSITION 2.3. *Assume that (x, y, s) is a strictly feasible point for* (P) *and* (D) *and that the cost vector c is replaced by $c + t\Delta c$, where $t \in \mathbb{R}$ and $\Delta c \in \mathbb{R}^n$. Suppose also that a Newton step is taken from (x, y, s) for the target vector $v := XSe$ for the perturbed problem. Then a full Newton step will yield a feasible iterate for the new problem if and only if*

$$| t | \leq \frac{1}{\|S^{-1}(I - A^T(AD^2A^T)^{-1}AD^2)\Delta c\|_\infty}, \tag{2.16}$$

where $D = X^{\frac{1}{2}}S^{-\frac{1}{2}}$. Moreover, in this case the new iterate will have duality gap at most $x^T s$.

Under primal-dual nondegeneracy, the bounds arising from Propositions 2.2 and 2.3 computed at near-optimal solutions for (P) and (D) asymptotically equal the symmetrized bounds arising from (SA1) and (SA2); see [13]. The goal of this paper is to investigate the quality of these bounds in the absence of the nondegeneracy assumption.

We first present a nice characterization of the distance of the strictly feasible primal-dual points (x, y, s) from strictly complementary optimal solutions in terms of the duality gap μn. Using this characterization, we derive some bounds on the components of such points. In what follows, x_B, x_N, s_B, and s_N denote the partitions of x and s according to the optimal partition $(\mathcal{B}, \mathcal{N})$, as before. Furthermore, we will use the bounds $O(\mu)$, $\Omega(\mu)$, and $\Theta(\mu)$ interchangeably for scalars as well as vectors and matrices, by which we mean each entry satisfies the stated bounds. $O(\mu)$ will indicate that the quantity (in absolute value) is bounded above by some positive multiple of μ, where the multiple depends on the primal-dual instance (P) and (D) but does not depend on the particular strictly feasible point or on μ. Similarly, $\Omega(\mu)$ will indicate a lower bound by some positive multiple of μ, and $\Theta(\mu)$ will mean a lower and upper bound by some positive multiples of μ.

The following proposition will be useful for the analysis that follows. Actually, the proposition continues to hold for any feasible solutions and even for a point where feasibility is violated by $O(\mu)$. The statement below suffices for the purposes of this paper.

PROPOSITION 2.4. *Let (x, y, s) be a strictly feasible point for* (P) *and* (D) *with duality gap μn. Then, there exists a strictly complementary optimal solution (x^*, y^*, s^*)*

of (P) *and* (D) *such that*

$$(x, y, s) = (x^*, y^*, s^*) + O(\mu). \tag{2.17}$$

Proof. Optimal solutions of (P) and (D) satisfy the linear system $Ax = b$, $A^T y + s = c$, $c^T x - b^T y = 0$, $x \geq 0$, $s \geq 0$. Any strictly feasible point (x, y, s) satisfies the same linear system with the third equality replaced by $c^T x - b^T y = \mu n$. Hoffman's lemma [6] indicates that there exists a solution $(\hat{x}, \hat{y}, \hat{s})$ of the first system such that $(\hat{x}, \hat{y}, \hat{s}) = (x, y, s) + O(\mu)$. The result follows immediately if $(\hat{x}, \hat{y}, \hat{s})$ is strictly complementary. If not, there exists an arbitrarily small perturbation of $(\hat{x}, \hat{y}, \hat{s})$ which leads to a strictly complementary solution, and (2.17) follows since $\mu > 0$. ∎

The following corollary immediately follows from Proposition 2.4, since $x^*_N = 0$ and $s^*_B = 0$ for any optimal solution of (P) and (D).

COROLLARY 2.5. *Let* (x, y, s) *be a strictly feasible point for* (P) *and* (D) *with duality gap* μn. *Then,*

$$x_N = O(\mu), \quad s_B = O(\mu). \tag{2.18}$$ ∎

Note that both Proposition 2.4 and Corollary 2.5 hold for *any* primal-dual strictly feasible (x, y, s). Next, we derive some more bounds by restricting the iterates to lie in a wide neighborhood given by (2.13).

PROPOSITION 2.6. *Let* $(x, y, s) \in \mathcal{N}_{-\infty}(\gamma)$ *with duality gap* μn *for* (P) *and* (D). *Then,*

$$XSe = \Theta(\mu), \;\; s_N = \Omega(1), \;\; x_B = \Omega(1), \;\; X_N S_N^{-1} e = O(\mu), \;\; S_B X_B^{-1} e = O(\mu). \tag{2.19}$$

Proof. Since $(x, y, s) \in \mathcal{N}_{-\infty}(\gamma)$, we have $x_i s_i \geq \gamma\mu$. Moreover, $x^T s = \mu n$. Therefore, $x_i s_i \leq \mu n$, since $x > 0$ and $s > 0$. This proves $XSe = \Theta(\mu)$. By Corollary 2.5, $x_N = O(\mu)$. Then, $XSe = \Theta(\mu)$ implies $s_N = \Omega(1)$. A similar argument shows $x_B = \Omega(1)$. Finally, $x_N = O(\mu)$ together with $s_N = \Omega(1)$ imply $X_N S_N^{-1} e = O(\mu)$. The proof of $S_B X_B^{-1} e = O(\mu)$ is similar. ∎

We mentioned in section 2 that Assumption 2.2 is somewhat restrictive. Actually, it is possible to extend our interior-point approach to LP instances where that particular assumption is not satisfied. We simply need to define an appropriate viewpoint. Assume that (x, y, s) is such that $Ax = b + \xi_b$, $A^T y + s = c + \xi_c$, $x > 0$, and $s > 0$ with $\|\xi_b\| = O(\mu)$ and $\|\xi_c\| = O(\mu)$, where μn is the duality gap. (Such a pair of solutions will be generated by several infeasible interior-point methods when applied to problems where optimal solutions exist.) It follows that Proposition 2.4 and Corollary 2.5 hold for such a point, as well as Proposition 2.6 with an appropriate definition of the wide neighborhood. One can then take precisely the same Newton step as before for a perturbed LP pair and obtain an iterate with precisely the same primal and dual infeasibilities with a lower duality gap. Then, the resulting interior-point bound can be interpreted as the range of perturbations for which a single Newton step yields a point with a smaller duality gap for a nearby LP instance of the perturbed problem. The analysis of sections 4 and 5 carry over in this case and will reveal that the interior-point bounds will still be related to the desired symmetrized partition bounds as μ tends to 0.

Another possible extension of our interior-point approach in this case is to try to correct for all of the primal and dual feasibilities in a single Newton step by setting the right-hand side of the first and second equations in (2.14) to $t\Delta b - \xi_b$ and $-\xi_c$,

respectively. The analysis remains unchanged for the case of a unique and nondegenerate primal-dual optimal solution (since B will be nonsingular). However, ξ_b and ξ_c might not be in the "right" spaces in the degenerate case, which would complicate the analysis. Consequently, we adopt the viewpoint described in the previous paragraph for LP instances failing to satisfy Assumption 2.2.

3. Equivalence. In this section, we show that the interior-point bounds are independent of the problem formulation. It is well known that, although (P) and (D) do not look symmetric, they can easily be reformulated so that (D) is in the form of (P) and vice versa [10, pp. 110–112]. More precisely, (D) is equivalent to

$$(\mathrm{D}') \quad \min_{s} \ \hat{x}^T s \quad \text{subject to } Ks = \hat{c}, \quad s \geq 0,$$

where $\hat{x}$ satisfies $A\hat{x} = b$, $\hat{c} := Kc$ and $K \in \mathbb{R}^{(n-m)\times n}$ is a matrix whose rows form a basis for the null space of A. The dual of (D′) is given by

$$(\mathrm{P}') \quad \max_{u,x} \ \hat{c}^T u \quad \text{subject to } K^T u + x = \hat{x}, \quad x \geq 0,$$

which is equivalent to (P).

Let us now focus on perturbations of c, i.e., let c be replaced by $c + t\Delta c$. By the above reformulation, this is the same as replacing the right-hand side of (D′) by $\hat{c} + tK\Delta c$. Therefore, Proposition 2.2 can be used to evaluate the interior-point bound at a strictly feasible primal-dual pair (s, x). (Note that the roles of x and s are interchanged.) We need to compute

$$X^{-1}K^T(KSX^{-1}K^T)^{-1}K\Delta c. \tag{3.1}$$

On the other hand, one can also use Proposition 2.3 to compute the interior-point bound directly at (x, s), which requires the evaluation of

$$S^{-1}(I - A^T(AXS^{-1}A^T)^{-1}AXS^{-1})\Delta c. \tag{3.2}$$

A simple manipulation of (3.1) gives rise to another equivalent formula:

$$X^{-1/2}S^{-1/2}\Psi X^{1/2}S^{-1/2}\Delta c, \tag{3.3}$$

where Ψ is the orthogonal projection matrix onto the range space of $X^{-1/2}S^{1/2}K^T$. Similarly, (3.2) is equivalent to

$$X^{-1/2}S^{-1/2}\Xi X^{1/2}S^{-1/2}\Delta c, \tag{3.4}$$

where Ξ is the orthogonal projection matrix onto the null space of $AX^{1/2}S^{-1/2}$. Therefore, in order to prove that (3.1) and (3.2) are equivalent, it suffices to show that Ψ and Ξ project onto the same subspace, or that the null space of $AX^{1/2}S^{-1/2}$ equals the range space of $X^{-1/2}S^{1/2}K^T$. This is easily proven by an inclusion argument: If w satisfies $AX^{1/2}S^{-1/2}w = 0$, then $X^{1/2}S^{-1/2}w = K^T u$ for some unique u. Thus, w is in the range space of $X^{-1/2}S^{1/2}K^T$. Conversely, if $w = X^{-1/2}S^{1/2}K^T u$ for some u, then $AX^{1/2}S^{-1/2}w = AK^T u = 0$. This proves the equivalence of the interior-point bounds.

We next argue that the range of t resulting from the optimal partition bounds is also independent of the formulation. If the two LPs are formulated in the form of (P)

and (D), then (AUX2) yields the range of t for perturbations of c. Premultiplying the equality constraints of (AUX2) by $K = [K_B, K_N]$ leads to (AUX1′) given by

$$\text{(AUX1}''') \quad \min_{w,\lambda}(\max_{w,\lambda})\,\lambda \quad \text{subject to } K_N w = \lambda K \Delta c, \quad w \geq -s_N^*, \tag{3.5}$$

which exactly yields the range of t for perturbations of the right-hand side of (D′) if one uses the form (D′) and (P′). Similarly, if (w, λ) is feasible for (AUX1′), then

$$\begin{bmatrix} \lambda \Delta c_B \\ \lambda \Delta c_N - w \end{bmatrix}$$

lies in the null space of K. Then, by our previous observation, there exists a v such that $B^T v = \lambda \Delta c_B$, $N^T v + w = \lambda \Delta c_N$, which is exactly the constraints of (AUX2), completing the proof of the claim.

Using this observation, we will carry out our analysis for perturbations of b only in the subsequent sections, and state the corresponding results for changes in c as corollaries. We begin with a special case of degeneracy and then consider the most general case.

4. Unique primal solution. Throughout this section, we assume that (P) has a unique but degenerate optimal solution x^*. Note that by Proposition 2.1, we have $|\mathcal{B}| = \text{rank } (B)$, i.e., B has linearly independent columns. In this particular case, Proposition 2.4 provides another useful bound on x_B for a strictly feasible primal-dual point (x, y, s).

COROLLARY 4.1. *Assume that* (P) *has a unique optimal solution* x^*. *Let* (x, y, s) *be primal-dual strictly feasible for* (P) *and* (D) *with duality gap* μn. *Then,*

$$x_B = x_B^* + O(\mu). \qquad \square \tag{4.1}$$

An analogous corollary follows if (D) has a unique optimal solution.

COROLLARY 4.2. *Assume that* (D) *has a unique optimal solution* (y^*, s^*). *Let* (x, y, s) *be primal-dual strictly feasible for* (P) *and* (D) *with duality gap* μn. *Then,*

$$s_N = s_N^* + O(\mu). \qquad \square \tag{4.2}$$

Next, we will analyze one-dimensional perturbations of b.

4.1. Perturbations of *b*. In this subsection, we assume that the right-hand side vector b is replaced by $b + t\Delta b$, where $\Delta b \in \mathbb{R}^m$ and $t \in \mathbb{R}$. We also assume that $(x, y, s) \in \mathcal{N}_{-\infty}(\gamma)$ is a primal-dual strictly feasible point for (P) and (D) for some $\gamma \in (0, 1]$. We will compare the interior-point bounds arising from Proposition 2.2 at (x, y, s) with the optimal values of (SA1), i.e., the symmetrized bounds. The interior-point bounds are given by the L_∞-norm of

$$S^{-1} A^T (A D^2 A^T)^{-1} \Delta b, \tag{4.3}$$

where $D^2 = XS^{-1}$.

Let us now consider (SA1). Since B has full column rank, Δb either does not lie in the range space of B, in which case the optimal values of (SA1) as well as (AUX1) are all 0, or there exists a unique $v \in \mathbb{R}^{|\mathcal{B}|}$ such that $Bv = \Delta b$, in which case the constraints of (SA1) reduce to (2.9), from which the symmetrized bounds can be obtained easily. We will consider both situations in turn.

Let us start with the second case. Without loss of generality, we can assume that Δb has unit L_2-norm, which implies a bound on v. Then, we need to compute

$$u = (AD^2A^T)^{-1}\Delta b = (AD^2A^T)^{-1}Bv \tag{4.4}$$

in order to obtain (4.3). However, (4.4) is equivalent to

$$AD^2A^Tu = Bv \quad \text{or} \quad BX_BS_B^{-1}B^Tu + NX_NS_N^{-1}N^Tu = Bv, \tag{4.5}$$

where B and N are the partitions of the coefficient matrix A with respect to $\mathcal{B}$ and $\mathcal{N}$, as before. Since B has linearly independent columns, there exists a matrix $C \in \mathbb{R}^{m\times(m-|\mathcal{B}|)}$ such that the augmented matrix $[B \quad C]$ is square and nonsingular: let W be its inverse. Therefore, premultiplying the second equality in (4.5) by W, we obtain

$$\begin{bmatrix} I \\ 0 \end{bmatrix} X_BS_B^{-1}\,[I \quad 0]\,\tilde{u} + \tilde{N}X_NS_N^{-1}\tilde{N}^T\tilde{u} = \begin{bmatrix} I \\ 0 \end{bmatrix} v, \tag{4.6}$$

where $\tilde{u} = W^{-T}u$, $\tilde{N} = WN$, and I is a $|\mathcal{B}| \times |\mathcal{B}|$ identity matrix. Therefore, if we partition $\tilde{N}$ and $\tilde{u}$ accordingly as

$$\tilde{N} = \begin{bmatrix} \tilde{N}_1 \\ \tilde{N}_2 \end{bmatrix}, \quad \tilde{u} = \begin{bmatrix} \tilde{u}_1 \\ \tilde{u}_2 \end{bmatrix},$$

then (4.6) can be decomposed in the following way:

$$\begin{bmatrix} D_B^2 + \tilde{N}_1D_N^2\tilde{N}_1^T & \tilde{N}_1D_N^2\tilde{N}_2^T \\ \tilde{N}_2D_N^2\tilde{N}_1^T & \tilde{N}_2D_N^2\tilde{N}_2^T \end{bmatrix} \begin{bmatrix} \tilde{u}_1 \\ \tilde{u}_2 \end{bmatrix} = \begin{bmatrix} v \\ 0 \end{bmatrix}, \tag{4.7}$$

where D_B and D_N are the corresponding partitions of D. By (4.3), we need to compute

$$S^{-1}A^Tu = S^{-1}(WA)^T\tilde{u} = \begin{bmatrix} S_B^{-1}\tilde{u}_1 \\ S_N^{-1}\left(\tilde{N}_1^T\tilde{u}_1 + \tilde{N}_2^T\tilde{u}_2\right) \end{bmatrix}. \tag{4.8}$$

For notational convenience, let us define

$$F := \tilde{N}_1D_N, \quad G := \tilde{N}_2D_N.$$

Note that G has full row rank since A does. The bottom equality in (4.7) can be rewritten as

$$GF^T\tilde{u}_1 + GG^T\tilde{u}_2 = 0, \quad \text{and so} \quad \tilde{u}_2 = -(GG^T)^{-1}GF^T\tilde{u}_1. \tag{4.9}$$

Substituting (4.9) into the top equality in (4.7) gives

$$\begin{aligned} \left(D_B^2 + FF^T - FG^T(GG^T)^{-1}GF^T\right)\tilde{u}_1 = v, \qquad &\text{or} \\ \left(D_B^2 + F(I - P_G)F^T\right)\tilde{u}_1 = v, \qquad & \end{aligned} \tag{4.10}$$

where P_G is the orthogonal projection matrix onto the range space of G^T. Therefore, $I - P_G$ is the orthogonal projection matrix onto the null space of G.

We now briefly review the Neumann lemma [3]. Let U be an invertible matrix and let V satisfy $\|U^{-1}V\| \leq 1/2$. (The particular norm being used does not really

matter: We will always use $\|\cdot\|$ for the Euclidean norm or the operator norm arising from it.) Then, $I + U^{-1}V$ is invertible with $\|I + U^{-1}V\| \leq 2$. Moreover $U + V$ is invertible and given by

$$(U+V)^{-1} = U^{-1} - U^{-1}V(I+U^{-1}V)^{-1}U^{-1}. \tag{4.11}$$

Now, we apply this result to (4.10) with $U := D_B^2$ and $V := F(I-P_G)F^T$. Proposition 2.6 implies that both U^{-1} and V are $O(\mu)$, since $I - P_G$ is a projection matrix and has unit Euclidean norm. Therefore, assuming that the duality gap μn is small,

$$\begin{aligned} \tilde{u}_1 &= \left(D_B^2 + F(I-P_G)F^T\right)^{-1} v, \\ &= D_B^{-2}v - D_B^{-2}F(I-P_G)F^T\left(I + D_B^{-2}F(I-P_G)F^T\right)^{-1} D_B^{-2}v. \end{aligned} \tag{4.12}$$

It then follows that

$$S_B^{-1}\tilde{u}_1 = X_B^{-1}v - X_B^{-1}F(I-P_G)F^T\left(I + D_B^{-2}F(I-P_G)F^T\right)^{-1} D_B^{-2}v. \tag{4.13}$$

However, by Proposition 2.6, F is $O(\mu^{1/2})$, D_B^{-2} is $O(\mu)$, and X_B^{-1} is $O(1)$. Consequently, the second term on the right-hand side of (4.13) is $O(\mu^2)$, since $\|I-P_G\| \leq 1$. Finally, Corollary 4.1 implies $X_B^{-1} = (X_B^*)^{-1} + O(\mu)$. Therefore,

$$S_B^{-1}\tilde{u}_1 = (X_B^*)^{-1}v + O(\mu). \tag{4.14}$$

We have thus obtained the top part of (4.8). For the lower part, we get

$$\begin{aligned} S_N^{-1}(\tilde{N}_1^T\tilde{u}_1 + \tilde{N}_2^T\tilde{u}_2) &= S_N^{-1}D_N^{-1}(F^T\tilde{u}_1 + G^T\tilde{u}_2), \\ &= (X_N S_N)^{-1/2}(I-P_G)F^T\tilde{u}_1, \end{aligned} \tag{4.15}$$

where we substituted (4.9) for $\tilde{u}_2$. Proposition 2.6 implies $(X_N S_N)^{-1/2} = O(\mu^{-1/2})$ and $F = O(\mu^{1/2})$. By (4.14), $\tilde{u}_1 = O(\mu)$ since s_B is $O(\mu)$. Combining these bounds with $\|I - P_G\| \leq 1$ leads to

$$S_N^{-1}(\tilde{N}_1^T\tilde{u}_1 + \tilde{N}_2^T\tilde{u}_2) = O(\mu). \tag{4.16}$$

Using (4.8), we conclude that the L_∞-norm of the quantity (4.3) that we need to evaluate is given by

$$\left\|S^{-1}A^T(AD^2A^T)^{-1}\Delta b\right\|_\infty = \left\| \begin{bmatrix} (X_B^*)^{-1}v + O(\mu) \\ O(\mu) \end{bmatrix} \right\|_\infty. \tag{4.17}$$

The reciprocal of (4.17) gives the desired interior-point bound. Consequently, if the duality gap μn is small, we conclude by comparing (4.17) with (2.9) that the interior-point approach yields exactly the same bound as the optimal solution to (SA1) asymptotically in μ.

Next, we address the situation in which Δb does not lie in the range space of B. In this case, the optimal values of both (AUX1) and (SA1) are clearly 0. Δb can be uniquely written as

$$\Delta b = Bv_B + Cv_C, \tag{4.18}$$

where $[B \quad C]$ is nonsingular as before and v_C is a nonzero vector. Once again, we need to compute (4.3). We follow a similar treatment as before, and corresponding to (4.7) we obtain

$$\begin{bmatrix} D_B^2 + \tilde{N}_1 D_N^2 \tilde{N}_1^T & \tilde{N}_1 D_N^2 \tilde{N}_2^T \\ \tilde{N}_2 D_N^2 \tilde{N}_1^T & \tilde{N}_2 D_N^2 \tilde{N}_2^T \end{bmatrix} \begin{bmatrix} \tilde{u}_1 \\ \tilde{u}_2 \end{bmatrix} = \begin{bmatrix} v_B \\ v_C \end{bmatrix}. \tag{4.19}$$

The bottom part can be expanded as

$$\tilde{N}_2 X_N \left[S_N^{-1} \tilde{N}_1^T \tilde{u}_1 + S_N^{-1} \tilde{N}_2^T \tilde{u}_2 \right] = v_C. \tag{4.20}$$

However, (4.8) implies that the term in the brackets is exactly the bottom part of the quantity (4.3) we seek. Let us denote that term by p and let $X_N p = q$. Then, (4.20) is equivalent to $\tilde{N}_2 q = v_C$. Since v_C is nonzero, the norm of q is bounded below; that is, $\|q\| \geq \alpha > 0$, where α is the norm of the least squares solution. Therefore, $\|q\|_\infty \geq \beta$ with $\beta := (\alpha/\sqrt{n - |\mathcal{B}|})$ (see, e.g., [3]). (Note that $|\mathcal{B}| < n$ since $|\mathcal{B}| = n$ can happen only if $m = n$, in which case Δb is always in the range of B.) However, $\|q\|_\infty \leq \|X_N\|_\infty \|p\|_\infty$ since $X_N p = q$. This implies

$$\|p\|_\infty \geq \frac{\|q\|_\infty}{\|X_N\|_\infty} \geq \frac{\beta}{\|X_N\|_\infty} = \Omega(1/\mu), \tag{4.21}$$

where the last equality follows from Corollary 2.5. Therefore, as μ tends to 0, $\|p\|_\infty$ tends to ∞, which implies that the interior-point bound given by its reciprocal tends to 0 as desired.

We remark that if $\mathcal{B} = \emptyset$, then $x^* = 0$ is the only optimal solution of (P), which can happen only if $b = 0$. In this case, the top part of (4.8) disappears. The interior-point bound is then given by the reciprocal of $\|p\|_\infty$, where p is as defined after (4.20). By the preceding argument, the interior-point bound tends to 0 as μ approaches 0. This is still in agreement with the optimal partition bounds since any nonzero perturbation of b leads to a change in the optimal partition, and hence, the optimal partition bounds in this case are also equal to 0. Therefore, we have proved the following theorem.

THEOREM 4.3. *Let $(x, y, s) \in \mathcal{N}_{-\infty}(\gamma)$ be a primal-dual strictly feasible point for* (P) *and* (D). *Assume that* (P) *has a unique but degenerate optimal solution, and that b is replaced by $b + t\Delta b$, where $t \in \mathbb{R}$ and $\Delta b \in \mathbb{R}^m$. Then the interior-point bound evaluated at (x, y, s) yields exactly the same value as the optimal solution of* (SA1) *asymptotically in μ, where $\mu = x^T s/n$.* □

The following corollary of Theorem 4.3 is an immediate consequence of the equivalence between (P) and (D), as discussed in section 3. One uses Corollary 4.2 in place of Corollary 4.1 in the preceding analysis.

COROLLARY 4.4. *Let $(x, y, s) \in \mathcal{N}_{-\infty}(\gamma)$ be a primal-dual strictly feasible point for* (P) *and* (D). *Assume that* (D) *has a unique but degenerate optimal solution and that c is replaced by $c + t\Delta c$, where $t \in \mathbb{R}$ and $\Delta c \in \mathbb{R}^n$. Then the interior-point bound evaluated at (x, y, s) yields exactly the same value as the optimal solution of* (SA2) *asymptotically in μ, where $\mu = x^T s/n$.* □

It does not appear that we can obtain better results for perturbations of c in the case of a unique primal optimal solution (but not dual optimal solution) than those arising from the analysis of the general case in the next section. A similar remark holds for perturbations of b in the case of a unique dual optimal solution (but not primal optimal solution).

5. General case. In this section, we turn our attention to the most general case, in which both (P) and (D) may have multiple optimal solutions. As the small example given at the end of section 2.1 reveals, some complications arise in the presence of multiple optimal solutions. For instance, unlike the previous case, the symmetrized bounds become dependent on the optimal solution of (P) used in the formulation of (SA1) if (P) has multiple optimal solutions. Furthermore, they do not necessarily

coincide with the "symmetrizations" of the optimal partition bounds arising from (AUX1). Similar remarks hold for the relationship between (SA2) and (AUX2) if (D) has multiple optimal solutions.

Despite this complication arising from the presence of multiple optimal solutions, we will attempt to say something about the quality of the interior-point bounds, at least in comparison with the symmetrized bounds. In the next subsection, we analyze perturbations of b in this general setting.

5.1. Perturbations of *b*. Let (P) have multiple optimal solutions and let b be replaced by $b + t\Delta b$, where $t \in \mathbb{R}$ and $\Delta b \in \mathbb{R}^m$. Suppose that $(x, y, s) \in \mathcal{N}_{-\infty}(\gamma)$ is primal-dual strictly feasible, where $\gamma \in (0, 1]$. For such a point, Proposition 2.4 guarantees the existence of a strictly complementary solution (x^*, y^*, s^*) whose distance from (x, y, s) is $O(\mu)$. We will compare the interior-point bounds evaluated at (x, y, s) with the optimal values of (SA1). Among other optimal solutions of (P), the x^* above will be the particular choice of the primal optimal solution to be used in the formulation of (SA1). The use of such an optimal solution in the relative interior of the primal optimal set is likely to leave more room for the decision variables of (SA1), since $x^*_B > 0$.

Let us first consider (SA1). The constraints of (SA1) are

$$\begin{array}{rcccl} Bu & = & \lambda\Delta b, \\ -x^*_B & \leq & u & \leq & x^*_B. \end{array} \tag{5.1}$$

Let rank $(B) = r$ and $|\mathcal{B}| = k$. Clearly we have $r \leq m$ and $r < k$, since Proposition 2.1 implies dim $(\Omega_P) = k - r$, which is positive by our assumption. This, in turn, implies that $r > 0$, since $r = 0$, if and only if $\mathcal{B} = \emptyset$ (assuming that no columns of A are identically zero). A QR factorization of B yields $B = QR$, where $Q \in \mathbb{R}^{m\times m}$ is orthogonal and $R \in \mathbb{R}^{m\times k}$ is upper triangular with

$$R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix}, \tag{5.2}$$

where R_1 has r rows. Note that R_1 has full row rank.

Premultiplying the equality constraints in (5.1) by Q^T yields

$$\begin{array}{rcccl} \begin{bmatrix} R_1 \\ 0 \end{bmatrix} u & = & \lambda \begin{bmatrix} \widetilde{\Delta b}_1 \\ \widetilde{\Delta b}_2 \end{bmatrix}, \\ -x^*_B & \leq & u & \leq & x^*_B, \end{array} \tag{5.3}$$

with $\widetilde{\Delta b} = Q^T\Delta b$. Clearly, (5.3) reveals that (SA1) has a nontrivial optimal solution λ^* if and only if $\widetilde{\Delta b}_2 = 0$.

First, we consider the nontrivial case. (Since $\widetilde{\Delta b}$ is nonzero, this implies that $k > 0$.) Let (λ^*, u^*) be an optimal solution to the maximization problem with $\lambda^* \neq 0$. Note that λ^* is finite since u^* is bounded (this follows, since $\mathcal{B} \neq \emptyset$). Then, we have

$$\widetilde{\Delta b} = Q^T\Delta b = (1/\lambda^*)Ru^*. \tag{5.4}$$

The interior-point approach, on the other hand, requires the evaluation of (4.3) at (x, y, s). By (5.4), we then need to evaluate the L_∞-norm of

$$(1/\lambda^*)S^{-1}A^T(AD^2A^T)^{-1}QRu^*. \tag{5.5}$$

Let

$$w = (AD^2A^T)^{-1}QRu^* \quad \text{or} \quad AD^2A^Tw = QRu^*. \tag{5.6}$$

Premultiplying the second equality in (5.6) by Q^T gives

$$\begin{bmatrix} R_1 & \tilde{N}_1 \\ 0 & \tilde{N}_2 \end{bmatrix} \begin{bmatrix} D_B^2 & 0 \\ 0 & D_N^2 \end{bmatrix} \begin{bmatrix} R_1^T & 0 \\ \tilde{N}_1^T & \tilde{N}_2^T \end{bmatrix} \begin{bmatrix} \tilde{w}_1 \\ \tilde{w}_2 \end{bmatrix} = \begin{bmatrix} R_1 \\ 0 \end{bmatrix} u^*, \tag{5.7}$$

where $\tilde{w}_1$ and $\tilde{w}_2$ are the appropriate partitions of $\tilde{w} = Q^Tw$, and $\tilde{N}_1$ and $\tilde{N}_2$ are those of $\tilde{N} = Q^TN$. Let us define

$$F := \tilde{N}_1D_N, \quad G := R_1D_B, \quad H := \tilde{N}_2D_N. \tag{5.8}$$

We can then decompose (5.7) into two equations as

$$\begin{array}{rcl} (GG^T + FF^T)\tilde{w}_1 + FH^T\tilde{w}_2 &=& R_1u^*, \\ HF^T\tilde{w}_1 + HH^T\tilde{w}_2 &=& 0. \end{array} \tag{5.9}$$

Note, in particular, that both G and H have full row rank since R_1 and A do. From the second equation in (5.9), we obtain

$$\tilde{w}_2 = -(HH^T)^{-1}HF^T\tilde{w}_1. \tag{5.10}$$

Substituting (5.10) into the first equation of (5.9) leads to

$$\begin{aligned} (GG^T + FF^T - FH^T(HH^T)^{-1}HF^T)\tilde{w}_1 &= R_1u^*, \quad \text{or} \\ (GG^T + F(I - P_H)F^T)\tilde{w}_1 &= R_1u^*, \end{aligned} \tag{5.11}$$

where $I - P_H$ is the orthogonal projection matrix onto the null space of H. Proposition 2.6 implies that the second term in parentheses in the second equation above is $O(\mu)$, since $\|I - P_H\| \leq 1$. In order to apply Neumann's lemma, we need to show that $(GG^T)^{-1}$ is bounded.

LEMMA 5.1. $(GG^T)^{-1} = O(\mu)$.

Proof. We use the "thin" QR factorization of $G^T = D_BR_1^T = YZ$, where Y has orthonormal columns and Z is upper triangular and nonsingular. Then, $(GG^T)^{-1} = Z^{-1}Z^{-T}$. Therefore, it suffices to find an upper bound on Z^{-1}. We have

$$D_BR_1^T = YZ, \quad \text{or} \quad R_1R_1^T = R_1D_B^{-1}YZ. \tag{5.12}$$

Therefore, $I = (R_1R_1^T)^{-1}R_1D_B^{-1}YZ$, or $Z^{-1} = (R_1R_1^T)^{-1}R_1D_B^{-1}Y$. However, by Proposition 2.6, $D_B^{-1} = O(\mu^{1/2})$, which implies that $Z^{-1} = O(\mu^{1/2})$, completing the proof. ∎

We can now apply Neumann's lemma to (5.11). Using the same notation as in (4.11), we have $U := GG^T$ and $V := F(I - P_H)F^T$. Note that both U^{-1} and V are $O(\mu)$. We obtain

$$\tilde{w}_1 = (GG^T)^{-1}GD_B^{-1}u^* - (GG^T)^{-1}V(I + (GG^T)^{-1}V)^{-1}(GG^T)^{-1}GD_B^{-1}u^*, \tag{5.13}$$

where we used $R_1 = GD_B^{-1}$.

By (5.5) and (5.6), we need

$$(1/\lambda^*)S^{-1}A^Tw = (1/\lambda^*)\begin{bmatrix} S_B^{-1}R_1^T\tilde{w}_1 \\ S_N^{-1}(\tilde{N}_1^T\tilde{w}_1 + \tilde{N}_2^T\tilde{w}_2) \end{bmatrix}. \tag{5.14}$$

Let us define

$$W = G^T(GG^T)^{-1}. \tag{5.15}$$

For the top part of (5.14) we need to evaluate

$$\begin{aligned} S_B^{-1}R_1^T\tilde{w}_1 &= S_B^{-1}D_B^{-1}G^T\tilde{w}_1, \\ &= (S_BX_B)^{-1/2}P_GD_B^{-1}u^* \\ &\quad - (S_BX_B)^{-1/2}WV(I+(GG^T)^{-1}V)^{-1}W^TD_B^{-1}u^*, \end{aligned} \tag{5.16}$$

where we used (5.13), (5.15) and where P_G is the orthogonal projection matrix onto the range space of G^T. Consider the second term in the right-hand side of the second equality. By Proposition 2.6, $(S_BX_B)^{-1/2}$ is $O(\mu^{-1/2})$, V is $O(\mu)$, and D_B^{-1} is $O(\mu^{1/2})$. Lemma 5.1 implies that $W = G^T(GG^T)^{-1} = YZ^{-T} = O(\mu^{1/2})$, since $\|Y\| \leq 1$. Therefore, the whole expression is $O(\mu^2)$. We conclude that the top part of (5.14) is

$$(1/\lambda^*)(S_BX_B)^{-1/2}P_G(S_BX_B)^{1/2}X_B^{-1}u^* + (1/\lambda^*)O(\mu^2). \tag{5.17}$$

Let us next consider the lower part of (5.14). We need to compute

$$S_N^{-1}\tilde{N}_1^T\tilde{w}_1 + S_N^{-1}\tilde{N}_2^T\tilde{w}_2. \tag{5.18}$$

By (5.13) the first term in (5.18) is given by

$$(S_NX_N)^{-1/2}F^T\left[W^T - (GG^T)^{-1}V(I+(GG^T)^{-1}V)^{-1}W^T\right]D_B^{-1}u^*. \tag{5.19}$$

Note that $W = O(\mu^{1/2})$ by the preceding discussion. As for the second term in brackets, we have that both $(GG^T)^{-1}$ and V are $O(\mu)$, which implies that the whole second term is $O(\mu^{5/2})$. Thus, the expression in brackets is $O(\mu^{1/2})$. By Proposition 2.6, $(S_NX_N)^{-1/2}$ is $O(\mu^{-1/2})$, whereas both F^T and D_B^{-1} are $O(\mu^{1/2})$. We therefore conclude that (5.19) is $O(\mu)$.

For the second term in (5.18), we use (5.10) together with (5.13):

$$-(S_NX_N)^{-1/2}H^T(HH^T)^{-1}HF^T\tilde{w}_1 = -(S_NX_N)^{-1/2}P_HF^T\tilde{w}_1. \tag{5.20}$$

Note that $\tilde{w}_1 = O(\mu)$ by the preceding arguments. The fact that $\|P_H\| \leq 1$, together with $(S_NX_N)^{-1/2}$ being $O(\mu^{-1/2})$ and F^T being $O(\mu^{1/2})$, implies that (5.20) is $O(\mu)$.

Therefore, we conclude that the lower part of (5.14) is $O(\mu)$. Combining this result with (5.17) yields the following:

$$r := (1/\lambda^*)\begin{bmatrix} (S_BX_B)^{-1/2}P_G(S_BX_B)^{1/2}X_B^{-1}u^* + O(\mu^2) \\ O(\mu) \end{bmatrix}. \tag{5.21}$$

Consequently, we need to evaluate the L_∞-norm of (5.21) and take its reciprocal. Observe that $X_B^{-1} = (X_B^*)^{-1} + O(\mu)$ by Proposition 2.4. Using this, we derive an upper bound on the L_∞-norm of (5.21).

$$\|r\|_\infty \leq |1/\lambda^*|(\|(S_BX_B)^{-1/2}\|_\infty\|P_G\|_\infty\|(S_BX_B)^{1/2}\|_\infty\|(X_B^*)^{-1}u^*\|_\infty + O(\mu)). \tag{5.22}$$

Since $(x,y,s) \in \mathcal{N}_{-\infty}(\gamma)$,

$$x_is_i = \mu n - \sum_{j\neq i} x_js_j \leq \mu n - \mu(n-1)\gamma = \mu(n-(n-1)\gamma). \tag{5.23}$$

Thus, $(x_i s_i)^{1/2} \le (\mu(n-(n-1)\gamma)^{1/2}$ and $(x_i s_i)^{-1/2} \le 1/(\gamma\mu)^{1/2}$. Furthermore, since $\|P_G\| \le 1$, we have $\|P_G\|_\infty \le k^{1/2}$ (see, e.g., [3]), where $|\mathcal{B}| = k$. Finally, since u^* is optimal for (SA1), $\|(X_B^*)^{-1}u^*\|_\infty \le 1$. Therefore,

$$\begin{aligned}\|r\|_\infty &\le \frac{1}{|\lambda^*|}\left(\frac{1}{(\gamma\mu)^{1/2}}k^{1/2}(\mu(n-(n-1)\gamma))^{1/2} + O(\mu)\right),\\ &= \frac{1}{|\lambda^*|}\left(\sqrt{\frac{(n-(n-1)\gamma)k}{\gamma}} + O(\mu)\right). \qquad (5.24)\end{aligned}$$

We conclude that the interior-point bound, which is the reciprocal of (5.24), is then bounded below by

$$\frac{1}{\|r\|_\infty} \ge \frac{\sqrt{\gamma}}{\sqrt{(n-(n-1)\gamma)k} + O(\mu)}|\lambda^*|. \qquad (5.25)$$

Note, in particular, that the lower bound tends to $1/\sqrt{k}$, independent of n, as $\mu \to 0$ if (x, y, s) is on the central path.

We next consider the case in which Δb is not in the range space of B. Again, in this case, the symmetrized bounds as well as the optimal partition bounds are all 0. The QR factorization of B can be rewritten as $B = QR = [Q_1 \quad Q_2]R = Q_1R_1$, where we use (5.2), and where $[Q_1 \quad Q_2]$ is the appropriate partition of Q. Since Q is orthogonal, Δb can uniquely be expressed as

$$\Delta b = Q_1 v_1 + Q_2 v_2, \qquad (5.26)$$

where v_2 is nonzero. Arguing as in section 4, we need to evaluate (4.3), which in turn requires the computation of

$$w = (AD^2A^T)^{-1}(Q_1v_1 + Q_2v_2) \quad \text{or} \quad AD^2A^Tw = Q_1v_1 + Q_2v_2. \qquad (5.27)$$

Premultiplying (5.27) by Q^T leads to

$$\begin{bmatrix} R_1 & \tilde{N}_1 \\ 0 & \tilde{N}_2 \end{bmatrix}\begin{bmatrix} D_B^2 & 0 \\ 0 & D_N^2 \end{bmatrix}\begin{bmatrix} R_1^T & 0 \\ \tilde{N}_1^T & \tilde{N}_2^T \end{bmatrix}\begin{bmatrix} \tilde{w}_1 \\ \tilde{w}_2 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \qquad (5.28)$$

which looks like (4.19). Essentially the same arguments as in section 4 reveal that the interior-point bound tends to 0 as μ approaches 0.

Therefore, we have proved the following theorem.

THEOREM 5.2. *Let $(x, y, s) \in \mathcal{N}_{-\infty}(\gamma)$ be a primal-dual strictly feasible point for* (P) *and* (D) *with duality gap μn. Assume that* (P) *has multiple optimal solutions and that b is replaced by $b + t\Delta b$, where $t \in \mathbb{R}$ and $\Delta b \in \mathbb{R}^m$. Let $k = |\mathcal{B}|$. If the strictly feasible solution given by Proposition* 2.4 *is used for symmetrization in* (SA1)*, then the ratio of the interior-point bound evaluated at (x, y, s) to the value of the optimal solution of* (SA1) *is at least*

$$\frac{\sqrt{\gamma}}{\sqrt{(n-(n-1)\gamma)k} + O(\mu)}. \qquad (5.29)$$

□

Note that the presence of multiple primal optimal solutions implies $k > 0$; therefore, the expression (5.29) is well-defined. As in section 4, Theorem 5.2 leads to the

following corollary by the discussion in section 3. Due to the interchange of the roles of the basic and nonbasic variables in the reformulation given in section 3, k in the denominator of (5.29) is replaced by $(n-k)$. Under the assumption of multiple dual optimal solutions, Proposition 2.1 indicates that $m > r$, which implies $k < n$, since A has full row rank.

COROLLARY 5.3. *Let $(x, y, s) \in \mathcal{N}_{-\infty}(\gamma)$ be a primal-dual strictly feasible point for* (P) *and* (D) *with duality gap μn. Assume that* (D) *has multiple optimal solutions and that c is replaced by $c+t\Delta c$, where $t \in \mathbb{R}$ and $\Delta c \in \mathbb{R}^n$. Let $k = |\mathcal{B}|$. If the strictly feasible solution given by Proposition* 2.4 *is used for symmetrization in* (SA2)*, then the ratio of the interior-point bound evaluated at (x, y, s) to the value of the optimal solution of* (SA2) *is at least*

$$\frac{\sqrt{\gamma}}{\sqrt{(n-(n-1)\gamma)(n-k)} + O(\mu)}. \qquad (5.30)$$

□

6. An example. In the previous sections, we have provided a theoretical basis for the behavior of the interior-point bounds evaluated at the near-optimal solutions. We present an example in this section to shed some light on the performance of the interior-point bounds in practice.

The example we consider in this section is a degenerate transportation problem discussed by Roos, Terlaky, and Vial [10, pp. 398–402]. For this problem, they report strikingly different results on sensitivity analysis on the right-hand side and the cost parameters using different commercial solvers. We aim to test our interior-point approach on this instance.

The problem is very simple. There are three distribution centers with capacity 2, 6, and 5 units, respectively, which can supply three warehouses each with a demand of 3 units at a cost of 1 per unit. We aim to minimize the total cost while meeting all the demand.

This problem can be formulated as a linear program in standard form as follows:

$$\begin{array}{ll} \min & \sum_{i=1}^{3}\sum_{j=1}^{3} x_{ij} \\ \text{subject to} & x_{11}+x_{12}+x_{13}+s_1 = 2, \\ & x_{21}+x_{22}+x_{23}+s_2 = 6, \\ & x_{31}+x_{32}+x_{33}+s_3 = 5, \\ & x_{11}+x_{21}+x_{31}-d_1 = 3, \\ & x_{12}+x_{22}+x_{32}-d_2 = 3, \\ & x_{13}+x_{23}+x_{33}-d_3 = 3, \\ & x_{ij},\ s_i,\ d_j \geq 0, \qquad\qquad i,j = 1,2,3, \end{array}$$

where x_{ij} denotes the amount shipped from distribution center i to warehouse j, s_i is the excess supply at distribution center i, and d_j is the shortage of demand at warehouse j, $i, j = 1, 2, 3$.

It is easy to verify that any feasible solution with objective value 9 is optimal. Therefore, there exist optimal solutions with all the variables x_{ij} and s_i, $i, j = 1, 2, 3$, positive, whereas all d_j, $j = 1, 2, 3$, will always be zero in any optimal solution. By Proposition 2.1, the primal optimal set has dimension 6, whereas the dual optimal solution is unique.

We solved this LP using CPLEX's standard barrier solver (http://www.ilog.com /products/cplex/product/barrier.cfm). By setting the tolerance level to different values (1e-3, 1e-4, 1e-5) we obtained several strictly feasible, near-optimal solutions with different duality gaps. We then evaluated the interior-point bounds (2.15) and (2.16)

TABLE 1
Interior-point bounds for the transportation example (b).

	μ	γ	b_1	b_2	b_3	b_4
1	3.94 e-4	0.98	1.653	2.554	2.542	2.496
2	2.62 e-5	0.98	1.639	2.572	2.519	2.480
3	1.86 e-6	0.74	1.487	2.641	2.332	2.163
CPLEX			$[-2, 1]$	$[-2, 1]$	$[-4, +\infty)$	-[1]
Range			$(-2, +\infty)$	$(-4, +\infty)$	$(-4, +\infty)$	$(-3, 4)$

TABLE 2
Interior-point bounds for the transportation example (c).

	μ	γ	c_{1j}	c_{2j}	c_{3j}
1	3.94 e-4	0.98	1.126 e-3	7.825 e-4	6.994 e-4
2	2.62 e-5	0.98	7.505 e-5	5.238 e-5	4.685 e-5
3	1.86 e-6	0.74	5.425 e-6	3.740 e-6	3.437 e-6
CPLEX			-[2]	-[2]	-[2]
Range			$\{0\}$	$\{0\}$	$\{0\}$

at these iterates for perturbations of the form $b+te_k$ and $c+te^l$, where e_k and e^l, with $k = 1, \ldots, 6$ and $l = 1, \ldots, 9$, denote the unit vectors in the appropriate dimension.

The results are reported in Table 1 and Table 2 for perturbations of the right-hand side parameters and the cost parameters, respectively. Rows 1–3 correspond to the three strictly feasible, near-optimal iterates ordered by descending duality gaps. For the iterates (x, y, s), μ denotes the duality measure given by $x^T s/n$, and γ is the parameter of the narrowest wide central-path neighborhood containing the iterate. The columns b_i in Table 1 correspond to changes in the right-hand side of the ith constraint, and the number in each column is the upper interior-point bound evaluated at the corresponding iterate. Since the changes in b_4–b_6 are symmetric, we report only the results for b_4. Similarly, the columns c_{ij} in Table 2 represent changes in the cost coefficient of the variable x_{ij}, and the number in that column is the upper interior-point bound evaluated at the corresponding iterate. Again, the changes in c_{ij} for fixed i are symmetric. The CPLEX row denotes the range obtained from that solver. (The basic variables returned by CPLEX were x_{11}, x_{21}, x_{22}, x_{23}, x_{33}, and s_3.) Finally, the last row in each table gives the correct ranges (optimal partition bounds) for the corresponding right-hand side or the cost parameters.

Table 1 reveals that the interior-point bounds provide useful information in comparison with the symmetrized optimal partition bounds as μ tends to 0, even if the LP is highly degenerate. Furthermore, the ratio of the interior-point bounds to the symmetrized optimal partition bounds is much better than the theoretical worst-case ratio (5.29). In fact, we experienced similar behavior in our extensive computational tests with randomly generated problems [12].

The results of Table 2 indicate the rapid convergence of the interior-point bounds to 0 as μ tends to 0, as expected. Note that the convergence rate is related to the duality measure μ, as illustrated by the previous theoretical results.

The particular instance we considered has multiple primal optimal solutions and a unique degenerate dual optimal solution. In order to get a complementary view, we

[1] CPLEX provided the following different results: $b_4 : [-1, 2]$, $b_5 : [-1, 2]$, $b_6 : [-1, 4]$.

[2] CPLEX ranges: $c_{11} : (-\infty, 0]$, $c_{12} : [0, +\infty)$, $c_{13} : [0, +\infty)$, $c_{21} : \{0\}$, $c_{22} : [-1, 0]$, $c_{23} : \{0\}$, $c_{31} : [0, +\infty)$, $c_{32} : [0, +\infty)$, $c_{33} : \{0\}$.

TABLE 3
Interior-point bounds for the modified transportation example (b).

	μ	γ	b_1	b_2	b_3	b_4	b_5	b_6
1	2.42 e-4	0.99	1.999	1.513 e-3	2.000	2.000	1.513 e-3	1.513 e-3
2	1.58 e-5	0.95	2.000	1.012 e-4	2.000	2.000	1.012 e-4	1.012 e-4
3	1.71 e-6	0.58	2.000	8.075 e-6	2.000	2.000	8.075 e-6	8.075 e-6
CPLEX			$[-2, +\infty)$	$[0, +\infty)$	$[-2, +\infty)$	$[-3, 2]$	$[-3, 0]$	$[-3, 0]$
Range			$(-2, +\infty)$	$\{0\}$	$(-2, +\infty)$	$(-3, 2)$	$\{0\}$	$\{0\}$

TABLE 4
Interior-point bounds for the modified transportation example (c).

	μ	γ	c_{11}	c_{12}	c_{13}	c_{21}	c_{22}	c_{23}	c_{31}	c_{32}	c_{33}
1	2.42 e-4	0.99	1.000	0.777	0.777	1.335	0.900	0.900	1.000	0.777	0.777
2	1.58 e-5	0.95	1.000	0.777	0.777	1.334	0.900	0.900	1.000	0.777	0.777
3	1.71 e-6	0.58	1.000	0.785	0.785	1.335	0.922	0.922	1.000	0.785	0.785
CPLEX			$[-1, +\infty)$	$[-1, +\infty)$	$[-1, +\infty)$	$[-1, +\infty)$	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$	$[-1, +\infty)$	$[-1, +\infty)$
Range			$(-1, +\infty)$	$(-1, +\infty)$	$(-1, +\infty)$	$(-2, +\infty)$	$(-2, 1)$	$(-2, 1)$	$(-1, 1)$	$(-1, +\infty)$	$(-1, +\infty)$

slightly modified the data of the transportation problem so that the primal problem has a unique degenerate optimal solution whereas the dual problem has multiple optimal solutions. More specifically, we increased the cost coefficients of x_{11}, x_{12}, x_{13}, x_{21}, x_{32}, and x_{33} from 1 to 2 in the objective function and maintained the remaining data. The resulting primal instance has the unique optimal solution given by $x_{22} = 3$, $x_{23} = 3$, $x_{31} = 3$, $s_1 = 2$, and $s_3 = 2$ with all other variables equal to 0.

We tested our interior-point approach on this problem instance. The results are tabulated in Tables 3 and 4 for the right-hand side and the cost parameters, respectively. (The basic variables returned by CPLEX in this example were x_{22}, x_{23}, x_{31}, s_1, s_2, and s_3.)

For perturbations of the right-hand side, Table 3 illustrates the convergence behavior predicted by the theoretical results. For perturbations of the cost vector, Table 4 reveals that the interior-point bounds provide very useful information in comparison with the symmetrized partition bounds at a moderate cost.

7. Conclusion. In this paper, we have studied the quality of the bounds arising from the interior-point perspective on sensitivity analysis developed by the authors in [13]. By relaxing the strong assumption of nondegeneracy, we have been able to consider all possible degeneracy scenarios and to investigate how our bounds compare with those arising from the optimal partition approach to sensitivity analysis.

If the primal problem has a degenerate but unique optimal solution, then our approach yields the same bounds as the "symmetrized" optimal partition bounds for perturbations of b. By the equivalence discussed in section 3, the same relationship holds for perturbations of c if the dual problem has a degenerate but unique optimal solution. This result directly extends the previous result proved in [13] under the assumption of a unique and nondegenerate solution.

We then considered general degenerate LPs. In this case, we were able to show that our interior-point approach would yield bounds that are at least a certain fraction of the symmetrized bounds, where the fraction depends on certain characteristics of the problem instance and of the iterate at which the bounds are evaluated. The behavior of the interior-point bounds on a highly degenerate transportation example indicates that useful information can be gained using the interior-point approach. Our extensive computational tests on random problems suggest that the ratio in practice is much better than the predicted worst-case ratio. Although this result is not as strong as the aforementioned results, our interior-point bounds still yield some useful information on the range of allowable perturbations. The fact that the cost of the evaluation of our bounds is simply the same as that of an interior-point iteration makes it more appealing, given the cost of solving two LPs, to obtain the range from the optimal partition approach.

Acknowledgments. We would like to thank two anonymous referees whose helpful comments and suggestions led to an improved exposition.

REFERENCES

[1] I. Adler and R. D. C. Monteiro, *A geometric view of parametric linear programming*, Algorithmica, 8 (1992), pp. 161–176.

[2] A. J. Goldman and A. W. Tucker, *Theory of linear programming*, in Linear Equalities and Related Systems, H. W. Kuhn and A. W. Tucker, eds., Princeton University Press, Princeton, NJ, 1956, pp. 53–97.

[3] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, 1996.

[4] H. J. Greenberg, *Simultaneous primal-dual right-hand-side sensitivity analysis from a strictly complementary solution of a linear program*, SIAM J. Optim., 10 (2000), pp. 427–442.
[5] H. J. Greenberg, A. G. Holder, K. Roos, and T. Terlaky, *On the dimension of the set of rim perturbations for optimal partition invariance*, SIAM J. Optim., 9 (1998), pp. 207–216.
[6] A. J. Hoffman, *On approximate solutions of systems of linear inequalities*, J. Res. National Bureau of Standards, 49 (1952), pp. 263–265.
[7] B. Jansen, J. de Jong, C. Roos, and T. Terlaky, *Sensitivity analysis in linear programming: Just be careful!*, European J. Oper. Res., 101 (1997), pp. 15–28.
[8] W.-J. Kim, C.-K. Park, and S. Park, *An ϵ-sensitivity analysis in the primal–dual interior point method*, European J. Oper. Res., 116 (1999), pp. 629–639.
[9] R. D. C. Monteiro and S. Mehrotra, *A general parametric analysis approach and its implication to sensitivity analysis in interior point methods*, Math. Programming, 72 (1996), pp. 65–82.
[10] C. Roos, T. Terlaky, and J.-P. Vial, *Theory and Algorithms for Linear Optimization : An Interior Point Approach*, Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley, Chichester, UK, 1997.
[11] S. J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, 1997.
[12] E. A. Yildirim, *An Interior-Point Perspective on Sensitivity Analysis in Linear Programming and Semidefinite Programming*, PhD thesis, Cornell University, Ithaca, NY, 2001.
[13] E. A. Yildirim and M. J. Todd, *Sensitivity analysis in linear programming and semidefinite programming using interior-point methods*, Math. Program., 90 (2001), pp. 229–261.

SIAM J. OPTIM.
Vol. 12, No. 3, pp. 715–723

SELF-SCALED BARRIERS FOR IRREDUCIBLE SYMMETRIC CONES*

RAPHAEL A. HAUSER† AND YONGDO LIM‡

Abstract. Self-scaled barrier functions are fundamental objects in the theory of interior-point methods for linear optimization over symmetric cones, of which linear and semidefinite programming are special cases. In this article we classify the special class of self-scaled barriers which are defined on irreducible symmetric cones. Together with a decomposition theorem for general self-scaled barriers this concludes the algebraic classification theory of these functions.

Key words. semidefinite programming, self-scaled barrier functions, interior-point methods, symmetric cones, Euclidean Jordan algebras

AMS subject classifications. Primary, 90C25, 52A41, 90C60; Secondary, 90C05, 90C20

PII. S1052623400370953

1. Introduction.

Self-scaled barriers are a special class of self-concordant barrier functions [20] and were introduced by Nesterov and Todd [21] for the purpose of extending long-step primal-dual symmetric interior-point methods from linear and semidefinite programming to more general convex optimization problems [21, 22]; see section 2 below. The domain of definition of a self-scaled barrier F is a regular (open, convex, with nonempty interior and not containing any full lines) cone Ω in a real Euclidean space $(V, \langle\cdot|\cdot\rangle)$. By abuse of language one often refers to F as a self-scaled barrier for the topological closure $\bar{\Omega}$ of Ω. Not every proper open convex cone Ω allows a self-scaled barrier, and for those which do, $\bar{\Omega}$ is called a *self-scaled cone* in the terminology of Nesterov and Todd [21].

It later emerged that a number of tools developed by Nesterov and Todd had previously been studied in the theory of Euclidean Jordan algebras. On the one hand, Güler [8] observed that the family of interiors of self-scaled cones is identical to the set of *symmetric cones* for which there exists a complete classification theory; see, e.g., [29] or [7]. On the other hand, Faybusovich [2] showed that interior-point methods for self-scaled programming can be analyzed purely in terms of Euclidean Jordan algebra techniques. Subsequently, a number of authors continued to develop the Jordan algebra approach, which is now accepted as a natural framework for the analysis of semidefinite programming; see, e.g., [3, 4, 5, 6, 27, 28, 1, 19]. The classification theory of self-scaled barriers to which this paper contributes continues the trend of relating the work of Nesterov and Todd to the existing Jordan algebra literature. Indeed, as a consequence the axiomatic theory of self-scaled barriers can be replaced by a simpler theory that assumes functions of a specific form which are algebraically related to the characteristic functions of symmetric cones (see below).

*Received by the editors April 11, 2000; accepted for publication (in revised form) July 31, 2001; published electronically February 8, 2002.

http://www.siam.org/journals/siopt/12-3/37095.html

†Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Silver Street, Cambridge CB3 9EW, England (rah48@damtp.cam.ac.uk). This author's research was supported in part by the Norwegian Research Council through project 127582/410 "Synode II," by the Engineering and Physical Sciences Research Council of the UK under grant GR/M30975, and by NSERC of Canada grants of J. Borwein and P. Borwein.

‡Department of Mathematics, Kyungpook National University, Taegu 702–701, Korea (ylim@knu.ac.kr). This author's research was supported in part by the Basic Research Program of the Korea Science and Engineering Foundation through project 2000-1-10100-007-3.

Self-scaled barriers were studied by Nesterov–Todd [21, 22], Güler [8], Faybusovich [2], Güler–Tunçel [25], and Hauser [10], and more recently by Hauser [12], Lim [18], Schmieta [26], and Güler [9]. Though an axiomatic theory of these functions exists, the only known examples are trivially related to the characteristic function of the cone Ω,

$$\varphi_\Omega(x) := \int_{\Omega^\sharp} \mathrm{e}^{-\langle x|s\rangle}\, ds, \tag{1.1}$$

where $\Omega^\sharp := \{s \in V : \langle x|s\rangle > 0\ \forall x \in \Omega\}$ is the polar of Ω with respect to $\langle\cdot|\cdot\rangle$. This was first observed by Güler, who showed that the universal barrier U (see [20]) of a symmetric cone is self-scaled and is a homothetic transformation $U = c_1 \ln\varphi_\Omega + c_2$ of the characteristic function φ_Ω, where $c_1 \geq 1$ and c_2 are constants. More generally, every symmetric cone Ω has a decomposition, unique up to indexing, into a direct sum of irreducible symmetric cones

$$\Omega = \Omega_1 \oplus \cdots \oplus \Omega_m,$$

where the Ω_i lie in subspaces $V_i \subseteq V$ decomposing V into a direct sum $V = \oplus_{i=1}^m V_i$. The *irreducible summands* Ω_i can be classified into five different types; see [7] and references therein. All known self-scaled barrier functions for Ω are of the form

$$H = c_0 + \bigoplus_{i=1}^m c_i \ln\varphi_{\Omega_i}, \tag{1.2}$$

where $c_1, \ldots, c_m \geq 1$. We use the direct sum notation for H in (1.2) and elsewhere to indicate that each $x = \oplus_{i=1}^m x_i \in \oplus_{i=1}^m \Omega_i = \Omega$ is mapped to $H(x) = c_0 + \sum_{i=1}^m c_i \ln\varphi_{\Omega_i}(x_i)$. It is also well known that each function of the form (1.2) is a self-scaled barrier for Ω.

The natural question arises as to whether *all* self-scaled barrier functions are of the form (1.2). Early dents into this question were made by Güler and Tunçel when considering *invariant barriers*; see [25, p. 124] and related material. In a chapter of his thesis [10] and in a subsequent report [11], Hauser showed that any self-scaled barrier H over a symmetric cone Ω decomposes into a direct sum $H = \oplus_{i=1}^m H_i$ of self-scaled barriers H_i over the irreducible components Ω_i, and that any *isotropic* (rotationally invariant) self-scaled barrier H_i on Ω_i is of the form $c_1 \ln\varphi_{\Omega_i} + c_2$ with $c_1 \geq 1$. Hauser also observed that any self-scaled barrier H on Ω is invariant under a rich class of rotations of Ω, i.e., elements of $\mathrm{O}(\Omega)$; see (2.1) below, where these particular rotations are defined in terms of the Hessians of H, and also see Lemma 2.2.19 of [10]. Suspecting that in the case in which Ω is irreducible this family of rotations is rich enough to generate all of $\mathrm{O}(\Omega)$, Hauser [10, 11] conjectured that all self-scaled barriers on irreducible symmetric cones are isotropic. This conjecture, proving the correctness of which is the primary objective of this paper, shows that all self-scaled barriers are indeed of the form (1.2) and concludes their algebraic classification.

Following the path suggested by Lemma 2.2.19 of [10], Hauser [12] showed the correctness of the isotropy conjecture in the special case of the cone of positive semidefinite symmetric matrices. The key mechanism in the proof, Proposition 3.3 of [12], later turned out to be closely related to Koecher's Theorem 4.9(b) in [15]. Using this theorem, Lim [18], Schmieta [26], and Güler [9] all independently proved the isotropy conjecture in the general case. Our article is based on Lim's generalization [18] of Hauser's approach from [12]. The first completed manuscript presenting the full classification of self-scaled barriers was Schmieta's report [26]. Güler's work [9] and Hauser's manuscript [11] were later combined in a joint report [13].

2. Preliminary notions. In this section we summarize definitions and identities that will be needed for proving the results of the main section. Recall from the introduction that Ω denotes a regular cone in a real Euclidean space $(V, \langle\cdot|\cdot\rangle)$. The set of vector space automorphisms of V that leave Ω invariant is called the *symmetry group* of Ω. We denote it by

$$G(\Omega) := \big\{\theta \in \mathrm{GL}(V) : \theta\Omega = \Omega\big\}.$$

The inner product on V defines an adjoint θ^* for each endomorphism θ, and this defines the orthogonal group $\mathrm{O}(V) := \big\{\theta \in \mathrm{GL}(V) : \theta^* = \theta^{-1}\big\}$. The subgroup

$$\mathrm{O}(\Omega) := G(\Omega) \cap \mathrm{O}(V) \tag{2.1}$$

is called the *orthogonal group* of Ω.

A ν-self-concordant logarithmically homogeneous barrier function (see [20] or [23] for a definition) $H \in \mathcal{C}^3(\Omega, \mathbb{R})$ is *self-scaled* if the following conditions are satisfied:

(i) $H''(w)x \in \Omega^\sharp$ for all $x, w \in \Omega$, and

(ii) $H_\sharp\big(H''(w)x\big) = H(x) - 2H(w) - \nu$ for all $x, w \in \Omega$.

The function $H_\sharp : s \mapsto \max\big\{-\langle x|s\rangle - H(x) : x \in \Omega\big\}$ is a self-scaled barrier defined on $\Omega^\sharp$; see [21]. It is assumed in the definition of a self-concordant function (see [20]) that the Hessians $H''(x)$ are nonsingular for all $x \in \Omega$.

If $H \in \mathcal{C}^3(\Omega, \mathbb{R})$ is a self-scaled barrier and $x \in \Omega, s \in \Omega^\sharp$, then there exists a unique *scaling point* $w_H(x, s) \in \Omega$ such that $s = H''(w_H(x, s))x$; see [21]. Furthermore, Nesterov and Todd [21] showed that

$$H''(w) \in \mathrm{Iso}\big(\Omega, \Omega^\sharp\big) \quad \forall w \in \Omega, \quad \text{and} \tag{2.2}$$

$$H''(x) = H''\big(w_H(x, s)\big) \circ H''_\sharp(s) \circ H''\big(w_H(x, s)\big). \tag{2.3}$$

It is customary to change the inner product $\langle\cdot|\cdot\rangle$ to $\langle\cdot|\cdot\rangle_f := \langle H''(f)\cdot|\cdot\rangle$, where f is a fixed element in Ω. We will always assume that $\langle\cdot|\cdot\rangle$ is already of this kind, i.e., that there exists an element $f \in \Omega$ such that $H''(f) = \mathrm{id}_\Omega$ is the identity when the Hessian is computed with respect to this inner product. Under this assumption it is easy to show that $\Omega^\sharp = \Omega$ and $H_\sharp = H + c$ for some constant c. Hence, in this framework we can reformulate properties (2.2) and (2.3) as follows:

$$H''(w) \in G(\Omega), \tag{2.4}$$

$$H''(x) = H''\big(w_H(x, s)\big) \circ H''(s) \circ H''\big(w_H(x, s)\big). \tag{2.5}$$

It follows from these properties that Ω can allow a self-scaled barrier only if it is a *symmetric cone*; see the definition below. On the other hand, every symmetric cone allows a self-scaled barrier; see, e.g., [8, 2].

Equation (2.5) is a reformulation of an identity which is called the *fundamental formula* in Jordan algebra theory; see (2.8) below. This identity is one of the keys to proving the isotropy conjecture, as it allows one to express all the Hessians of H in terms of the Hessian $H''(e)$ at a single point $e \in \Omega$ and in terms of the Hessians of the standard logarithmic barrier function (cf. (1.1))

$$F(x) = \ln \varphi_\Omega(x), \tag{2.6}$$

which is self-scaled [8]. Rothaus [24] showed that for every cone automorphism $\theta \in G(\Omega)$ there exists a unique $\omega \in \mathrm{O}(\Omega)$ and a unique $w \in \mathrm{int}(\Omega)$ such that θ has the

polar decomposition $\theta = \omega \circ F''(w)$. Note that since $\mathrm{O}(\Omega) \subset \mathrm{O}(V)$ and $F''(w)$ is a self-adjoint positive definite automorphism of V, Rothaus's polar decomposition is identical to Cartan's polar decomposition; see, e.g., [14]. The nontrivial part of Rothaus's result is the fact that both factors lie in $G(\Omega)$. The uniqueness of Cartan's polar decomposition trivially implies the following proposition.

PROPOSITION 1. *The set of self-adjoint positive definite automorphisms of V that preserve Ω coincides with $\{F''(w) : w \in \Omega\}$.*

If the regular cone $\Omega \subset V$ is self-dual with respect to the inner product $\langle\cdot|\cdot\rangle$ and is homogeneous, i.e., if the orbit of any element $x \in \Omega$ under the action of the symmetry group $G(\Omega)$ is Ω (a property which is also called transitivity), then Ω is a *symmetric cone*. Symmetric cones have been intensively studied in the theory of Euclidean Jordan algebras.

A *Jordan algebra* V over the field $\mathbb{R}$ or $\mathbb{C}$ is a commutative algebra with a multiplicative identity element e and such that the Jordan algebra law

$$x^2(xy) = x(x^2y) \tag{2.7}$$

holds for all $x, y \in V$. By L let us denote the left translation $L(x)y = xy$, and by P the so-called *quadratic representation*

$$P(x) = 2L(x)^2 - L(x^2)$$

of elements $x \in V$. The Jordan algebra law (2.7) can be equivalently formulated as $(xy)x^2 = x(yx^2)$. This weak associativity condition is strong enough to ensure that the subalgebra generated by $\{e, x\}$ in V is associative. An element $x \in V$ is said to be *invertible* if there exists an element y in the subalgebra generated by x and e such that $xy = e$. It is known that an element $x \in V$ is invertible if and only if $P(x)$ is invertible. In this case $P(x)^{-1} = P(x^{-1})$ holds true. If x and y are invertible, then $P(x)y$ is invertible and $(P(x)y)^{-1} = P(x^{-1})y^{-1}$. Moreover, the so-called *fundamental formula*

$$P(P(x)y) = P(x)P(y)P(x) \tag{2.8}$$

holds for any elements $x, y \in V$; see Proposition II.3.2(iii) in [7].

A *Euclidean Jordan algebra* is a finite-dimensional real Jordan algebra V equipped with an *associative* inner product $\langle\cdot|\cdot\rangle$, i.e., an inner product that satisfies the law

$$\langle xy|z\rangle = \langle y|xz\rangle \qquad \forall x, y, z \in V.$$

Henceforth the Euclidean space V introduced in section 1 is always assumed to be a Euclidean Jordan algebra. The reader may bear in mind the example of the space $\mathrm{Sym}(n, \mathbb{R})$ of $n \times n$ real symmetric matrices, which becomes a Euclidean Jordan algebra when it is endowed with the Jordan product $(1/2)(XY + YX)$ and the inner product $\langle X|Y\rangle = \mathrm{tr}(XY)$.

A *Jordan frame* of V is a system of orthogonal primitive idempotents $\{c_1, \dots, c_r\}$, where r is the rank of V. The *spectral theorem for Euclidean Jordan algebras* (see, e.g., Theorem III.1.2 of [7]) states that for every element $x \in V$ there exist a Jordan frame $\{c_1, \dots, c_r\}$ and a set of real numbers $\{\lambda_1, \dots, \lambda_r\}$ (the *eigenvalues* of x) such that $x = \sum_{i=1}^{r} \lambda_i c_i$.

By virtue of the power associative property $x^p \cdot x^q = x^{p+q}$ (see, e.g., [7]), the *Jordan algebra exponential map*

$$\exp : V \to V,$$
$$x \mapsto \sum_{n=0}^{\infty} x^n/n!$$

is well defined. This map is bijective with image set $\Omega := \exp V$. Here Ω coincides with the interior of the set of square elements of V, and this is also the set of invertible squares of V. A fundamental theorem of Euclidean Jordan algebras asserts that

(i) Ω is a symmetric cone and

(ii) every symmetric cone in a real Euclidean space arises in this way.

Note that in the example $V = \mathrm{S}ym(n, \mathbb{R})$ the mapping exp is the standard matrix exponential, and the corresponding symmetric cone is the open convex cone of positive definite $n \times n$ matrices $\Omega = \mathrm{S}ym(n, \mathbb{R})^+$.

Symmetric cones have been completely classified. There are five types of irreducible symmetric cones: The cones $\mathrm{S}ym(n, \mathbb{R})^+$, the cones of positive definite Hermitian and Hermitian quaternion $n \times n$ matrices, the Lorentzian cones, and a 27-dimensional exceptional cone. General symmetric cones are direct products of irreducible symmetric cones. The connected component $\mathrm{Aut}(V)_\circ$ of the identity id_V in the Jordan algebra automorphism group $\mathrm{Aut}(V)$ is a subgroup of $\mathrm{O}(\Omega)$. Ω is irreducible if and only if V is simple, and in this case we have $\mathrm{Aut}(V)_\circ = \mathrm{O}(\Omega)$. For all of these statements, see [7] and the references therein. Throughout this section we will assume that V is a simple Euclidean Jordan algebra with the associative inner product $\langle x|y\rangle = \mathrm{tr}(xy)$ and that Ω is the symmetric cone associated with V.

The symmetric cone Ω carries a $G(\Omega)$-invariant Riemannian metric defined by

$$\gamma_x(u, v) = \langle P(x^{-1})u|v\rangle \quad \forall x \in \Omega,\ u, v \in V.$$

For this metric the Jordan inversion $x \to x^{-1}$ is an involutive isometry on Ω that fixes e. The curve $t \mapsto P(a^{1/2})(P(a^{-1/2})b)^t$ is the unique geodesic that joins a to b in Ω, and the Riemannian distance $\delta(a, b)$ is given by $\delta(a, b) = (\sum_{i=1}^n \log^2 \lambda_i)^{1/2}$, where the λ_i are the eigenvalues of $P(a^{-1/2})b$. The *geometric mean* $a\#b$ of two elements $a, b \in \Omega$ is defined by

$$a\#b = P(a^{1/2})(P(a^{-1/2})b)^{1/2}.$$

This is the unique midpoint (*geodesic middle*) of a and b for the Riemannian distance δ. The metric δ is a Bruhat–Tits metric, i.e., a complete metric satisfying the semiparallelogram law, with midpoint $a\#b$; see [16] for further details. For example, if $V = \mathrm{S}ym(n, \mathbb{R})$ and $\Omega = \mathrm{S}ym(n, \mathbb{R})^+$, then we have $A\#B = A^{1/2}(A^{-1/2}BA^{-1/2})^{1/2}A^{1/2}$.

The following basic properties of geometric means will be useful for our purpose.

PROPOSITION 2 (see [17]). *Let a and b be elements of Ω. Then*

(i) *the quadratic equation $P(x)a^{-1} = b$ has the unique solution $a\#b$ in Ω,*

(ii) *$a\#b = b\#a$ (the commutativity property),*

(iii) *$(a\#b)^{-1} = a^{-1}\#b^{-1}$ (the inversion property),*

(iv) *$P(a\#b) = P(a)\#P(b) = P(a^{1/2})(P(a^{-1/2})P(b)P(a^{-1/2}))^{1/2}P(a^{1/2})$, and*

(v) *$g(a\#b) = g(a)\#g(b)$ for all $g \in G(\Omega)$ (the transformation property).*

Let $F(x) = -\ln \det(x)$ be the standard logarithmic barrier function on the symmetric cone Ω, where det is the determinant function of the Jordan algebra V;

see [7]. Then one can see that $F'(x) = -x^{-1}$ and the Hessian of F is given by $F''(x) = P(x^{-1})$. Proposition 2 implies that the geometric mean $a\#b^{-1}$ is the scaling point of a and $b \in \Omega$ defined by F. Indeed,

$$F''(a\#b^{-1})a = P\big((a\#b^{-1})^{-1}\big)a = P(a^{-1}\#b)a = b,$$

that is, $w_F(a,b) = a\#b^{-1}$.

3. Self-scaled barriers for irreducible symmetric cones. This is the main section of the article. Theorem 5 below proves the isotropy conjecture raised in [10], the missing piece in the classification theory of self-scaled barriers that motivated our research.

LEMMA 3. *Let Ω be an irreducible symmetric cone and $\alpha : \Omega \to \Omega$ a function such that*

$$x^{-1}\#y = \alpha(x)^{-1}\#\alpha(y) \tag{3.1}$$

for all $x, y \in \Omega$. Then $\alpha = \lambda \cdot \mathrm{id}_\Omega$ for some positive real number λ.

Proof. Upon exchanging the roles of x and y, Proposition 2(i) implies that condition (3.1) is equivalent to

$$\alpha(x) = P(y^{-1}\#x)\alpha(y). \tag{3.2}$$

Setting $y = e$ and using $e^{-1}\#x = x^{1/2}$ in (3.2), we get

$$\alpha(x) = P(x^{1/2})\alpha(e) \tag{3.3}$$

for all $x \in \Omega$. Let us show that $k(\alpha(e)) = \alpha(e)$ for all $k \in \mathrm{Aut}(V)_\circ$. Applying (3.2) and (3.3) to both x and y, we get

$$P(x^{1/2})\alpha(e) = P(y^{-1}\#x)\alpha(y) = P(y^{-1}\#x)\big(P(y^{1/2})\alpha(e)\big),$$

and hence we obtain the identity $\alpha(e) = (P(x^{-1/2})P(y^{-1}\#x)P(y^{1/2}))\alpha(e)$ for all $x, y \in \Omega$. Set

$$K := \{P(x^{-1/2})P(y^{-1}\#x)P(y^{1/2}) : x, y \in \Omega\}.$$

It follows from the definition of the geometric mean and from the fundamental formula that

$$P(a\#b) = P(a^{1/2})(P(P(a^{-1/2})b))^{1/2}P(a^{1/2}).$$

Together with Proposition 2(ii) this implies

$$\begin{aligned} &P(x^{-1/2})P(y^{-1}\#x)P(y^{1/2}) \\ &\qquad = P(x^{-1/2})P(x\#y^{-1})P(y^{1/2}) \\ &\qquad = P(x^{-1/2})\Big(P(x^{1/2})P\big(P(x^{-1/2})y^{-1}\big)^{1/2}P(x^{1/2})\Big)P(y^{1/2}) \\ &\qquad = P\big(P(x^{1/2})y\big)^{-1/2}P(x^{1/2})P(y^{1/2}). \end{aligned}$$

Therefore, the set K can be written as $K = \big\{P\big(P(x)y^2\big)^{-1/2}P(x)P(y) | x, y \in \Omega\big\}$. By Koecher's Theorem 4.9(b) (see [15]), K generates $\mathrm{Aut}(V)_\circ$. This implies that the

point $\alpha(e)$ is fixed by all Jordan automorphisms $k \in \mathrm{Aut}(V)_\circ$. Finally, Corollary IV.2.7 of [7] (in which the assumption of irreducibility for Ω is essential) says that the group $\mathrm{Aut}(V)_\circ$ acts transitively on the set of primitive idempotents of V. The spectral theorem applied to $\alpha(e)$ therefore implies that $\alpha(e) = \lambda e$ for some positive real number λ. Together with (3.3) this implies that $\alpha(x) = P(x^{1/2})(\lambda e) = \lambda P(x^{1/2})(e) = \lambda x$ for all $x \in \Omega$. □

COROLLARY 4. *Let H be an arbitrary self-scaled barrier for the irreducible symmetric cone Ω. Then there exists a positive constant λ such that $H''(x) = \lambda \cdot F''(x)$ for all $x \in \Omega$.*

Proof. Since the Hessians $H''(x)$ are positive definite cone automorphisms, Proposition 1 implies that there exists a well-defined function $\Upsilon : \Omega \to \Omega$ such that

$$H''(x) = P(\Upsilon(x)^{-1}). \tag{3.4}$$

Since H is self-scaled, we have

$$\begin{aligned} P(\Upsilon(x)^{-1}) &\overset{(3.4),(2.5)}{=} H''(w_H) \circ H''(y) \circ H''(w_H) \\ &= P(\Upsilon(w_H)^{-1}) \circ P(\Upsilon(y)^{-1}) \circ P(\Upsilon(w_H)^{-1}) \\ &\overset{(2.8)}{=} P(P(\Upsilon(w_H)^{-1})\Upsilon(y)^{-1}) \end{aligned}$$

for all $x, y \in \Omega$, where $w_H = w_H(x, y)$ denotes the scaling point of x and y for the self-scaled barrier H. The quadratic representation P is injective on Ω; see Lemma 2.3 of [17]. Therefore, the identity above shows that

$$\Upsilon(x)^{-1} = P\big(\Upsilon(w_H)^{-1}\big)\Upsilon(y)^{-1}$$

for all $x, y \in \Omega$. By Proposition 2, we have

$$\Upsilon(w_H)^{-1} = \Upsilon(y) \# \Upsilon(x)^{-1} = \Upsilon(x)^{-1} \# \Upsilon(y) \tag{3.5}$$

for all $x, y \in \Omega$. Now, $y = H''(w_H)(x) = P(\Upsilon(w_H)^{-1})(x)$ by definition of w_H, and Proposition 2(i) shows that we have $\Upsilon(w_H)^{-1} = x^{-1} \# y$, which together with (3.5) implies

$$x^{-1} \# y = \Upsilon(x)^{-1} \# \Upsilon(y)$$

for all $x, y \in \Omega$. The proof is now completed by Lemma 3. □

THEOREM 5. *If H is a self-scaled barrier for Ω, then there exist constants $c_1 > 0$ and $c_0 \in \mathbb{R}$ such that*

$$H : x \to -c_1 \ln \det(x) + c_0 \qquad \forall x \in \Omega.$$

Proof. It follows from Corollary 4 and the fundamental theorem of differential and integral calculus that H is of the form $c_1 F + \varphi + c_0$, where $c_1 = \lambda > 0$, $c_0 \in \mathbb{R}$, and $\varphi \in V$ is a linear form on V, i.e., there exists an element $y \in V$ such that $\varphi : x \mapsto \mathrm{tr}(yx)$ for all $x \in V$. One of the conditions in the definition of a ν-self-concordant barrier B for a convex open domain D is that the length of the Newton step $B''(x)^{-1}[-B'(x)]$ at $x \in D$, measured in the Riemannian metric $\|.\|_x$ defined by $B''(x)$, be uniformly bounded by $\nu^{1/2}$ (see, e.g., [20, 21, 23]); i.e.,

$$\begin{aligned} \|B''(x)^{-1}[-B'(x)]\|_x^2 &:= \big\langle B''(x)\big[-B''(x)^{-1}[B'(x)]\big] \big| - B''(x)^{-1}B'(x)\big\rangle \\ &= \big\langle B'(x) \big| \big(B''(x)\big)^{-1}[B'(x)]\big\rangle \le \nu. \end{aligned}$$

In particular, in the case of H this means that

$$\begin{aligned}\nu &\geq \|H'(x)\|_x^2 = \|y - \lambda x^{-1}\|_x^2 \overset{C.S.}{\geq} \left(\|y\|_x - \|\lambda x^{-1}\|_x\right)^2 \\ &\overset{Cor.4}{=} \left(\mathrm{tr}\big(\lambda^{-1}(P(x)y)y\big)^{1/2} - \big(\lambda^{-1}(P(x)(\lambda x^{-1}))(\lambda x^{-1})\big)^{1/2}\right)^2 \\ &= \lambda^{-1}\Big(\mathrm{tr}((P(x)y)y)^{1/2} - \lambda r^{1/2}\Big)^2\end{aligned}$$

for all $x \in \Omega$. But clearly, this implies that $y = 0$. □

Acknowledgments. Both authors would like to thank Professor Leonid Faybusovich for suggesting this collaboration and for proposing various important improvements. Raphael Hauser also wishes to express his warmest thanks to Syvert Nørsett for inviting him to NTNU Trondheim, and to Peter and Jonathan Borwein for inviting him to SFU in Vancouver, where part of this research was done.

REFERENCES

[1] F. Alizadeh and S. Schmieta, *Optimization with Semidefinite, Quadratic and Linear Constraints*, RRR Report 23-97, Rutgers Center for Operations Research, Piscataway, NJ, 1997.

[2] L. Faybusovich, *Jordan Algebras, Symmetric Cones and Interior-Point Methods*, Technical report, Department of Mathematics, University of Notre Dame, Notre Dame, IN, 1995.

[3] L. Faybusovich, *Euclidean Jordan algebras and interior-point algorithms*, Positivity, 1 (1997), pp. 331–357.

[4] L. Faybusovich, *Euclidean Jordan Algebras and Generalized Affine-Scaling Vector Fields*, Technical report, Department of Mathematics, University of Notre Dame, Notre Dame, IN, 1998.

[5] L. Faybusovich, *A Jordan-Algebraic Approach to Potential-Reduction Algorithms*, Technical report, Department of Mathematics, University of Notre Dame, Notre Dame, IN, 1998.

[6] L. Faybusovich and R. Arana, *A Long-Step Primal-Dual Algorithm for the Symmetric Programming Problem*, Technical report, University of Notre Dame, Notre Dame, IN, 1999.

[7] J. Faraut and A. Korányi, *Analysis on Symmetric Cones*, Oxford Mathematical Monograph, Oxford University Press, New York, 1994.

[8] O. Güler, *Barrier functions in interior-point methods*, Math. Oper. Res., 21 (1996), pp. 860–885.

[9] O. Güler, *Interior-Point Methods on Symmetric and Homogeneous Cones*, Invited presentation, 17th International Symposium on Mathematical Programming, Mathematical Programming Society, Atlanta, GA, 2000.

[10] R. A. Hauser, *On Search Directions for Self-Scaled Conic Programming*, Ph.D. thesis, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, 2000.

[11] R. A. Hauser, *Self-Scaled Barrier Functions: Decomposition and Classification*, Numerical Analysis Report DAMTP 1999/NA13, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, England, 1999.

[12] R. A. Hauser, *Self-Scaled Barriers for Semidefinite Programming*, Numerical Analysis Report DAMTP 2000/NA02, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, England, 2000.

[13] R. A. Hauser and O. Güler, *Self-scaled barrier functions on symmetric cones and their classification*, Found. Comput. Math., to appear.

[14] J. Hilgert and K. H. Neeb, *Lie Gruppen und Lie Algebren*, Vieweg Verlag, Braunschweig/Wiesbaden, Germany, 1991.

[15] M. Koecher, *Jordan Algebras and Their Applications*, Lectures notes, University of Minnesota, Minneapolis, MN, 1962.

[16] J. D. Lawson and Y. Lim, *The geometric mean, matrices, metrics, and more*, Amer. Math. Monthly, to appear.

[17] Y. Lim, *Geometric means on symmetric cones*, Arch. Math., 75 (2000), pp. 39–45.

[18] Y. Lim, *private communication*, Kyugpook National University, Taegu, Korea.

[19] M. Muramatsu, *On Commutative Class of Search Directions for Linear Programming over Symmetric Cones*, Report CS-00-02, Department of Computer Science, The University of Electro-Communications, Tokyo, Japan, 2000.
[20] Yu. E. Nesterov and A. S. Nemirovskii, *Interior Point Polynomial Algorithms in Convex Programming*, SIAM, Philadelphia, 1994.
[21] Yu. E. Nesterov and M. J. Todd, *Self-scaled barriers and interior-point for convex programming*, Math. Oper. Res., 22 (1997), pp. 1–42.
[22] Yu. E. Nesterov and M. J. Todd, *Primal-dual interior-point methods for self-scaled cones*, SIAM J. Optim., 8 (1998), pp. 324–364.
[23] J. Renegar, *A Mathematical View of Interior-Point Methods in Convex Optimization*, SIAM, Philadelphia, 2001.
[24] O. S. Rothaus, *Domains of positivity*, Abh. Math. Sem. Univ. Hamburg, 24 (1960), pp. 189–235.
[25] L. Tunçel, *Convex Optimization: Barrier Functions and Interior-Point Methods*, Technical report B–336, Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Tokyo, Japan, 1998.
[26] S. H. Schmieta, *Complete Classification of Self-Scaled Barrier Functions*, Technical report, Department of IEOR, Columbia University, New York, 2000.
[27] J. Sturm, *Similarity and other spectral relations for symmetric cones*, Linear Algebra Appl., 90 (2001), pp. 205–227.
[28] J. Sturm, *Using* SeDuMi 1.02*, a* MATLAB *toolbox for optimization over symmetric cones*, Optim. Methods Softw., 11–12 (1999), pp. 625–653.
[29] E. B. Vinberg, *The theory of convex homogeneous cones*, Trudy Moskov. Mat. Obsc., 12 (1963), pp. 303–358; Trans. Moscow Math. Soc., 13 (1963), pp. 340–403.

SIAM J. OPTIM.
Vol. 12, No. 3, pp. 724–739

AN IMPLEMENTABLE ACTIVE-SET ALGORITHM FOR COMPUTING A B-STATIONARY POINT OF A MATHEMATICAL PROGRAM WITH LINEAR COMPLEMENTARITY CONSTRAINTS*

MASAO FUKUSHIMA† AND PAUL TSENG‡

Abstract. We consider a mathematical program with a smooth objective function and linear inequality/complementarity constraints. We propose an ϵ-active set algorithm which, under a uniform LICQ on the ϵ-feasible set, generates iterates whose cluster points are B-stationary points of the problem. If the objective function is quadratic and ϵ is set to zero, the algorithm terminates finitely. Some numerical experience with the algorithm is reported.

Key words. MPEC, B-stationary point, ϵ-active set

AMS subject classifications. 65K05, 90C30, 90C33

PII. S1052623499363232

1. Introduction. We consider the following mathematical program with equilibrium constraints (MPEC):

$$\begin{array}{lll} \text{minimize} & f(z) & \\ \text{subject to} & G_i(z) \geq 0, & i = 1, \ldots, m, \\ & H_i(z) \geq 0, & i = 1, \ldots, m, \\ & G_i(z)H_i(z) = 0, & i = 1, \ldots, m, \\ & g_j(z) \leq 0, & j = 1, \ldots, p, \\ & h_l(z) = 0, & l = 1, \ldots, q, \end{array} \tag{1}$$

where f is a real-valued continuously differentiable function on $\Re^n$ and G_i, H_i, g_j, h_l are real-valued *affine* functions on $\Re^n$.

This problem has been of much interest, and many algorithms aimed at global convergence have been proposed for its solution, as is evidenced by [1, 2, 5, 7, 10] and the extensive references therein. However, these algorithms in general are only guaranteed to compute either a B-stationary point under the nondegeneracy (lower-level strict complementarity) assumption, which is somewhat restrictive in practice, or a C-stationary point for the problem, rather than the desired B-stationary point. Scholtes and Stöhr [13] showed that a trust region method converges to a B-stationary point, provided that the trust region radii do not tend to zero. However, it is not clear whether we can expect the latter condition to hold in the degenerate case. A modification of this method was proposed in the Ph.D. thesis of Stöhr, for which boundedness of trust region radii away from zero and global convergence to a B-stationary point

*Received by the editors October 3, 1999; accepted for publication (in revised form) June 22, 2001; published electronically February 8, 2002. This research is supported by Scientific Research Grant-in-Aid from the Ministry of Education, Science, Sports and Culture of Japan and by National Science Foundation grant CCR-9731273.

http://www.siam.org/journals/siopt/12-3/36323.html

†Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan (fuku@amp.i.kyoto-u.ac.jp).

‡Department of Mathematics, University of Washington, Seattle, WA 98195 (tseng@math.washington.edu).

were shown under LICQ (for MPEC) and upper-level strict complementarity, plus some mild assumptions [15, Theorem 2.4, Corollary 3.1]. Recently, Fukushima and Pang [3] considered a continuation method based on a smoothed approximation of MPEC and showed, under LICQ (for MPEC) and an "asymptotic weak nondegeneracy" assumption, convergence of a second-order stationary point of the smoothed problem to a B-stationary point of MPEC as the smoothing parameter tends to zero. Subsequently, Scholtes [12] showed an analogous result for a method based on another smoothed approximation of MPEC. However, the algorithms considered in [3, 12] are conceptual and, with the exception of Stöhr's method, global convergence to a B-stationary point has yet to be established for an implementable algorithm for solving (degenerate) MPECs. A piecewise sequential quadratic programming algorithm of Luo, Pang, and Ralph [7] is shown to be locally superlinearly/quadratically convergent to a B-stationary point under LICQ plus a second-order sufficient condition [7, Corollary 6.4.4], [8].

The purpose of this paper is to propose an implementable ϵ-active set algorithm for solving MPEC (1) and establish global convergence of the proposed algorithm to a B-stationary point of (1) under a uniform LICQ on the ϵ-feasible set. Moreover, we show that if the objective function is quadratic and ϵ is set to zero, then the algorithm terminates finitely at a B-stationary point of (1).

A few words about notation: Since functions G_i, H_i, g_j, and h_l are all assumed to be affine, the gradients of these functions are constant vectors. Nevertheless, we will throughout write $\nabla G_i(z^k)$, $\nabla H_i(z^k)$, etc., to specify the point under consideration. This will better illustrate the nature of the proposed algorithm and might suggest possible extension to problems involving nonlinear constraints. Throughout, $\|\cdot\|_1$ and $\|\cdot\|$ denote, respectively, the 1-norm and the 2-norm.

2. An ϵ-active set and B-stationary point. Let $\mathcal{P}$ denote the set of all pairs (A, B) such that A and B are subsets of $\{1, \ldots, m\}$ and $A \cup B = \{1, \ldots, m\}$. Fix $\epsilon \geq 0$. For each $(A, B) \in \mathcal{P}$, define the restricted ϵ-feasible set:

$$\begin{aligned}
\mathcal{F}_\epsilon[A,B] \;:=\; \{\, z \in \Re^n : \quad & \epsilon \geq G_i(z) \geq 0, \quad i \in A, \\
& G_i(z) \geq 0, \quad i \in B \setminus A, \\
& \epsilon \geq H_i(z) \geq 0, \quad i \in B, \\
& H_i(z) \geq 0, \quad i \in A \setminus B, \\
& g_j(z) \leq 0, \quad j = 1, \ldots, p, \\
& h_l(z) = 0, \quad l = 1, \ldots, q \,\}.
\end{aligned}$$

Note that the sets $\{\mathcal{F}_\epsilon[A,B] : (A,B) \in \mathcal{P}\}$ are not mutually disjoint. Define the ϵ-feasible set for MPEC (1):

$$\mathcal{F}_\epsilon := \bigcup_{(A,B)\in\mathcal{P}} \mathcal{F}_\epsilon[A,B].$$

Then, for $\epsilon = 0$, $\mathcal{F}_\epsilon$ is just the feasible set for MPEC (1). For each $z \in \mathcal{F}_\epsilon$, define the ϵ-active index sets:

$$\begin{aligned}
A_\epsilon(z) &:= \{i : G_i(z) \leq \epsilon\}, \\
B_\epsilon(z) &:= \{i : H_i(z) \leq \epsilon\}, \\
I_\epsilon(z) &:= \{j : g_j(z) \geq -\epsilon\}.
\end{aligned}$$

By the definition of $\mathcal{F}_\epsilon$, we always have $A_\epsilon(z) \cup B_\epsilon(z) = \{1, \ldots, m\}$.

For any feasible solution $\bar{z} \in \mathcal{F}_0$ of MPEC (1), let $\mathcal{T}(\bar{z}; \mathcal{F}_0)$ denote the tangent cone of $\mathcal{F}_0$ at $\bar{z}$. Then $\bar{z}$ is called a *B-stationary point* of MPEC (1) if it satisfies

$$\nabla f(\bar{z})^T d \geq 0 \quad \forall\, d \in \mathcal{T}(\bar{z}; \mathcal{F}_0).$$

In MPEC, the tangent cone $\mathcal{T}(\bar{z}; \mathcal{F}_0)$ is normally represented as a finite union of closed convex cones [7] and hence, in general, is nonconvex, unless the nondegeneracy (lower-level strict complementarity) condition is satisfied. This fact gives rise to a combinatorial nature of MPEC that makes a problem intractable. Recent attempts to identify a favorable class of MPECs have been focused on constraint qualifications pertaining to MPEC which enable us to characterize a B-stationary point in a simple and convenient manner [8, 11, 14, 16]. In particular, each feasible solution $\bar{z}$ of MPEC (1) is a feasible solution of the restricted problem:

$$\begin{array}{lll} \text{minimize} & f(z) & \\ \text{subject to} & G_i(z) = 0, & i \in A, \\ & G_i(z) \geq 0, & i \in B \setminus A, \\ & H_i(z) = 0, & i \in B, \\ & H_i(z) \geq 0, & i \in A \setminus B, \\ & g_j(z) \leq 0, & j = 1, \ldots, p, \\ & h_l(z) = 0, & l = 1, \ldots, q, \end{array} \tag{2}$$

with $A = A_0(\bar{z})$ and $B = B_0(\bar{z})$. We associate with $\bar{z}$ a *relaxed problem*, denoted $\mathrm{R}(\bar{z})$, which is obtained from the restricted problem (2) by replacing the equality constraints $G_i(z) = 0$ and $H_i(z) = 0$ for $i \in A \cap B$ with the inequality constraints $G_i(z) \geq 0$ and $H_i(z) \geq 0$. The restricted problem and the relaxed problem are both ordinary nonlinear programs with linear constraints, and LICQ for one implies LICQ for the other. Moreover, $\bar{z}$ is a Karush–Kuhn–Tucker (KKT) point of the relaxed problem $\mathrm{R}(\bar{z})$ if and only if $\bar{z}$ is a KKT point of the restricted problem (2) with *nonnegative* KKT multipliers associated with the equality constraints $G_i(z) = 0$ and $H_i(z) = 0$ for $i \in A \cap B$. Notice that the above definition of a B-stationary point, also used in [3, 11], coincides with the notion of a (primal) stationary point used in [7, p. 115] and [8]. This definition differs from one used in [14], although, under the LICQ for the restricted and relaxed problems, the two definitions are equivalent.

The following theorem, which links KKT points of the relaxed problem to B-stationary points of MPEC, plays an essential role in our analysis.

THEOREM 2.1. *Let $\bar{z}$ be a feasible solution of MPEC* (1) *such that the LICQ holds for the relaxed problem $R(\bar{z})$. Then $\bar{z}$ is a KKT point of the relaxed problem $R(\bar{z})$ if and only if $\bar{z}$ is a B-stationary point of MPEC* (1).

This theorem has been proved under a more general setting in [11] (see also [7, Proposition 4.3.7], [8], [14], [16, p. 384] for related results and discussions). The significance of Theorem 2.1 lies in showing that, under the LICQ for the relaxed problem $\mathrm{R}(\bar{z})$, B-stationarity for MPEC can be completely characterized by the KKT conditions for problem $\mathrm{R}(\bar{z})$, which is an ordinary nonlinear program. This observation, first suggested in [7, Proposition 4.3.7] and clarified in [8], has paved the way for developing conceptual methods that generate a sequence converging to a B-stationary point of MPEC under the LICQ for the relaxed problem [3, 12]. It also motivated

in [8] the development and the local superlinear/quadratic convergence analysis of a piecewise sequential quadratic programming method.

Theorem 2.1 motivates the following conceptual active set method for computing a B-stationary point of MPEC. For a given index set pair $(A, B) \in \mathcal{P}$, let $\hat{z}$ be a KKT point of the restricted problem (2). Then $A_0(\hat{z}) \supseteq A$ and $B_0(\hat{z}) \supseteq B$, so that if the KKT multipliers associated with the equality constraints $G_i(z) = 0$, $i \in A \cap B_0(\hat{z})$, and $H_i(z) = 0$, $i \in A_0(\hat{z}) \cap B$, are all nonnegative, then $\hat{z}$ is a KKT point of $\mathrm{R}(\hat{z})$ and we terminate the method. Otherwise, we choose one of these equality constraints with negative multiplier and we drop the corresponding index from either A or B. Under the LICQ, the restricted problem corresponding to the resulting index set pair (A, B) has a feasible descent direction d from $\hat{z}$, which we then use to obtain a feasible solution z^{new} with $f(z^{new}) < f(\hat{z})$. We replace A and B by $A_0(z^{new})$ and $B_0(z^{new})$, respectively, and reiterate.

The above active set method is conceptual since, in practice, the KKT point $\hat{z}$ can be computed only approximately. To make it implementable, we need a notion of an approximate KKT point which we will make precise in the next section; see (4)–(7). Also, to prevent cycling of the active sets, we need to be able to take a sufficiently large step from $\hat{z}$ in the descent direction d so as to achieve sufficient descent each time the active sets change. This is achieved by working with ϵ-feasible set $\mathcal{F}_\epsilon$ and ϵ-active index sets $A_\epsilon(\cdot), B_\epsilon(\cdot), I_\epsilon(\cdot)$ for $\epsilon \geq 0$. In particular, the stepsize will be at least in the order of ϵ; see (13). Accordingly, we assume the following *uniform LICQ* on $\mathcal{F}_\epsilon$ for some fixed $\epsilon \geq 0$:

$$\eta < \left\| \sum_{i \in A_\epsilon(z)} \nabla G_i(z) v_i + \sum_{i \in B_\epsilon(z)} \nabla H_i(z) w_i - \sum_{j \in I_\epsilon(z)} \nabla g_j(z) \lambda_j - \sum_{l=1}^{q} \nabla h_l(z) \mu_l \right\|_1$$

$$\text{whenever } \; z \in \mathcal{F}_\epsilon \; \text{ and } \; \max_{i,j,l} \{|v_i|, |w_i|, |\lambda_j|, |\mu_l|\} > 1,$$

where $\eta > 0$ is some constant. Notice that since G_i, H_i, g_j, h_l are affine for all i, j, l, we can equivalently replace η by 0 in this assumption. However, the constant η will play a useful role in our algorithm and its analysis. Recall that $\bar{z} \in \mathcal{F}_0$ implies $A_0(\bar{z}) \cup B_0(\bar{z}) = \{1, \ldots, m\}$. Hence, for $\epsilon = 0$, the uniform LICQ on $\mathcal{F}_0$ essentially amounts to LICQ (in the sense of ordinary nonlinear programming) for the relaxed problem $\mathrm{R}(\bar{z})$ being satisfied at every feasible solution $\bar{z}$ of MPEC (1).

3. An ϵ-active set algorithm. In this section, we describe the ϵ-active set algorithm for solving MPEC (1). At each iteration $k \in \{0, 1, \ldots\}$, given the current iterate $z^k \in \mathcal{F}_{\epsilon_k}[A^k, B^k]$ with some $\epsilon_k \in [0, \epsilon]$ and index set pair $(A^k, B^k) \in \mathcal{P}$, we compute an approximate KKT point of the following subproblem (compare with (2)):

$$\begin{array}{lll} \text{minimize} & f(z) & \\ \text{subject to} & G_i(z) = G_i(z^k), & i \in A^k, \\ & G_i(z) \geq 0, & i \in B^k \setminus A^k, \\ & H_i(z) = H_i(z^k), & i \in B^k, \\ & H_i(z) \geq 0, & i \in A^k \setminus B^k, \\ & g_j(z) \leq 0, & j = 1, \ldots, p, \\ & h_l(z) = 0, & l = 1, \ldots, q. \end{array} \tag{3}$$

This is a linearly constrained nonlinear program with z^k as a feasible solution, so we can compute using a feasible descent algorithm with starting point z^k.

Theorem 2.1 and the discussion preceding it suggest that, under uniform LICQ, a KKT point $\hat{z}^k$ of (3) is an approximate B-stationary point of MPEC (1), provided that ϵ_k is sufficiently small and the KKT multipliers associated with the equality constraints $G_i(z) = G_i(z^k)$, $i \in A^k \cap B_{\epsilon_k}(\hat{z}^k)$, and $H_i(z) = H_i(z^k)$, $i \in A_{\epsilon_k}(\hat{z}^k) \cap B^k$, are nearly nonnegative. Note that the latter condition implies that the KKT multipliers associated with the constraints involving G_i and H_i for $i \in A_{\epsilon_k}(\hat{z}^k) \cap B_{\epsilon_k}(\hat{z}^k)$ are nearly nonnegative. This is because the constraints involving G_i, $i \in A_{\epsilon_k}(\hat{z}^k) \setminus A^k$, are inequality constraints in (3), so the corresponding KKT multipliers are nonnegative, and the same is true for the constraints involving H_i, $i \in B_{\epsilon_k}(\hat{z}^k) \setminus B^k$. Our algorithm seeks such a point $\hat{z}^k$ by successively generating an approximate KKT point of the subproblem (3) with dynamically adjusted index sets A^k and B^k.

A rough sketch of our algorithm is stated below. A more detailed description will be given shortly. In what follows, we assume uniform LICQ on $\mathcal{F}_\epsilon$ for some $\epsilon \geq 0$.

Step 0. Choose initial $\epsilon_0 \in [0, \epsilon]$, $(A^0, B^0) \in \mathcal{P}$, and $z^0 \in \mathcal{F}_{\epsilon_0}[A^0, B^0]$. Let $k := 0$.

Step 1. Compute an approximate KKT point $\hat{z}^k$ of the subproblem (3).

Step 2. If one of the KKT multipliers associated with the equality constraints $G_i(z) = G_i(z^k)$, $i \in A^k \cap B_{\epsilon_k}(\hat{z}^k)$, and $H_i(z) = H_i(z^k)$, $i \in A_{\epsilon_k}(\hat{z}^k) \cap B^k$, is below a negative threshold, remove the corresponding index from the ϵ_k-active set so that the objective function value may be decreased sufficiently by moving from $\hat{z}^k$ along a descent direction d^k. This yields a new ϵ_k-feasible point $\tilde{z}^k$ with a lower objective value; proceed to Step 3. Otherwise, set $\tilde{z}^k = \hat{z}^k$ and proceed to Step 3.

Step 3. Decrease ϵ_k to obtain ϵ_{k+1}, and project $\tilde{z}^k$ onto $\mathcal{F}_{\epsilon_{k+1}}[A_{\epsilon_k}(\tilde{z}^k), B_{\epsilon_k}(\tilde{z}^k)]$ to obtain an ϵ_{k+1}-feasible solution z^{k+1}. Set A^{k+1}, B^{k+1} to be the corresponding ϵ_{k+1}-active index sets. Increment k by 1 and go to Step 1.

The algorithm thus generates a sequence $\{z^k\}$ such that $\{f(z^k)\}$ is almost decreasing, while maintaining ϵ_k-feasibility to the original MPEC (1) for a decreasing sequence $\{\epsilon_k\}$. Now we describe Steps 1, 2, and 3 in detail. Both the accuracy of the approximate KKT point for subproblem (3) and the negative threshold for the multipliers will be controlled by a parameter $\nu_k \in [0, 1]$ that, like ϵ_k, is decreased to zero with k.

In Step 1, we compute a point $\hat{z}^k$ to be an approximate KKT point of the subproblem (3) in the sense that

$$\hat{z}^k \in \mathcal{F}_{\epsilon_k}[A^k, B^k], \qquad f(\hat{z}^k) \leq f(z^k), \tag{4}$$

and, for some $\delta_k \in [0, \eta\nu_k/2]$, there exist multipliers $(v_i^k)_{i \in \hat{A}^k}$, $(w_i^k)_{i \in \hat{B}^k}$, $(\lambda_j^k)_{j \in I^k}$, $(\mu_l^k)_{l=1}^q$ satisfying

$$\begin{aligned} \|r^k\|_1 &\leq \delta_k, \\ v_i^k &\geq 0 \qquad \forall i \in \hat{A}^k \setminus A^k, \\ w_i^k &\geq 0 \qquad \forall i \in \hat{B}^k \setminus B^k, \\ \lambda_j^k &\geq 0 \qquad \forall j \in I^k, \end{aligned} \tag{5}$$

where η is the constant in the uniform LICQ, r^k is the residual vector given by

$$
\begin{aligned}
r^k \;:=\; & -\nabla f(\hat{z}^k) + \sum_{i\in\hat{A}^k} \nabla G_i(\hat{z}^k) v_i^k + \sum_{i\in\hat{B}^k} \nabla H_i(\hat{z}^k) w_i^k \\
& - \sum_{j\in I^k} \nabla g_j(\hat{z}^k)\lambda_j^k - \sum_{l=1}^{q} \nabla h_l(\hat{z}^k)\mu_l^k,
\end{aligned} \tag{6}
$$

and $\hat{A}^k$, $\hat{B}^k$, I^k are ϵ_k-active index sets given by

$$
\hat{A}^k := A_{\epsilon_k}(\hat{z}^k), \qquad \hat{B}^k := B_{\epsilon_k}(\hat{z}^k), \qquad I^k := I_{\epsilon_k}(\hat{z}^k). \tag{7}
$$

In subsection 3.1, we discuss how such $\hat{z}^k$ can be computed in finite time by applying a feasible descent method to the subproblem (3), starting at z^k. Notice that subproblem (3) need not have a unique KKT point.

In Step 2, we observe from (7) and $\epsilon_k \le \epsilon$ that

$$
A_\epsilon(\hat{z}^k) \supseteq \hat{A}^k \supseteq A^k, \qquad B_\epsilon(\hat{z}^k) \supseteq \hat{B}^k \supseteq B^k, \qquad I_\epsilon(\hat{z}^k) \supseteq I^k. \tag{8}
$$

Moreover, $A^k \cup B^k = \hat{A}^k \cup \hat{B}^k = \{1,\dots,m\}$. By (5), we have three cases:

(a) $v_{i_k}^k < -\nu_k$ for some index $i_k \in A^k \cap \hat{B}^k$;

(b) $w_{i_k}^k < -\nu_k$ for some index $i_k \in \hat{A}^k \cap B^k$;

(c) $v_i^k \ge -\nu_k$ and $w_i^k \ge -\nu_k$ for all $i \in \hat{A}^k \cap \hat{B}^k$.

Note that (a) and (b) are not mutually exclusive.

In cases (a) and (b), we find a descent direction d^k for the objective function f at $\hat{z}^k$, along which ϵ_k-feasibility for MPEC (1) is maintained. In particular, when $\nu_k > 0$, d^k is a solution of the linear system

$$
\begin{aligned}
\nabla f(\hat{z}^k)^T d &\le -\eta\nu_k/2, \\
-e \;\le\; d &\le e, \\
\nabla G_{i_k}(\hat{z}^k)^T d &\ge 0 && \text{in case (a)}, \\
\nabla G_{i_k}(\hat{z}^k)^T d &= 0 && \text{in case (b)}, \\
\nabla G_i(\hat{z}^k)^T d &= 0 && \forall i \in \hat{A}^k \setminus \{i_k\}, \\
\nabla G_i(\hat{z}^k)^T d &\ge -G_i(\hat{z}^k) && \forall i \in A_\epsilon(\hat{z}^k) \setminus \hat{A}^k, \\
\nabla H_{i_k}(\hat{z}^k)^T d &= 0 && \text{in case (a)}, \\
\nabla H_{i_k}(\hat{z}^k)^T d &\ge 0 && \text{in case (b)}, \\
\nabla H_i(\hat{z}^k)^T d &= 0 && \forall i \in \hat{B}^k \setminus \{i_k\}, \\
\nabla H_i(\hat{z}^k)^T d &\ge -H_i(\hat{z}^k) && \forall i \in B_\epsilon(\hat{z}^k) \setminus \hat{B}^k, \\
\nabla g_j(\hat{z}^k)^T d &\le 0 && \forall j \in I^k, \\
\nabla g_j(\hat{z}^k)^T d &\le -g_j(\hat{z}^k) && \forall j \in I_\epsilon(\hat{z}^k) \setminus I^k, \\
\nabla h_l(\hat{z}^k)^T d &= 0 && l = 1,\dots,q;
\end{aligned} \tag{9}
$$

when $\nu_k = 0$, d^k is a solution of (9) with "$\le -\eta\nu_k/2$" replaced by "< 0." Here e denotes the vector of 1's. The existence of d^k is justified by Lemma 3.2 in subsection 3.2. We note that the nonpositive quantities $-G_i(\hat{z}^k)$, $-H_i(\hat{z}^k)$, $-g_j(\hat{z}^k)$ on the right-hand side of (9) can alternatively be replaced by zero. However, the resulting linear system would have a smaller solution set. Once d^k is found, we compute the maximum ϵ_k-feasible stepsize

$$
\bar{t}_k := \min\{t_{\max}, \sup\{t : \hat{z}^k + t d^k \in \mathcal{F}_{\epsilon_k}\}\}, \tag{10}
$$

where $t_{\max} > 0$ is a chosen constant, and set

$$\nu_{k+1} := \nu_k, \qquad \tilde{z}^k := \hat{z}^k + t_k d^k, \tag{11}$$

where t_k is given by the Armijo rule:

$$\begin{aligned} t_k \;=\; & \text{largest } t \in \{\bar{t}_k, \sigma_1\bar{t}_k, \sigma_1^2\bar{t}_k, \ldots\} \text{ satisfying} \\ & f(\hat{z}^k + td^k) \le f(\hat{z}^k) + \sigma_2 t \nabla f(\hat{z}^k)^T d^k, \end{aligned} \tag{12}$$

with $\sigma_1 \in (0,1)$ and $\sigma_2 \in (0,1)$ chosen constants. We then proceed to Step 3. Notice that $\bar{t}_k$ is computable by a minimum ratio formula. Moreover, we have

$$\bar{t}_k \ge \min\Big\{t_{\max},\, \epsilon_k / \max_{i,j}\{\|\nabla G_i(\hat{z}^k)\|_1, \|\nabla H_i(\hat{z}^k)\|_1, \|\nabla g_j(\hat{z}^k)\|_1\}\Big\} \tag{13}$$

and, by a standard argument for the Armijo stepsize rule, $t_k > 0$.

In case (c), if ϵ_k and ν_k are both below some chosen tolerances $\epsilon_{\rm tol}$ and $\nu_{\rm tol}$, respectively, then we terminate the method and output $\hat{z}^k$ as an approximate B-stationary point of MPEC (1). Otherwise, we decrease ν_k by setting

$$\nu_{k+1} := \omega_1 \nu_k, \qquad \tilde{z}^k := \hat{z}^k, \tag{14}$$

where $\omega_1 \in (0,1)$ is a chosen constant. We then proceed to Step 3.

In Step 3, we decrease ϵ_k by setting

$$\epsilon_{k+1} := \min\{\max\{\rho(\nu_{k+1}), \omega_2\epsilon_k\}, \epsilon_k\} = \mathrm{mid}\{\omega_2\epsilon_k, \rho(\nu_{k+1}), \epsilon_k\}, \tag{15}$$

where $\omega_2 \in (0,1)$ is a chosen constant and $\rho : \Re_+ \to \Re_+$ is a chosen continuous function satisfying $\rho(\nu) = 0 \;\Leftrightarrow\; \nu = 0$. Thus, the two accuracy parameters ϵ_k and ν_k are linked through ρ. We then update z^k and (A^k, B^k) by

$$\begin{aligned} z^{k+1} \;&:=\; \arg \min_{z \in \mathcal{F}_{\epsilon_{k+1}}[\tilde{A}^k, \tilde{B}^k]} \|z - \tilde{z}^k\|, \\ (A^{k+1}, B^{k+1}) \;&:=\; (A_{\epsilon_{k+1}}(z^{k+1}), B_{\epsilon_{k+1}}(z^{k+1})), \end{aligned} \tag{16}$$

where $\tilde{A}^k := A_{\epsilon_k}(\tilde{z}^k)$ and $\tilde{B}^k := B_{\epsilon_k}(\tilde{z}^k)$, and return to Step 1.

Summarizing the above arguments, we formally state the algorithm as follows.

THE ϵ-ACTIVE SET ALGORITHM FOR MPEC (1).

Step 0. Choose arbitrary constants $\sigma_1 \in (0,1), \sigma_2 \in (0,1)$, $t_{\max} \in (0,\infty)$, $\omega_1 \in (0,1)$, $\omega_2 \in (0,1)$, and a continuous function $\rho : \Re_+ \to \Re_+$ such that $\rho(\nu) = 0 \;\Leftrightarrow\; \nu = 0$. Choose $\epsilon_0 \in [0, \epsilon]$, $\epsilon_{\rm tol} \in [0, \epsilon_0]$, $\nu_0 \in [0,1]$, $\nu_{\rm tol} \in [0, \nu_0]$, $(A^0, B^0) \in \mathcal{P}$, and $z^0 \in \mathcal{F}_{\epsilon_0}[A^0, B^0]$. Let $k := 0$.

Step 1. Compute an approximate KKT point $\hat{z}^k \in \mathcal{F}_{\epsilon_k}[A^k, B^k]$ of the subproblem (3) in the sense that (4)–(6) hold, with $\hat{A}^k, \hat{B}^k, I^k$ given by (7). Proceed to Step 2.

Step 2. In case (a) or (b), find a d^k satisfying (9), set $\bar{t}_k$ by (10) and set t_k by (12), determine ν_{k+1} and $\tilde{z}^k$ by (11), and proceed to Step 3. In case (c), if $\epsilon_k \le \epsilon_{\rm tol}$ and $\nu_k \le \nu_{\rm tol}$, then terminate; otherwise, determine ν_{k+1} and $\tilde{z}^k$ by (14), and proceed to Step 3.

Step 3. Determine ϵ_{k+1} by (15), and z^{k+1} and (A^{k+1}, B^{k+1}) by (16). Increment k by 1 and return to Step 1.

As described, the algorithm requires knowledge of η in choosing δ_k. If η is not known, we can estimate η on-line using the following back-tracking scheme with an initial η chosen arbitrarily: In Step 3, if (a) or (b) occurs but (9) does not have a solution, then decrease η by a constant fraction and repeat iteration k. Under the uniform LICQ, η would be decreased only a finite number of times (see the proof of Lemma 3.2), and the convergence result given in Theorem 4.1 would still hold for this variant of the algorithm.

The initial z^0 can be found by computing z^0 to be an ϵ_0-feasible solution of (1), i.e., z^0 is a feasible solution of (1) with $G_i(z)H_i(z) = 0$ replaced by $\min\{G_i(z), H_i(z)\} \le \epsilon_0$. Then set $A^0 = A_{\epsilon_0}(z^0)$, $B^0 = B_{\epsilon_0}(z^0)$. Thus, computing z^0 requires the approximate solution of a linear complementarity problem, for which many algorithms are available.

In (14), instead of setting ν_{k+1} to $\omega_1\nu_k$, one can more generally set ν_{k+1} to be the term after ν_k along some prespecified decreasing sequence (not necessarily geometric) tending to zero. The term $\omega_2\epsilon_k$ in (15) can be similarly generalized. The updating rule for ϵ_k enforces that ϵ_k goes to zero at the rate of $\rho(\nu_k)$. Thus ϵ_k, which measures primal feasibility, and ν_k, which measures dual feasibility, can go to zero at different rates, depending on the choice of the function ρ.

3.1. Computing $\hat{z}^k$.

LEMMA 3.1. *For any $\hat{A}^k \supseteq A^k$, $\hat{B}^k \supseteq B^k$, $I^k \subseteq \{1, \ldots, p\}$, there exist $(v_i^k)_{i\in\hat{A}^k}$, $(w_i^k)_{i\in\hat{B}^k}$, $(\lambda_j^k)_{j\in I^k}$, $(\mu_l^k)_{l=1}^q$ satisfying* (5) *with $\delta_k > 0$ (respectively, $\delta_k = 0$) and r^k given by* (6) *whenever the following linear system (respectively, the following linear system with "$\le -\delta_k$" replaced by "< 0") has no solution:*

$$
\begin{aligned}
\nabla f(\hat{z}^k)^T d &\le -\delta_k, \\
-e \;\le\; d &\;\le\; e, \\
\nabla G_i(\hat{z}^k)^T d &= 0 \qquad \forall\, i \in A^k, \\
\nabla G_i(\hat{z}^k)^T d &\ge 0 \qquad \forall\, i \in \hat{A}^k \setminus A^k, \\
\nabla H_i(\hat{z}^k)^T d &= 0 \qquad \forall\, i \in B^k, \\
\nabla H_i(\hat{z}^k)^T d &\ge 0 \qquad \forall\, i \in \hat{B}^k \setminus B^k, \\
\nabla g_j(\hat{z}^k)^T d &\le 0 \qquad \forall j \in I^k, \\
\nabla h_l(\hat{z}^k)^T d &= 0 \qquad \forall\, l = 1, \ldots, q,
\end{aligned}
\tag{17}
$$

where e denotes the vector of 1*'s.*

Proof. First observe that

$$\hat{A}^k = A^k \cup (\hat{A}^k \setminus A^k), \quad \hat{B}^k = B^k \cup (\hat{B}^k \setminus B^k).$$

Then, by Farkas's lemma, system (17) has no solution if and only if the following dual linear system has a solution:

$$
\begin{aligned}
0 \;&>\; -\delta_k\lambda_0 + e^T\pi^+ + e^T\pi^-,\\
0 \;&=\; \nabla f(\hat{z}^k)\lambda_0 + \pi^+ - \pi^- - \sum_{i\in\hat{A}^k} \nabla G_i(\hat{z}^k)v_i - \sum_{i\in\hat{B}^k} \nabla H_i(\hat{z}^k)w_i\\
&\quad + \sum_{j\in I^k} \nabla g_j(\hat{z}^k)\lambda_j + \sum_{l=1}^{q} \nabla h_l(\hat{z}^k)\mu_l, \qquad (18)\\
&\lambda_0 \geq 0, \quad \pi^+ \geq 0, \quad \pi^- \geq 0,\\
&v_i \geq 0 \quad \forall\, i \in \hat{A}^k\backslash A^k,\\
&w_i \geq 0 \quad \forall\, i \in \hat{B}^k\backslash B^k,\\
&\lambda_j \geq 0 \quad \forall\, j \in I^k.
\end{aligned}
$$

The first inequality in the above system implies that $\lambda_0 > 0$ and that $(e^T\pi^+ + e^T\pi^-)/\lambda_0 \leq \delta_k$. Then, dividing the whole system by λ_0 and setting $r^k = (\pi^+ - \pi^-)/\lambda_0$ and $v_i^k = v_i/\lambda_0$, $w_i^k = w_i/\lambda_0$, $\lambda_j^k = \lambda_j/\lambda_0$, $\mu_l^k = \mu_l/\lambda_0$, we obtain condition (5) with r^k given by (6), because

$$\|r^k\|_1 = \|\pi^+ - \pi^-\|_1/\lambda_0 \leq (\|\pi^+\|_1 + \|\pi^-\|_1)/\lambda_0 = (e^T\pi^+ + e^T\pi^-)/\lambda_0 \leq \delta_k.$$

Suppose that (17) with "$\leq -\delta_k$" replaced by "< 0" has no solution. Then (17) has no solution for every $\delta_k > 0$, so the preceding argument shows that, for every $\delta_k > 0$, there exists $r^k \in \Re^n$ such that $\|r^k\|_1 \leq \delta_k$ and the following linear system (cf. (5) and (6)) has a solution:

$$
\begin{aligned}
v_i \;&\geq 0 \qquad \forall i \in \hat{A}^k \setminus A^k,\\
w_i \;&\geq 0 \qquad \forall i \in \hat{B}^k \setminus B^k,\\
\lambda_j \;&\geq 0 \qquad \forall j \in I^k,
\end{aligned}
$$

$$\sum_{i\in\hat{A}^k} \nabla G_i(\hat{z}^k)v_i + \sum_{i\in\hat{B}^k} \nabla H_i(\hat{z}^k)w_i - \sum_{j\in I^k} \nabla g_j(\hat{z}^k)\lambda_j - \sum_{l=1}^{q} \nabla h_l(\hat{z}^k)\mu_l = \nabla f(\hat{z}^k) + r^k.$$

By a result of Hoffman [4], the least 2-norm solution of the above system is in the order of the right-hand side. Since $r^k \to 0$ as $\delta_k \to 0$, so that the right-hand side converges, then the least 2-norm solution of the above system is bounded and any cluster point is a solution of this system with $r^k = 0$. □

For the case of $\epsilon_k > 0$ and $\nu_k > 0$, a point $\hat{z}^k$ such that (4) holds and (17) has no solution is computable in finite time by any feasible descent method for the linearly constrained subproblem (3), with starting point z^k, whose generated points converge in subsequence to a KKT point of (3). Specifically, it would be enough to find a point $\hat{z}^k$ sufficiently near to a KKT point z of problem (3) so that $A_0(z) \subseteq \hat{A}^k$, $B_0(z) \subseteq \hat{B}^k$, $I_0(z) \subseteq I^k$, and $\|\nabla f(\hat{z}^k) - \nabla f(z)\|_1 < \delta_k$. For such $\hat{z}^k$, the system (17) has no solution, because any solution of (17) would be a feasible descent direction for subproblem (3) at z, contradicting z being a KKT point of (3). For the case of $\epsilon_k = 0$ and $\nu_k = 0$ and f quadratic, such a $\hat{z}^k$ is a KKT point of (3) and is computable in finite time by using a feasible descent method coupled with an active set identification strategy.

3.2. Existence of d^k.

LEMMA 3.2. *Suppose there exists an index $i_k \in A^k \cap B^k$ such that* (a) $v_{i_k}^k < -\nu_k$ *or* (b) $w_{i_k}^k < -\nu_k$. *Then, under the uniform LICQ, the linear system* (9) *has a solution*

whenever $\nu_k > 0$, and the linear system (9) *with "$\le -\eta\nu_k/2$" replaced by "< 0" has a solution when $\nu_k = 0$.*

Proof. Let $\nu_k > 0$, and suppose that system (9) does not have a solution. Then, by Farkas's lemma, its dual system has a solution so that, by the same argument as in the proof of Lemma 3.1 and using (8), we can show that there exist $(v_i)_{i\in A_\epsilon(\hat z^k)}$, $(w_i)_{i\in A_\epsilon(\hat z^k)}$, $(\lambda_j)_{j\in I_\epsilon(\hat z^k)}$, $(\mu_l)_{l=1}^q$ satisfying

$$\begin{gathered}\|r\|_1 \le \eta\nu_k/2,\\ v_{i_k} \ge 0 \ \text{ in case (a)}, \quad w_{i_k} \ge 0 \ \text{ in case (b)},\\ v_i \ge 0 \quad \forall\, i \in A_\epsilon(\hat z^k) \setminus \hat A^k,\\ w_i \ge 0 \quad \forall\, i \in B_\epsilon(\hat z^k) \setminus \hat B^k,\\ \lambda_j \ge 0 \quad \forall\, j \in I_\epsilon(\hat z^k),\end{gathered}$$

where

$$\begin{aligned} r \ :=\ & -\nabla f(\hat z^k) + \sum_{i\in A_\epsilon(\hat z^k)} \nabla G_i(\hat z^k) v_i + \sum_{i\in B_\epsilon(\hat z^k)} \nabla H_i(\hat z^k) w_i \\ & - \sum_{j\in I_\epsilon(\hat z^k)} \nabla g_j(\hat z^k)\lambda_j - \sum_{l=1}^q \nabla h_l(\hat z^k)\mu_l. \end{aligned} \tag{19}$$

Subtracting (6) from (19) yields

$$\begin{aligned} r - r^k \ =\ & -\sum_{i\in A_\epsilon(\hat z^k)} \nabla G_i(\hat z^k)(v_i^k - v_i) - \sum_{i\in B_\epsilon(\hat z^k)} \nabla H_i(\hat z^k)(w_i^k - w_i) \\ & + \sum_{j\in I_\epsilon(\hat z^k)} \nabla g_j(\hat z^k)(\lambda_j^k - \lambda_j) + \sum_{l=1}^q \nabla h_l(\hat z^k)(\mu_l^k - \mu_l), \end{aligned}$$

with v_i^k (respectively, w_i^k, λ_j^k) set to zero if it is not defined in (5). Since $v_{i_k} - v_{i_k}^k > \nu_k$ in case (a) and $w_{i_k} - w_{i_k}^k > \nu_k$ in case (b), while $\delta_k \le \eta\nu_k/2$ so that $\|r^k\|_1 \le \delta_k \le \eta\nu_k/2$, we would have $\|r - r^k\|_1 \le \|r\|_1 + \|r^k\|_1 \le \eta\nu_k$, contradicting the uniform LICQ at $\hat z^k$. Thus, system (9) has a solution.

When $\nu_k = 0$, again by the same argument as in the proof of Lemma 3.1, we have $r = 0$ in (19). Since $\nu_k = 0$ implies $\delta_k = 0$, we have $r^k = 0$ in (5). Hence, by the same reasoning as above, we can derive a contradiction. The proof is complete. ∎

4. Convergence to a B-stationary point. To establish convergence of the ϵ-active set algorithm, we make the following assumptions:

A1. The objective function f is bounded below on $\mathcal{F}_\epsilon$.

A2. The uniform LICQ on $\mathcal{F}_\epsilon$ holds.

A3. The generated sequences $\{z^k\}$ and $\{\hat z^k\}$ are bounded.

Assumption A3 would be implied by the boundedness of $\mathcal{F}_\epsilon$. A more general sufficient condition for this assumption to hold will be given in Lemma 4.2 at the end of this section. In our convergence analysis, we suppose that $\epsilon_{\rm tol} = \nu_{\rm tol} = 0$ when $\epsilon_0 > 0$ and $\nu_0 > 0$, so that the algorithm may generate an infinite sequence $\{z^k\}$.

THEOREM 4.1. *Under assumptions* A1–A3*, the following hold for the sequence $\{(z^k, \hat z^k, \tilde z^k, \epsilon_k, \nu_k)\}$ generated by the ϵ-active set algorithm:*

(a) *If $\epsilon_0 > 0$, $\nu_0 > 0$, $\epsilon_{\rm tol} = \nu_{\rm tol} = 0$, and f is Lipschitz continuous with constant L on a set Z containing $\{z^k\}$ and $\{\tilde z^k\}$, then $\epsilon_k \downarrow 0$, $\nu_k \downarrow 0$, and every cluster point of $\{z^k\}$ is a B-stationary point of MPEC* (1).

(b) *If $\epsilon_0 = \nu_0 = 0$ and f is quadratic, then there exists a $\bar{k} \in \{0, 1, \ldots\}$ such that $\hat{z}^{\bar{k}}$ is a B-stationary point of MPEC* (1).

Proof. We have from (16) that $z^k \in \mathcal{F}_{\epsilon_k}$ for all k, and from (4), (11), (12), (14) that

$$f(\tilde{z}^k) \le f(\hat{z}^k) \le f(z^k) \quad \text{and} \quad \hat{z}^k, \tilde{z}^k \in \mathcal{F}_{\epsilon_k} \qquad \forall\, k. \tag{20}$$

Let $\mathcal{K} := \{k : \text{Case (a) or (b) occurs in iteration } k\}$ and $\mathcal{K}' := \{k : \text{Case (c) occurs in iteration } k\}$.

(a) Suppose $\epsilon_0 > 0, \nu_0 > 0$, and f is Lipschitz continuous with constant L on set Z containing $\{z^k\}$ and $\{\tilde{z}^k\}$. If $\nu_k \to 0$, then $|\mathcal{K}'| = \infty$, $\delta_k \to 0$, and the updating formula for ϵ_k would imply $\epsilon_k \to 0$, so any cluster point $\bar{z}$ of $\{\hat{z}^k\}_{k \in \mathcal{K}'}$ would be a KKT point of the *relaxed problem* $\mathrm{R}(\bar{z})$, which is a B-stationary point of MPEC (1) under the uniform LICQ. Suppose instead $\nu_k \not\to 0$, so that $|\mathcal{K}'| < \infty$, $|\mathcal{K}| = \infty$, and $\nu = \lim_{k \to \infty} \nu_k > 0$. We will obtain a contradiction below.

For each iteration k, $\tilde{z}^k$ satisfies all constraints defining $\mathcal{F}_{\epsilon_{k+1}}[\tilde{A}^k, \tilde{B}^k]$, except for possibly

$$\begin{aligned} G_i(z) &\le \epsilon_{k+1} \quad \forall\, i \in \tilde{A}^k, \\ H_i(z) &\le \epsilon_{k+1} \quad \forall\, i \in \tilde{B}^k. \end{aligned}$$

Then, by a well-known lemma of Hoffman [4], there exists a constant $\tau > 0$, which depends only on $\nabla G_i, \nabla H_i, \nabla g_j, \nabla h_l$, such that

$$\begin{aligned} \|z^{k+1} - \tilde{z}^k\| &\le \tau \left(\sum_{i \in \tilde{A}^k} \left| [G_i(\tilde{z}^k) - \epsilon_{k+1}]_+ \right| + \sum_{i \in \tilde{B}^k} \left| [H_i(\tilde{z}^k) - \epsilon_{k+1}]_+ \right| \right) \\ &= \tau \left(\sum_{i \in \tilde{A}^k} \left| [G_i(\tilde{z}^k) - \epsilon_{k+1}]_+ - [G_i(\tilde{z}^k) - \epsilon_k]_+ \right| \right. \\ &\qquad \left. + \sum_{i \in \tilde{B}^k} \left| [H_i(\tilde{z}^k) - \epsilon_{k+1}]_+ - [H_i(\tilde{z}^k) - \epsilon_k]_+ \right| \right) \\ &\le \tau \left(\sum_{i \in \tilde{A}^k} |\epsilon_{k+1} - \epsilon_k| + \sum_{i \in \tilde{B}^k} |\epsilon_{k+1} - \epsilon_k| \right) \\ &\le 2\tau m(\epsilon_k - \epsilon_{k+1}), \end{aligned} \tag{21}$$

where the equality follows from the fact that $\tilde{z}^k \in \mathcal{F}_{\epsilon_k}[\tilde{A}^k, \tilde{B}^k]$ and the second inequality uses the nonexpansive property of $[\cdot]_+ := \max\{0, \cdot\}$ with respect to $|\cdot|$. Since f is Lipschitz continuous with constant L on Z containing z^{k+1} and $\tilde{z}^k$, it follows that

$$f(z^{k+1}) \le f(\tilde{z}^k) + L\|z^{k+1} - \tilde{z}^k\| \le f(\tilde{z}^k) + 2L\tau m(\epsilon_k - \epsilon_{k+1}). \tag{22}$$

This together with (20) yields

$$f(z^{k+1}) \le f(z^k) + 2L\tau m(\epsilon_k - \epsilon_{k+1}) \quad \forall\, k.$$

Since, by our assumption, $\{f(z^k)\}$ is bounded below and $\{\epsilon_k\}$ is monotonically nonincreasing and positive, this in turn implies that $\{f(z^k)\}$ converges and so $f(z^{k+1}) -$

$f(z^k) \to 0$. Since, for each $k \in \mathcal{K}$,

$$\begin{aligned} f(\tilde{z}^k) &= f(\hat{z}^k + t_k d^k) \\ &\leq f(\hat{z}^k) + \sigma_2 t_k \nabla f(\hat{z}^k)^T d^k \\ &\leq f(z^k) - \sigma_2 t_k \eta \nu_k / 2 \\ &\leq f(z^k) - \sigma_2 t_k \eta \nu / 2, \end{aligned}$$

this and (22) imply $\{t_k\}_{k \in \mathcal{K}} \to 0$.

The Armijo stepsize rule for determining t_k implies, for each $k \in \mathcal{K}$, either (i) $t_k = \bar{t}_k$ or (ii) $t_k < \bar{t}_k$. Since $\nu = \lim_{k \to \infty} \nu_k > 0$, the updating rule (15) for ϵ_{k+1} implies $\lim_{k \to \infty} \epsilon_k > 0$. So it follows from (13), together with the boundedness of $\{\hat{z}^k\}$, that $\{\bar{t}_k\}$ is bounded away from zero. Since $\{t_k\}_{k \in \mathcal{K}} \to 0$, this implies that case (i) can occur for only a finite number of iterations $k \in \mathcal{K}$, so it must be that case (ii) occurs for all $k \in \mathcal{K}$ sufficiently large, in which case (12) yields

$$f(\hat{z}^k + (t_k/\sigma_1) d^k) - f(\hat{z}^k) > \sigma_2 (t_k/\sigma_1) \nabla f(\hat{z}^k)^T d^k.$$

Since $\{\hat{z}^k\}$ is assumed bounded and $-e \leq d^k \leq e$ for all $k \in \mathcal{K}$, this together with the fact that $\{t_k\}_{k \in \mathcal{K}} \to 0$ would yield in the limit

$$\nabla f(\hat{z}^\infty)^T d^\infty \geq \sigma_2 \nabla f(\hat{z}^\infty)^T d^\infty,$$

where $(\hat{z}^\infty, d^\infty)$ is any cluster point of $\{(\hat{z}^k, d^k)\}_{k \in \mathcal{K}}$. Then we would have from $\sigma_2 \in (0,1)$ that $\nabla f(\hat{z}^\infty)^T d^\infty \geq 0$, contradicting the fact that $\nabla f(\hat{z}^k)^T d^k \leq -\eta \nu_k / 2 \leq -\eta \nu / 2$ for all $k \in \mathcal{K}$.

(b) Suppose $\epsilon_0 = \nu_0 = 0$ and that f is quadratic. Then we have $\epsilon_k = \nu_k = 0$ for all k, and hence the conditions (5) yield that $\hat{z}^k$ is a KKT point of the subproblem (3). So it suffices to show that case (c) occurs for some $\bar{k} \in \{0, 1, \ldots\}$, because then the algorithm terminates with $\hat{z}^k$ being a KKT point of the relaxed problem $\mathrm{R}(\hat{z}^k)$, and hence, by Theorem 2.1, a B-stationary point of MPEC (1) under the LICQ for $\mathrm{R}(\hat{z}^k)$.

For each $k \in \mathcal{K}$, we have from $t_k > 0$ and $z^{k+1} = \tilde{z}^k$ that

$$f(z^{k+1}) = f(\hat{z}^k + t_k d^k) \leq f(\hat{z}^k) + \sigma_2 t_k \nabla f(\hat{z}^k)^T d^k < f(\hat{z}^k) \leq f(z^k),$$

so the values $f(\hat{z}^k)$, $k \in \mathcal{K}$, are distinct. On the other hand, since the subproblem (3) is a quadratic program, a lemma from [9] shows that the set of values

$$\{\ f(z)\ :\ z \text{ is a KKT point of (3)}\ \}$$

is *finite*. Since $\epsilon_k = 0$ for all k and the number of index set pairs $(A, B) \in \mathcal{P}$ is finite, the number of distinct quadratic programs of the form (3) is finite. Therefore, there cannot be an infinite number of indices $k \in \mathcal{K}$, because each $\hat{z}^k$, $k \in \mathcal{K}$, is a KKT point of a quadratic program of the form (3). Hence there must be an index $k \in \mathcal{K}'$. This completes the proof. □

Theorem 4.1(a) concludes that $\epsilon_k \downarrow 0$ and $\nu_k \downarrow 0$. Thus, if instead of $\epsilon_{\mathrm{tol}} = \nu_{\mathrm{tol}} = 0$ we set $\epsilon_{\mathrm{tol}} > 0$, $\nu_{\mathrm{tol}} > 0$, then finite termination of the algorithm can be concluded. Notice that assumption A2 involves the ϵ used in the ϵ-active set algorithm. If an ϵ for which A2 holds is not known a priori, we can initialize ϵ arbitrarily in the algorithm, and whenever a d^k satisfying (9) does not exist in case (a) or (b) of Step

2,[1] we decrease ϵ and restart the algorithm at Step 0. The next lemma gives sufficient conditions for assumption A3 to hold.

LEMMA 4.2. *Suppose there exists $L > 0$ such that*

$$\tilde{\mathcal{F}}_{\epsilon_0} := \{z \in \mathcal{F}_{\epsilon_0} : f(z) \le f(z^0) + 2L\tau m\epsilon_0\}$$

is bounded and f is Lipschitz continuous on $Z := \tilde{\mathcal{F}}_{\epsilon_0} + 2\tau m\epsilon_0\mathcal{B}$ with constant L, where $\mathcal{B}$ denotes the unit sphere in $\Re^n$. Then $\{z^k\}$ and $\{\hat{z}^k\}$ lie in $\tilde{\mathcal{F}}_{\epsilon_0}$, and $\{\tilde{z}^k\}$ lies in Z. In particular, these sequences are all bounded.

Proof. An induction argument using (20), (21), and (22) yields, for each $k \in \mathcal{K} \cup \mathcal{K}'$,

$$z^k, \hat{z}^k \in \mathcal{F}_{\epsilon_k}, \quad f(\tilde{z}^k) \le f(\hat{z}^k) \le f(z^k) \le f(z^0) + 2L\tau m(\epsilon_0 - \epsilon_k),$$

and hence $z^k, \hat{z}^k \in \tilde{\mathcal{F}}_{\epsilon_0}$ and $\tilde{z}^k \in Z$. □

5. Some numerical experience. To gain some understanding of the practical performance of the ϵ-active set algorithm, we implemented in MATLAB a version of this algorithm for the two cases of quadratic f and nonquadratic f. We report below the implementation details and our numerical experience.

We first describe the case of quadratic f. In our implementation, we set $\epsilon_0 = \nu_0 = 0$ since f is quadratic. We also set $\sigma_1 = 0.5$, $\sigma_2 = 0.001$. To account for roundoff errors, any number below 10^{-12} in magnitude is treated as zero. In Step 1, we set $\hat{z}^k$ to be a KKT point of the quadratic program (3), with $(v_i^k)_{i=1}^m$ and $(w_i^k)_{i=1}^m$ being Lagrange multipliers associated with the first four sets of equations/inequalities. This is implemented by calling the quadratic program solver from the MATLAB Optimization Toolbox (Version 1.5.2), with z^k as the starting point. In Step 2, the index i_k is chosen to minimize $\min\{v_i^k, w_i^k\}$ among all indices $i \in A^k \cap B^k$. We choose d^k to minimize $\nabla f(\hat{z}^k)^T d$ among all d satisfying the constraints of (9), ignoring the first inequality. This is implemented by calling the linear program solver from the MATLAB Optimization Toolbox. We stop at Step 2 of iteration k when $v_i^k \ge 0, w_i^k \ge 0$ for all $i \in \hat{A}^k \cap \hat{B}^k$.

In our testing, we use the MATLAB program QPECgen (Version 1.1) of Jiang and Ralph [6] to generate test problems. This program generates random instances of mathematical programs with quadratic objective function and affine variational inequality constraints. We set the QPECgen parameters as in Table 8 of [6]. The problems thus generated are special cases of (1) with

$$f \text{ convex quadratic}, \qquad H_i(x, y) = y_i \quad \forall i, \qquad q = 0,$$

where $x \in \Re^{n-m}$ and $y \in \Re^m$. The dimensions n, m, p for the generated problems are shown in Table 1. The values of `second_deg` and `mono_M`, which are QPECgen parameters that control the degree of degeneracy (failure of lower-level strict complementarity) and the monotonicity of $[G_i(x, y)]_{i=1}^m$ in y, are also shown.

To generate the feasible starting point z^0, we considered solving

$$\begin{array}{lll} \text{minimize} & \sum_{i=1}^m G_i(z)H_i(z) & \\ \text{subject to} & G_i(z) \ge 0, & i = 1, \ldots, m, \\ & H_i(z) \ge 0, & i = 1, \ldots, m, \\ & g_j(z) \le 0, & j = 1, \ldots, p, \end{array}$$

[1] This can be detected from solving (9) as a linear program.

for a global optimal solution. However, the MATLAB quadratic program solver was unable to find a global optimal solution of this nonconvex quadratic program. Thus, we use a less direct scheme for generating z^0. We first modify the problem by negating the linear terms in the objective function. Then we apply the ϵ-active set algorithm to the modified problem, initialized with the recommended solution that is provided by QPECgen. This generates a feasible solution of (1) which we take to be z^0. (We cannot use the recommended solution as z^0, because it is already a B-stationary point.)

The performance of the algorithm on the test problems is reported in Table 1. As can be seen from Table 1, the algorithm terminated in a finite number of iterations on each problem. Moreover, on all except problems 9 and 10, the final z is verified to be a B-stationary point since the gradients of the active constraints were linearly independent. (In particular, the determinant of C^TC, where C is the matrix with columns being the gradients of the active constraints, is a large positive number. On problems 9 and 10, this determinant is below 10^{-12} in magnitude and hence is treated as zero.) Since f is convex, each B-stationary point is a local optimal solution of MPEC (1). For the first 8 problems, the final f-value agrees with those reported in Table 8 of [6] up to the fifth significant digit. The number of iterations increases significantly with n and m, especially on the two largest monotone problems. The reason for this is not well understood, but it does not appear to be related to degeneracy. For example, on the last problem of Table 1, the final solution has 3 degenerate indices, and yet the number of iterations is small relative to the problem size. The work at each iteration k varies, depending on the effort to solve the quadratic program (3) using the warm starting point z^k and the effort to solve the linear program derived from the system (9).

TABLE 1
Performance of the ϵ-active set algorithm on QPECgen problems.

(n, m, p)	`second_deg`	`mono_M`	**iter**[1]	**initial** f[2]	**final** f[3]	**deg**[4]
$(58, 50, 4)$	0	1	13	-45.682	-142.823	0
$(108, 100, 4)$	0	1	12	-620.389	-664.385	0
$(158, 150, 4)$	0	1	32	-475.081	-535.741	0
$(208, 200, 4)$	0	1	239	-55.876	-109.594	0
$(58, 50, 4)$	4	1	14	61.830	-41.876	4
$(108, 100, 4)$	4	1	12	-555.222	-599.936	4
$(158, 150, 4)$	4	1	32	-450.023	-536.443	4
$(208, 200, 4)$	4	1	179	19.916	-23.781	4
$(58, 50, 4)$	4	0	15	-78.398	-137.869	1
$(108, 100, 4)$	4	0	17	-165.875	-245.399	0
$(158, 150, 4)$	4	0	34	-349.095	-382.025	1
$(208, 200, 4)$	4	0	25	-39.048	-39.674	3

[1]The number of iterations until termination.
[2]This is $f(z^0)$.
[3]This is $f(z^k)$, where k indexes the last iteration.
[4]This is the cardinality of $A^k \cap B^k$, where k indexes the last iteration.

We next describe the case of nonquadratic f. Our implementation for this case is similar to that for the quadratic case but with the following differences: (i) we set

$$\epsilon_0 = \nu_0 = 10^{-6}, \quad \omega_1 = \omega_2 = 0.1, \quad \rho(\nu) = \nu,$$

and we use an initial estimate of $\eta = 10^{-4}$; (ii) in Step 1, we use a sequential quadratic programming (SQP) method to compute $\hat{z}^k$ as an approximate KKT point of the

nonlinear program (3), with $(v_i^k)_{i=1}^m$ and $(w_i^k)_{i=1}^m$ being Lagrange multipliers associated with the first four sets of equations/inequalities. More precisely, starting at $z = z^k$, we iteratively update z by solving a quadratic program obtained by replacing f in (3) with its quadratic approximation at z. Letting $z^\sharp$ denote the optimal solution of this quadratic program, we perform an inexact line search, using an Armijo rule analogous to (12), from z in the direction $z^\sharp - z$ to obtain the new z. We terminate the SQP method and set $\hat{z}^k = z$ when

$$\text{either} \quad \|z^\sharp - z\|_\infty < 10^{-12} \quad \text{or} \quad \nabla f(z)^T(z^\sharp - z) > -10^{-14}.$$

Each quadratic program is solved using the solver from the MATLAB Optimization Toolbox, with the current z as the starting point.

In our testing, we use a modification of the QPECgen problems from Table 1, whereby a cubic function $\sum_{i=1}^n (z_i)^3$ is added to the quadratic objective function. The starting point z^0 is chosen to be the same as in the quadratic case. Hence any change in the number of iterations from the quadratic case is due (mainly) to the change in the objective function. Notice that the added cubic function has the effect of pushing z towards zero, which on some problems enhances the degree of degeneracy upon termination.

TABLE 2
Performance of the ϵ-active set algorithm on modified QPECgen problems.

(n, m, p)	second_deg	mono_M	**iter**[1]	**initial** f[2]	**final** f[3]	**deg**[4]	**nq**[5]
(58, 50, 4)	0	1	18	−31.190	−135.466	0	62
(108, 100, 4)	0	1	10	−601.857	−645.175	0	39
(158, 150, 4)	0	1	32	−443.478	−504.749	0	122
(208, 200, 4)	0	1	233	−44.091	−98.463	0	443
(58, 50, 4)	4	1	19	78.157	−35.457	0	66
(108, 100, 4)	4	1	13	−536.606	−580.979	1	48
(158, 150, 4)	4	1	32	−416.060	−505.686	0	123
(208, 200, 4)	4	1	152	30.998	−9.030	8	248
(58, 50, 4)	4	0	17	−69.105	−168.218	1	55
(108, 100, 4)	4	0	16	−155.402	−236.762	0	48
(158, 150, 4)	4	0	21	−323.154	−346.061	3	43
(208, 200, 4)	4	0	16	−36.958	−37.405	5	37

[5]The number of quadratic programs solved until termination.

The performance of the algorithm on the test problems is reported in Table 2. From Table 2, it can be seen that the number of iterations is roughly comparable to that shown in Table 1 for the case of quadratic f. However, unlike the quadratic case, where one quadratic program needs to be solved per iteration, here two or more quadratic programs need to be solved per iteration on average. The work to solve each quadratic program varies, depending on the starting point. The efficiency of the algorithm can conceivably be improved by using a method more efficient than our simple SQP method to compute $\hat{z}^k$. Like the quadratic case, on all except problems 9 and 10, the final z is verified to be a B-stationary point since the gradients of the active constraints were linearly independent.

6. Conclusion. We have proposed an active set algorithm for solving mathematical programs with linear complementarity constraints and have established convergence to a B-stationary point of the problem under the uniform LICQ on the ϵ-feasible set. To the authors' knowledge, this is the first implementable algorithm that

enjoys global convergence to a B-stationary point without a nondegeneracy (lower-level strict complementarity) or upper-level strict complementarity assumption. We have also reported some numerical results that support the theoretical advantage of the algorithm.

Acknowledgments. The authors thank Stefan Scholtes, the two anonymous referees, and the Associate Editor, Daniel Ralph, for their helpful comments and suggestions on the original version of this paper.

REFERENCES

[1] F. Facchinei, H. Jiang, and L. Qi, *A smoothing method for mathematical programs with equilibrium constraints*, Math. Program., 85 (1999), pp. 107–134.

[2] M. Fukushima, Z.-Q. Luo, and J.-S. Pang, *A globally convergent sequential quadratic programming algorithm for mathematical programs with linear complementarity constraints*, Comput. Optim. Appl., 10 (1998), pp. 5–34.

[3] M. Fukushima and J.-S. Pang, *Convergence of a smoothing continuation method for mathematical programs with complementarity constraints*, in Ill-Posed Variational Problems and Regularization Techniques, Lecture Notes in Econom. and Math. Systems 477, M. Théra and R. Tichatschke, eds., Springer-Verlag, Berlin, Heidelberg, 1999, pp. 99–110.

[4] A. J. Hoffman, *On approximate solutions of systems of linear inequalities*, J. Res. Natl. Bur. Standards, 49 (1952), pp. 263–265.

[5] H. Jiang and D. Ralph, *Smooth SQP methods for mathematical programs with nonlinear complementarity constraints*, SIAM J. Optim., 10 (2000), pp. 779–808.

[6] H. Jiang and D. Ralph, *QPECgen, a MATLAB generator for mathematical programs with nonlinear complementarity constraints*, Comput. Optim. Appl., 13 (1999), pp. 25–59.

[7] Z.-Q. Luo, J.-S. Pang, and D. Ralph, *Mathematical Programs with Equilibrium Constraints*, Cambridge University Press, Cambridge, UK, 1996.

[8] Z.-Q. Luo, J.-S. Pang, and D. Ralph, *Piecewise sequential quadratic programming for mathematical programs with nonlinear complementarity constraints*, in Multilevel Optimization: Algorithms and Applications, A. Migdalas, P. M. Pardalos, and P. Värbrand, eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998, pp. 209–229.

[9] Z.-Q. Luo and P. Tseng, *Error bound and convergence analysis of matrix splitting algorithms for the affine variational inequality problem*, SIAM J. Optim., 2 (1992), pp. 43–54.

[10] J. V. Outrata, M. Kočvara, and J. Zowe, *Nonsmooth Approach to Optimization Problems with Equilibrium Constraints*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.

[11] J.-S. Pang and M. Fukushima, *Complementarity constraint qualifications and simplified B-stationarity conditions for mathematical programs with equilibrium constraints*, Comput. Optim. Appl., 13 (1999), pp. 111–136.

[12] S. Scholtes, *Convergence properties of a regularization scheme for mathematical programs with complementarity constraints*, SIAM J. Optim., 11 (2001), pp. 918–936.

[13] S. Scholtes and M. Stöhr, *Exact penalization of mathematical programs with equilibrium constraints*, SIAM J. Control Optim., 37 (1999), pp. 617–652.

[14] S. Scholtes and H. Scheel, *Mathematical programs with complementarity constraints: Stationarity, optimality, and sensitivity*, Math. Oper. Res., 25 (2000), pp. 1–22.

[15] M. Stöhr, *Nonsmooth Trust Region Methods and Their Applications to Mathematical Programs with Equilibrium Constraints*, Ph.D. thesis, Fakultät für Wirtschaftswissenschaften, Universität Fridericiana zu Karlsruhe, Karlsruhe, Germany, 1999.

[16] J. J. Ye, *Optimality conditions for optimization problems with complementarity constraints*, SIAM J. Optim., 9 (1999), pp. 374–387.

SIAM J. OPTIM.
Vol. 17, No. 4, pp. 1253–1257

AN IMPLEMENTABLE ACTIVE-SET ALGORITHM FOR COMPUTING A B-STATIONARY POINT OF A MATHEMATICAL PROGRAM WITH LINEAR COMPLEMENTARITY CONSTRAINTS: ERRATUM*

MASAO FUKUSHIMA† AND PAUL TSENG‡

Abstract. In [M. Fukushima and P. Tseng, *SIAM J. Optim.*, 12 (2002), pp. 724–739], an ϵ-active set algorithm was proposed for solving a mathematical program with a smooth objective function and linear inequality/complementarity constraints. It is asserted therein that, under a uniform LICQ on the ϵ-feasible set, this algorithm generates iterates whose cluster points are B-stationary points of the problem. However, the proof has a gap and shows only that each cluster point is an M-stationary point. We discuss this gap and show that B-stationarity can be achieved if the algorithm is modified and an additional error bound condition holds.

Key words. MPEC, B-stationary point, ϵ-active set, error bound

AMS subject classifications. 65K05, 90C30, 90C33

DOI. 10.1137/050642460

1. Introduction. In a recent paper by the authors [3], an ϵ-active set algorithm was proposed for solving the following mathematical program with equilibrium constraints (MPEC):

$$\begin{array}{lll} \text{minimize} & f(z) & \\ \text{subject to} & G_i(z) \geq 0, & i = 1, \ldots, m, \\ & H_i(z) \geq 0, & i = 1, \ldots, m, \\ & G_i(z)H_i(z) = 0, & i = 1, \ldots, m, \\ & g_j(z) \leq 0, & j = 1, \ldots, p, \\ & h_l(z) = 0, & l = 1, \ldots, q, \end{array} \tag{1}$$

where f is a real-valued continuously differentiable function on $\Re^n$ and G_i, H_i, g_j, h_l are real-valued *affine* functions on $\Re^n$. In Theorem 4.1(a) of [3], it is asserted that every cluster point of iterates generated by the algorithm is a B-stationary point of (1). However, the proof has a gap and shows only that every cluster point is an M-stationary point. We will discuss this gap and a modified algorithm that achieves B-stationarity under an additional error bound condition.

The gap occurs on [3, page 734] in the line "If $\nu_k \to 0$, then $|\mathcal{K}'| = \infty$, $\delta_k \to 0$, and the updating formula for ϵ_k would imply $\epsilon_k \to 0$, so any cluster point $\bar{z}$ of $\{\hat{z}^k\}_{k \in \mathcal{K}'}$ would be a KKT point of the *relaxed problem* $\mathrm{R}(\bar{z})$, which is a B-stationary point of

*Received by the editors October 11, 2005; accepted for publication (in revised form) June 7, 2006; published electronically January 22, 2007. This research is supported by Scientific Research Grant-in-Aid from the Ministry of Education, Science, Sports and Culture of Japan, and by National Science Foundation grant DMS-0511283.

http://www.siam.org/journals/siopt/17-4/64246.html

†Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan (fuku@amp.i.kyoto-u.ac.jp).

‡Department of Mathematics, University of Washington, Seattle, WA 98195 (tseng@math.washington.edu).

MPEC (1) under the uniform LICQ." In particular, we have for all $k \in \mathcal{K}'$ that

$$v_i^k \geq -\nu_k \quad \text{and} \quad w_i^k \geq -\nu_k \quad \forall\, i \in \hat{A}^k \cap \hat{B}^k, \tag{2}$$

where $\hat{A}^k$, $\hat{B}^k$ are given by [3, eq. (7)] and v_i^k, w_i^k are multipliers associated with $\hat{z}^k$ (see [3, eqs. (5), (6)]).[1] Thus, if a subsequence $\{\hat{z}^k\}_{k\in\mathcal{K}''}$ ($\mathcal{K}'' \subseteq \mathcal{K}'$) converges to some $\bar{z}$, then by further passing to a subsequence if necessary, we can assume that the index sets $\hat{A}^k$ and $\hat{B}^k$ are constant (i.e., $\hat{A}^k = \bar{A}$, $\hat{B}^k = \bar{B}$ for some $\bar{A}$, $\bar{B}$) for all $k \in \mathcal{K}''$. Since $\bar{z}$ satisfies the uniform LICQ, $\{(v_i^k)_{i\in\bar{A}}, (w_i^k)_{i\in\bar{B}}\}_{k\in\mathcal{K}''}$ also converges to some $(\bar{v}_i)_{i\in\bar{A}}, (\bar{w}_i)_{i\in\bar{B}}$.[2] By (2),

$$\bar{v}_i \geq 0 \quad \text{and} \quad \bar{w}_i \geq 0 \quad \forall\, i \in \bar{A} \cap \bar{B}.$$

This together with [3, eqs. (5), (6)] implies that $\bar{z}$ is an *M-stationary point* (see [4, 5] and (5) below). If in addition

$$\bar{A} \cap \bar{B} = A_0(\bar{z}) \cap B_0(\bar{z}), \tag{3}$$

then $\bar{z}$ is a B-stationary point of (1). In general, however, we can only assert that $\bar{A} \cap \bar{B} \subseteq A_0(\bar{z}) \cap B_0(\bar{z})$. This is the gap.

2. A modified ϵ-active set algorithm. We now describe a way, based on the active set identification approach of Facchinei, Fischer, and Kanzow [1], to modify the ϵ-active set algorithm so that (3) holds under an additional error bound condition. To simplify the notation, we will consider only the complementarity constraints, i.e., we assume $p = q = 0$ in (1). The general case can be treated analogously. The Lagrangian associated with (1) is

$$L(z, v, w) := f(z) + \sum_{i=1}^{m} (G_i(z)v_i + H_i(z)w_i).$$

We assume that there exists a computable continuous function $R : \Re^n \times \Re^m \times \Re^m \to [0, \infty)$ providing a local Hölder error bound at each M-stationary point $\bar{z}$ that is not B-stationary, i.e., there exist scalars $\tau > 0$, $\gamma > 0$, and $\delta > 0$ (depending on $\bar{z}$) such that

$$\|(z, v, w) - (\bar{z}, \bar{v}, \bar{w})\| \leq \tau R(z, v, w)^\gamma \quad \text{whenever } \|(z, v, w) - (\bar{z}, \bar{v}, \bar{w})\| \leq \delta, \tag{4}$$

where the multiplier vectors $\bar{v}, \bar{w}$ satisfy

$$\nabla_z L(\bar{z}, \bar{v}, \bar{w}) = 0, \ \left\{ \begin{array}{l} \bar{v}_i \perp G_i(\bar{z}) \geq 0 \\ \bar{w}_i \perp H_i(\bar{z}) \geq 0 \end{array} \right\}, \ G_i(\bar{z})H_i(\bar{z}) = 0, \ \left\{ \begin{array}{c} \bar{v}_i\bar{w}_i \geq 0 \\ \bar{v}_i \geq 0 \text{ or } \bar{w}_i \geq 0 \end{array} \right\} \ \forall i. \tag{5}$$

Here, $a \perp b$ means $ab = 0$. Due to uniform LICQ, $\bar{v}, \bar{w}$ are uniquely determined by $\bar{z}$. In fact, (5) characterizes M-stationarity for any $\bar{z} \in \Re^n$. We also assume that

$$R(\bar{z}, \bar{v}, \bar{w}) = 0 \quad \Longleftrightarrow \quad (\bar{z}, \bar{v}, \bar{w}) \text{ satisfies (5)}. \tag{6}$$

[1] Throughout, we use the same notation as [3].

[2] This follows from [3, eq. (6)], $\|r^k\|_1 \leq \delta_k \to 0$ (see [3, eq. (5)]), and the fact that if $b^k = C^k u^k$ for all k and $b^k \to b \in \Re^q$, $C^k \to C \in \Re^{q\times p}$ with C having linearly independent columns, then $u^k \to u \in \Re^p$ with u being the unique solution of $b = Cu$.

The "residual" function $R(z,v,w)$ can be constructed analogous to the NLP and NCP cases [1, 2]. In particular, consider

$$
\begin{aligned}
R(z,v,w) \quad &:= \quad \|\nabla_z L(z,v,w)\| + \sum_{i=1}^{m}\Big(|\min\{G_i(z),|v_i|\}| + |\min\{H_i(z),|w_i|\}| \\
&+ \quad |G_i(z)H_i(z)| + |\min\{0,v_i w_i\}| + |\min\{0,v_i\}\min\{0,w_i\}|\Big).
\end{aligned} \tag{7}
$$

Then, R is continuous and satisfies (6). Arguing as in the proof of Corollary 6.6.4 in [2], we have that the local error bound (4) holds if the M-stationary point $\bar{z}$ is isolated and f and ∇f are continuous and subanalytic (G and H, by being affine, are automatically continuous and subanalytic). A referee suggests that the assumption of $\bar{z}$ being isolated is benign when G and H are affine. In particular, it is readily shown that the M-stationary points of (1) are isolated if f is strictly convex on the null space of the active constraint gradients. Alternatively, it can be shown that the local error bound (4) holds with $\gamma = 1$ if a certain second-order sufficient condition holds at $\bar{z}$. This is a topic for further research.

Let $\theta : (0,\infty) \to (0,\infty)$ be any continuous nondecreasing function satisfying $\lim_{t\downarrow 0} t/\theta(t^\gamma) = 0$ for any $\gamma > 0$. An example is $\theta(t) = -C/\log(\min\{t, 0.9\})$ with $C > 0$. Using (4), (6) and following [1, 2], the function

$$\Theta(z,v,w) := \theta(R(z,v,w))$$

has the active set identification property that, for any M-stationary point $\bar{z}$ that is not B-stationary and corresponding multiplier vectors $\bar{v}, \bar{w}$, we have

$$\lim_{(z,v,w)\to(\bar{z},\bar{v},\bar{w})} \frac{G_i(z)}{\Theta(z,v,w)} = \begin{cases} 0 & \text{if } G_i(\bar{z}) = 0, \\ \infty & \text{if } G_i(\bar{z}) > 0, \end{cases}$$

and similarly with "G_i" replaced by "H_i."

Let us define

$$\bar{A}^k := \left\{ i \in \{1,\dots,m\} : \frac{G_i(\hat{z}^k)}{\Theta(\hat{z}^k, v^k, w^k)} \le 1 \right\},$$

$$\bar{B}^k := \left\{ i \in \{1,\dots,m\} : \frac{H_i(\hat{z}^k)}{\Theta(\hat{z}^k, v^k, w^k)} \le 1 \right\},$$

where the ith component of v^k is v_i^k if $i \in \hat{A}^k$ and is zero otherwise (and w^k is defined analogously). Since $(\hat{z}^k, v^k, w^k)$ satisfies [3, eqs. (4)–(6)], if (2) holds, then $R(\hat{z}^k, v^k, w^k)$ would tend to zero as $\hat{z}^k \to \bar{z}$ and $\epsilon_k, \delta_k, \nu_k$ tend to zero and, for $\hat{z}^k$ sufficiently near $\bar{z}$, we would have (v^k, w^k) sufficiently near $(\bar{v}, \bar{w})$ (due to [3, A2]) and

$$\bar{A}^k = A_0(\bar{z}), \quad \bar{B}^k = B_0(\bar{z}), \tag{8}$$

as well as

$$A_\epsilon(\hat{z}^k) \supseteq \bar{A}^k \supseteq \hat{A}^k, \qquad B_\epsilon(\hat{z}^k) \supseteq \bar{B}^k \supseteq \hat{B}^k, \tag{9}$$

where $\epsilon \ge 0$ is defined as in [3] (see page 727 therein).[3] Let

$$\bar{\epsilon}_k := \max\left\{ \epsilon_k, \max_{i\in\bar{A}^k} G_i(\hat{z}^k), \max_{i\in\bar{B}^k} H_i(\hat{z}^k) \right\}. \tag{10}$$

[3]The first containment in (9) holds whenever $\Theta(\hat{z}^k, v^k, w^k) \le \epsilon$, which in turn holds whenever $R(\hat{z}^k, v^k, w^k)$ is sufficiently small. By (8) and [3, eq. (7)], the second containment in (9) holds whenever $A_0(\bar{z}) \supseteq A_{\epsilon_k}(\hat{z}^k)$, which in turn holds whenever $\hat{z}^k$ is near $\bar{z}$ and ϵ_k is sufficiently small. The other two containments can be argued similarly.

Since $\bar{\epsilon}_k \geq \epsilon_k$, [3, eq. (4)] implies that $\hat{z}^k \in \mathcal{F}_{\bar{\epsilon}_k}[A^k, B^k]$ for all k. In fact, it can be seen that $\hat{z}^k$ remains an approximate KKT point of the subproblem [3, eq. (3)] (in the sense of [3, eqs. (4)–(6)]) when ϵ_k is replaced by $\bar{\epsilon}_k$ and $\hat{A}^k, \hat{B}^k$ are correspondingly replaced by $A_{\bar{\epsilon}_k}(\hat{z}^k), B_{\bar{\epsilon}_k}(\hat{z}^k)$. Thus, we can modify Step 2 of the ϵ-active set algorithm by possibly making this replacement when we are in case (c) and (9) holds.

THE MODIFIED $\boldsymbol{\epsilon}$-ACTIVE SET ALGORITHM FOR MPEC (1).

This is the same as the ϵ-active set algorithm in [3, pp. 730–731], except that when we are in case (c) in Step 2, we do the following: If

$$\text{(9) holds,} \quad \bar{A}^k \cap \bar{B}^k \neq \hat{A}^k \cap \hat{B}^k, \quad \bar{\epsilon}_k < \bar{\epsilon} \tag{11}$$

($\bar{\epsilon}$ is a threshold which initially can be any positive scalar below ϵ), then repeat Step 2 with ϵ_k replaced by $\bar{\epsilon}_k$ (and with $\hat{A}^k$, $\hat{B}^k$ redefined accordingly, i.e., they are replaced by $A_{\bar{\epsilon}_k}(\hat{z}^k), B_{\bar{\epsilon}_k}(\hat{z}^k)$ in Step 2, (9), (11)), and update $\bar{\epsilon} \leftarrow \bar{\epsilon}/2$. Otherwise, if $\epsilon_k \leq \epsilon_{\rm tol}$ and $\nu_k \leq \nu_{\rm tol}$, then terminate; otherwise, determine ν_{k+1} and $\tilde{z}^k$ by [3, eq. (14)], and proceed to Step 3.

If (11) holds, then $\epsilon_k < \bar{\epsilon}_k$,[4] which in turn implies $\bar{A}^k = A_{\bar{\epsilon}_k}(\hat{z}^k)$ and $\bar{B}^k = B_{\bar{\epsilon}_k}(\hat{z}^k)$.[5] Thus, when Step 2 is repeated, the second relation in (11) is violated.

THEOREM 2.1. *Under assumptions* [3, A1–A3], *the following results hold for the sequence* $\{(z^k, \hat{z}^k, \tilde{z}^k, \epsilon_k, \nu_k)\}$ *generated by the modified* ϵ*-active set algorithm, with* $\bar{\mathcal{K}} := \{k : \textit{at iteration } k, \textit{ Step 2 is repeated}\}$.

(a) *Suppose that each M-stationary point* $\bar{z}$ *of MPEC* (1) *that is not B-stationary satisfies* (4), *where* $(\bar{v}, \bar{w})$ *satisfies* (5) *and* R *satisfies* (6). *If* $\epsilon_0 > 0$, $\nu_0 > 0$, $\epsilon_{\rm tol} = \nu_{\rm tol} = 0$, f *is Lipschitz continuous with constant* L *on a set* Z *containing* $\{z^k\}$ *and* $\{\tilde{z}^k\}$, *and* $|\bar{\mathcal{K}}| < \infty$ *(respectively,* $|\bar{\mathcal{K}}| = \infty$*), then* $\epsilon_k \downarrow 0$, $\nu_k \downarrow 0$, *and every cluster point of* $\{\hat{z}^k\}$ *(respectively,* $\{\hat{z}^k\}_{k\in\bar{\mathcal{K}}}$*) is a B-stationary point of MPEC* (1).

(b) *If* $\epsilon_0 = \nu_0 = 0$ *and* f *is quadratic, then there exists a* $\bar{k} \in \{0, 1, \ldots\}$ *such that* $\hat{z}^{\bar{k}}$ *is a B-stationary point of MPEC* (1).

Proof. The first paragraph of the proof is identical to the proof of [3, Thm. 4.1], except we define $\mathcal{K} := \{k :$ We enter Step 3 from case (a) or (b) in Step 2 at iteration $k\}$ and $\mathcal{K}' := \{k :$ We enter Step 3 from case (c) in Step 2 at iteration $k\}$. The proof of (b) is identical to the proof of [3, Thm. 4.1(b)]. We prove (a) below.

(a) Suppose $\nu_k \to 0$. Then $|\mathcal{K}'| = \infty$, $\delta_k \to 0$, and the updating formulas for ϵ_k and $\bar{\epsilon}$ imply $\epsilon_k \to 0$, so any cluster point $\bar{z}$ of $\{\hat{z}^k\}_{k\in\mathcal{K}'}$ is an M-stationary point of MPEC (1). First, suppose $|\bar{\mathcal{K}}| < \infty$, so that $\bar{\epsilon} > 0$ is constant after a while. Let $\{\hat{z}^k\}_{k\in\mathcal{K}''}$ ($\mathcal{K}'' \subseteq \mathcal{K}'$) be any subsequence converging to $\bar{z}$. Since [3, eqs. (4)–(6)] and (2) hold for all $k \in \mathcal{K}''$, we have from [3, A2] and the same argument as in section 1 that $\{(v^k, w^k)\}_{k\in\mathcal{K}''} \to (\bar{v}, \bar{w})$ satisfying (5). By (6), $R(\bar{z}, \bar{v}, \bar{w}) = 0$. Since R is continuous, $\{R(\hat{z}^k, v^k, w^k)\}_{k\in\mathcal{K}''} \to 0$. If $\bar{z}$ is not B-stationary for (1), then the error bound (4) would hold and this would imply that (8) and (9) hold for all $k \in \mathcal{K}''$ sufficiently large. Moreover, $\{\bar{\epsilon}_k\}_{k\in\mathcal{K}''} \to 0$, so that $\bar{\epsilon}_k < \bar{\epsilon}$ for all $k \in \mathcal{K}''$ sufficiently

[4] If $\epsilon_k = \bar{\epsilon}_k$, then (10) and [3, eq. (7)] would imply $\bar{A}^k \subseteq \hat{A}^k$ and $\bar{B}^k \subseteq \hat{B}^k$, so (9) would yield $\bar{A}^k = \hat{A}^k$ and $\bar{B}^k = \hat{B}^k$, contradicting (11).

[5] Why? Since $\epsilon_k < \bar{\epsilon}_k$, we have from (10) and the definition of $\bar{A}^k$ and $\bar{B}^k$ that

$$\bar{\epsilon}_k = \max\left\{\max_{i\in\bar{A}^k} G_i(\hat{z}^k), \max_{i\in\bar{B}^k} H_i(\hat{z}^k)\right\} \leq \Theta(\hat{z}^k, v^k, w^k).$$

Thus, if $i \notin \bar{A}^k$, then $G_i(\hat{z}^k) > \Theta(\hat{z}^k, v^k, w^k) \geq \bar{\epsilon}_k$. By (10), if $i \in \bar{A}^k$, then $G_i(\hat{z}^k) \leq \bar{\epsilon}_k$. This shows that $\bar{A}^k = A_{\bar{\epsilon}_k}(\hat{z}^k)$. An analogous argument shows that $\bar{B}^k = B_{\bar{\epsilon}_k}(\hat{z}^k)$.

large. Thus, at each such iteration $k \in \mathcal{K}''$, we would have upon entering Step 3 that $\bar{A}^k \cap \bar{B}^k = \hat{A}^k \cap \hat{B}^k$ (since (11) must be violated). Then it would follow from (2) and (8) that $\bar{z}$ is a B-stationary point of (1), a contradiction. Second, suppose $|\bar{\mathcal{K}}| = \infty$. Then, as we discussed earlier, for each iteration $k \in \bar{\mathcal{K}}$, the second relation in (11) is violated upon entering Step 3, i.e., $\bar{A}^k \cap \bar{B}^k = \hat{A}^k \cap \hat{B}^k$. Then, an argument similar to the one above shows that every cluster point $\bar{z}$ of $\{\hat{z}^k\}_{k \in \bar{\mathcal{K}}}$ is a B-stationary point of (1).

Suppose instead $\nu_k \not\to 0$, so that $|\mathcal{K}'| < \infty$, $|\mathcal{K}| = \infty$, and $\nu = \lim_{k \to \infty} \nu_k > 0$. The remainder of the proof is identical to the proof of [3, Thm. 4.1(a)], except that, due to ϵ_k being replaced by $\bar{\epsilon}_k$ in Step 2 for all iterations $k \in \bar{\mathcal{K}}$, instead of [3, eq. (22)] we have

$$f(z^{k+1}) \le f(\tilde{z}^k) + 2L\tau m(\epsilon_k - \epsilon_{k+1} + \Delta_k) \quad \forall\, k,$$

where $\Delta_k := \bar{\epsilon}_k$ if $k \in \bar{\mathcal{K}}$ and $\Delta_k := 0$ otherwise. Since (11) holds at each iteration $k \in \bar{\mathcal{K}}$ and $\bar{\epsilon}$ is halved at each such iteration, it follows that $\sum_{k=0}^{\infty} \Delta_k = \sum_{k \in \bar{\mathcal{K}}} \bar{\epsilon}_k < \infty$. Then it can be argued similarly as in the proof of [3, Thm. 4.1(a)] that $\{f(z^k)\}$ converges and so on. □

We illustrate the assumptions of Theorem 2.1 with the following example of (1):

$$\text{minimize} \quad f(z) \quad \text{subject to} \quad z_1 \ge 0,\ z_2 \ge 0,\ z_1 z_2 = 0.$$

This example satisfies assumption [3, A2] for any $\epsilon \ge 0$. If $f(z) = (z_2)^p$ $(p \ge 1)$, then assumption [3, A1] also holds and each M-stationary point, which is of the form $(\bar{z}_1, 0)$ with $\bar{z}_1 \ge 0$, is B-stationary. If $f(z) = z_1^4 + z_2^2 - z_2$, then assumptions [3, A1, A3] also hold and the M-stationary points, $(0,0)$ and $(0, \frac{1}{2})$, are isolated with $(0, \frac{1}{2})$ B-stationary. For R given by (7), the error bound (4) holds at $(0,0)$. However, if $f(z) = z_2^2 - z_2$, then the M-stationary point $\bar{z} = (0,0)$, with multipliers $\bar{v} = 0, \bar{w} = -1$, is not B-stationary and is not isolated. Moreover, for any continuous R satisfying (6), the error bound (4) does not hold at $(0,0)$. This is because, for any fixed $\delta > 0$, $(\delta, 0)$ is M-stationary with multipliers $v = 0, w = -1$, so $R((\delta, x_2), 0, -1) \to R(\delta, 0), 0, -1) = 0$ as $x_2 \to 0$. But $\|((\delta, x_2), 0, -1) - ((0,0), 0, -1)\| \to \delta$ as $x_2 \to 0$.

Acknowledgments. The authors thank Lifeng Chen for notifying them of the gap in the proof of [3, Thm. 4.1]. They also thank two referees for their helpful comments.

REFERENCES

[1] F. Facchinei, A. Fischer, and C. Kanzow, *On the accurate identification of active constraints*, SIAM J. Optim., 9 (1998), pp. 14–32.

[2] F. Facchinei and J.-S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems*, Vol. II, Springer-Verlag, New York, 2003.

[3] M. Fukushima and P. Tseng, *An implementable active-set algorithm for computing a B-stationary point of a mathematical program with linear complementarity constraints*, SIAM J. Optim., 12 (2002), pp. 724–739.

[4] J. Outrata, *Optimality conditions for a class of mathematical programs with equilibrium constraints*, Math. Oper. Res., 24 (1999), pp. 627–644.

[5] H. Scheel and S. Scholtes, *Mathematical programs with complementarity constraints: Stationarity, optimality, and sensitivity*, Math. Oper. Res., 25 (2000), pp. 1–22.

SIAM J. OPTIM.
Vol. 12, No. 3, pp. 740–755

ROBUST FILTERING VIA SEMIDEFINITE PROGRAMMING WITH APPLICATIONS TO TARGET TRACKING*

LINGJIE LI†, ZHI-QUAN LUO†, TIMOTHY N. DAVIDSON†, K. MAX WONG†, AND ELOI BOSSÉ‡

Abstract. In this paper we propose a novel finite-horizon, discrete-time, time-varying filtering method based on the robust semidefinite programming (SDP) technique. The proposed method provides robust performance in the presence of norm-bounded parameter uncertainties in the system model. The robust performance of the proposed method is achieved by minimizing an upper bound on the worst-case variance of the estimation error for all admissible systems. Our method is recursive and computationally efficient. In our simulations, the new method provides superior performance to some of the existing robust filtering approaches. In particular, when applied to the problem of target tracking, the new method has led to a significant improvement in tracking performance. Our work shows that the robust SDP technique and the interior point algorithms can bring substantial benefits to practically important engineering problems.

Key words. robust filtering, Kalman filtering, semidefinite programming, target tracking

AMS subject classifications. 90C90, 90C22, 90C51

PII. S1052623499358586

1. Introduction.

Consider the following classical discrete-time linear state-space model:

$$\left\{ \begin{array}{rcll} \underline{x}_{i+1} & = & \mathbf{F}_i \underline{x}_i + \mathbf{G}_i \underline{u}_i, & \underline{x}_0 \text{ given}, \\ \underline{y}_i & = & \mathbf{H}_i \underline{x}_i + \underline{v}_i, & i \geq 0, \end{array} \right. \tag{1.1}$$

where $\mathbf{F}_i \in \mathcal{R}^{n\times n}$, $\mathbf{G}_i \in \mathcal{R}^{n\times m}$, and $\mathbf{H}_i \in \mathcal{R}^{p\times n}$ are known matrices which describe the dynamic system, and $\underline{x}_i$ describes the state of the system at time i, while $\underline{u}_i$ and $\underline{v}_i$ denote the process and measurement noise terms, respectively. In many linear filtering applications, we are faced with the problem of estimating the states of the dynamic system (1.1) from the noisy measurements $\underline{y}_i$ (see [6, 11, 8]). A popular solution to this problem is given by the Kalman filter [6, 11, 8] which, under some standard assumptions on the statistics of the noise sources and initial state, minimizes the mean squared estimation error (MSE). The MSE is the trace of $\mathcal{E}\{(\underline{x}_i - \hat{\underline{x}}_i)(\underline{x}_i - \hat{\underline{x}}_i)^T\}$, where $\mathcal{E}$ denotes the statistical expectation and $\hat{\underline{x}}_i$ denotes the estimate of $\underline{x}_i$ at time i. Moreover, the Kalman filter is recursive and computationally efficient. In its "innovations form," the Kalman filter is given by

$$\hat{\underline{x}}_{i+1} = \mathbf{F}_i \hat{\underline{x}}_i + \mathbf{K}_{K,i}\big(\underline{y}_i - \mathbf{H}_i \hat{\underline{x}}_i\big), \quad \hat{\underline{x}}_0 = 0, \tag{1.2}$$

*Received by the editors June 23, 1999; accepted for publication (in revised form) June 13, 2001; published electronically February 8, 2002. This research was supported by a grant from the Defense Research Establishment of Canada at Valcartier, QC, Canada.
http://www.siam.org/journals/siopt/12-3/35858.html

†Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON, L8S 4L7, Canada (lli@redlinecommunications.com, luozq@mcmaster.ca, davidson@sunrise.crl.mcmaster.ca, wong@mail.ece.mcmaster.ca). The second author is also supported by the Canada Research Chair program.

‡Defense Research Establishment Valcartier, Decision Support Technologies Section, 2459 PIE XI Nord, B.P. 8800, Courcelette, QC, G0A 1R0, Canada (eloi.bosse@drev.dnd.ca).

where the so-called Kalman gain matrix $\mathbf{K}_{K,i}$ can be computed via the following (analytic) recursion:

$$\mathbf{K}_{K,i} = \mathbf{F}_i \mathbf{P}_i \mathbf{H}_i^T \left(\mathbf{R}_i + \mathbf{H}_i \mathbf{P}_i \mathbf{H}_i^T \right)^{-1},$$

$$\mathbf{P}_{i+1} = (\mathbf{F}_i - \mathbf{K}_{K,i}\mathbf{H}_i)\, \mathbf{P}_i\, (\mathbf{F}_i - \mathbf{K}_{K,i}\mathbf{H}_i)^T + [\, \mathbf{G}_i \quad -\mathbf{K}_{K,i}\,] \begin{bmatrix} \mathbf{Q}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_i \end{bmatrix} \begin{bmatrix} \mathbf{G}_i^T \\ -\mathbf{K}_{K,i}^T \end{bmatrix},$$

where $\mathbf{Q}_i = \mathcal{E}\{\underline{u}_i \underline{u}_i^T\}$ and $\mathbf{R}_i = \mathcal{E}\{\underline{v}_i \underline{v}_i^T\}$ are the noise covariance matrices. (The statistical assumptions made here are stated in section 2.) The matrix $\mathbf{P}_i$ in the recursion is the error covariance matrix $\mathcal{E}\{(\underline{x}_i - \underline{\hat{x}}_i)(\underline{x}_i - \underline{\hat{x}}_i)^T\}$. However, one drawback of the Kalman filter is that it requires precise knowledge of the system matrices $\mathbf{F}_i$, $\mathbf{G}_i$, and $\mathbf{H}_i$ and noise covariances $\mathbf{Q}_i$ and $\mathbf{R}_i$, because even a small deviation from the "nominal" values of these matrices can induce substantial performance loss in the Kalman filter. As a result, the Kalman filter can be ineffective in practice especially when we are faced with imprecise knowledge of the dynamic system mode or, in other words, when the matrices $\mathbf{F}_i$, $\mathbf{G}_i$, and $\mathbf{H}_i$ are known only approximately. This sensitivity of the Kalman filter has led researchers to tackle *robust filtering* problems, in which the objective is to design estimators which provide acceptable performance in the presence of uncertainties in the models of the dynamic system and the noise.

One approach to robust filtering is that of H^∞ filtering (see [5] and references therein). In that approach no statistical model of the disturbances $\underline{u}_i$ and $\underline{v}_i$ is employed; they are merely assumed to have finite energy. The idea is to obtain an estimator which minimizes (or, in the suboptimal case, bounds) the maximum energy gain from the disturbances to the estimation errors. This modelling paradigm also allows us to incorporate unstructured uncertainties in the dynamic system model (1.1) (see, for example, [4, 17]). An advantage of the H^∞ approach is that the solution closely resembles the Kalman filter and can be efficiently implemented. Therefore, in applications in which statistical knowledge of the disturbances and information regarding the structure of the modelling uncertainties are difficult to acquire, H^∞ filters are an appropriate choice. Unfortunately, when the system model and the noise processes are known quite accurately, the Kalman filter may actually perform substantially better than the H^∞ filter. This is because the uncertainty model for the H^∞ filter is unstructured, and hence the H^∞ filter may be attempting to provide robustness to disturbances and modelling errors which rarely, or never, occur, at the expense of filter performance in the presence of more likely disturbances and modelling errors. In many applications, including target tracking, we have some knowledge of the structure of the uncertainties in the system model and partial knowledge of disturbance statistics. It is natural to expect that careful incorporation of this knowledge into the estimator will lead to appreciable improvement in estimator performance. A major challenge is to determine whether this can be done in a computationally efficient manner. From recent work in the control field, it appears that determining filters which provide optimal robustness to highly structured uncertainties can be computationally expensive [1].

An alternative to the Kalman and H^∞ filtering methods is to find a "robust Kalman filter" which minimizes (an upper bound on) the variance of the estimation error in the presence of a system model with norm-bounded structured parametric uncertainty and bounded uncertainty in the noise statistics. Models of this type are common in control theory (e.g., [7] and references therein) and are particularly appropriate in the context of target tracking. Previous approaches to this problem, with no uncertainty in the noise statistics, have been based on analytic recursions on

some performance bounds [13, 15]. Note that robust H^∞ designs which bound the worst-case error energy gain in the presence of the same system model uncertainties are also available [9, 16].

In this paper we derive a new robust filtering algorithm using the recently developed robust semidefinite programming (SDP) technique [2]. The new method is recursive in the sense that the subproblem solved at each step depends on the solution at the previous step, and is computationally efficient since each subproblem is a semidefinite program of a fixed size which can be efficiently solved using an interior point algorithm. We demonstrate the performance of the novel algorithm in a standard benchmark example and in a target-tracking example, and show that it can provide superior performance to the existing approaches to this particular problem [13, 15], and to the Kalman and H^∞ approaches. Our work shows that the robust SDP technique and the interior point algorithms [12, 14] can bring substantial benefits to a practically important engineering problem.

The paper is organized as follows. In section 2 the robust state estimator problem is introduced. Then, in section 3, this problem is formulated as convex optimization and solved in polynomial-time using the recent robust SDP technique. In section 4, simulation results are presented and, in section 5, some concluding remarks are given.

Throughout this paper, for a square matrix $\mathbf{X}$, the notation $\mathbf{X} \geq 0$ (resp., $\mathbf{X} \leq 0$) means $\mathbf{X}$ is symmetric and positive semidefinite (resp., negative semidefinite).

2. Problem formulation. Consider the following time-varying, discrete-time, uncertain linear state-space model:

$$\left\{ \begin{array}{rcll} \underline{x}_{i+1} & = & [\mathbf{F}_i + \Delta\mathbf{F}_i]\,\underline{x}_i + \mathbf{G}_i\underline{u}_i, & \underline{x}_0, \\ \underline{y}_i & = & [\mathbf{H}_i + \Delta\mathbf{H}_i]\,\underline{x}_i + \underline{v}_i, & i \geq 0, \end{array} \right. \tag{2.1}$$

where $\mathbf{F}_i \in \mathcal{R}^{n\times n}$, $\mathbf{G}_i \in \mathcal{R}^{n\times m}$, and $\mathbf{H}_i \in \mathcal{R}^{p\times n}$ are known matrices which describe the nominal system. The matrices $\Delta\mathbf{F}_i$ and $\Delta\mathbf{H}_i$ represent the parameter uncertainties in the dynamic model. They are assumed to have the following structure:

$$\begin{bmatrix} \Delta\mathbf{F}_i \\ \Delta\mathbf{H}_i \end{bmatrix} = \begin{bmatrix} \mathbf{C}_{1,i} \\ \mathbf{C}_{2,i} \end{bmatrix} \mathbf{Z}_i\mathbf{E}_i \quad \text{with} \quad \mathbf{Z}_i^T\mathbf{Z}_i \leq \mathbf{I}, \tag{2.2}$$

where $\mathbf{C}_{1,i} \in \mathcal{R}^{n\times r}$, $\mathbf{C}_{2,i} \in \mathcal{R}^{p\times r}$, and $\mathbf{E}_i \in \mathcal{R}^{t\times n}$ are known matrices. We remark that the above model (2.2) of uncertainties has been used extensively in the robust control literature (e.g., [7] and references therein). The process noise $\{\underline{u}_i\}$, the measurement noise $\{\underline{v}_i\}$, and the initial state $\underline{x}_0$ in (2.1) are all assumed to be random. These random variables have known mean values, which we can take to be zero without loss of generality, and partially known covariances, as follows:

$$\mathcal{E}\left\{ \begin{bmatrix} \underline{u}_i \\ \underline{v}_i \\ \underline{x}_0 \end{bmatrix} \begin{bmatrix} \underline{u}_j \\ \underline{v}_j \\ \underline{x}_0 \end{bmatrix}^T \right\} = \begin{bmatrix} \mathbf{Q}_i\delta_{ij} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_i\delta_{ij} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{\Pi_0} \end{bmatrix}, \tag{2.3}$$

where δ_{ij} denotes the Kronecker delta function that is equal to unity for $i = j$ and zero elsewhere, $\mathbf{Q}_i = \bar{\mathbf{Q}}_i + \Delta\mathbf{Q}_i$, and $\mathbf{R}_i = \bar{\mathbf{R}}_i + \Delta\mathbf{R}_i$. The matrices $\bar{\mathbf{Q}}_i \in \mathcal{R}^{m\times m}$, $\bar{\mathbf{R}}_i \in \mathcal{R}^{p\times p}$, and $\mathbf{\Pi}_0 \in \mathcal{R}^{n\times n}$ are assumed to be known and describe the nominal second-order statistics of the noise and the initial state. The matrices $\Delta\mathbf{Q}_i$ and $\Delta\mathbf{R}_i$ represent the uncertainties in the noise statistics and satisfy the following bounds:

$$-\epsilon\mathbf{I} \leq \Delta\mathbf{Q}_i \leq \epsilon\mathbf{I}, \quad -\epsilon\mathbf{I} \leq \Delta\mathbf{R}_i \leq \epsilon\mathbf{I}. \tag{2.4}$$

Notice that when there is no uncertainty in the system model (2.1), namely $\epsilon = 0$ and $\mathbf{E}_i = 0$, then we recover the standard linear time-varying state-space model (1.1).

Let us use $\Theta_i = \{\Delta\mathbf{Q}_i, \Delta\mathbf{R}_i, \mathbf{Z}_i\}$ to denote the uncertainty variable at stage i and define the uncertainty region at stage i as

$$\Omega_i = \{\ \Theta_i : \Theta_i \text{ satisfies (2.2) and (2.4)}\ \}.$$

The problem is to estimate the state-sequence $\{\underline{x}_i,\ i \geq 0\}$, or some linear combination of this sequence $\{\underline{s}_i = \mathbf{L}_i\underline{x}_i, i \geq 0\}$, where $\mathbf{L}_i$ is a known matrix, from the corrupted measurements. The goal of the robust filter is to provide a uniformly small estimation error for any process and measurement noise satisfying (2.3) and (2.4) and for all admissible modelling uncertainties satisfying (2.2). These a priori bounds on the uncertainties represent the designer's partial knowledge of the noise statistics and system model. They are to be incorporated into the problem formulation to guarantee robust performance.

To formulate the robust filtering problem, consider the following form of state estimator:

$$\underline{\hat{x}}_{i+1} = \mathbf{A}_i\underline{\hat{x}}_i + \mathbf{K}_i(\underline{y}_i - \mathbf{H}_i\underline{\hat{x}}_i), \quad \underline{\hat{x}}_0 = 0, \tag{2.5}$$

where $\mathbf{A}_i$, $\mathbf{K}_i$ are filtering matrices to be determined, and $\underline{\hat{x}}_i$ denotes the estimate of the state $\underline{x}_i$. The above estimator is written in an innovation form that is similar to the structure of the Kalman filter given in (1.2). Notice that we use the nominal innovation $(\underline{y}_i - \mathbf{H}_i\underline{\hat{x}}_i)$, even though $\Delta\mathbf{H}_i$ may be nonzero. This structure is used for convenience, but it is general enough to generate all the full-order estimators, since $\mathbf{A}_i$ and $\mathbf{K}_i$ are free parameters. The goal of a robust filtering algorithm is to choose these free parameters to minimize (a function of) the estimation error covariance $\mathcal{E}\{(\underline{x}_i - \underline{\hat{x}}_i)(\underline{x}_i - \underline{\hat{x}}_i)^T\}$.

To express that goal precisely, we consider the following augmented system, which represents the cascade of the system in (2.1) and the estimator in (2.5):

$$\underline{\bar{x}}_{i+1} = \left[\bar{\mathbf{F}}_i + \bar{\mathbf{C}}_i\mathbf{Z}_i\bar{\mathbf{E}}_i\right]\underline{\bar{x}}_i + \bar{\mathbf{G}}_i\underline{\bar{u}}_i, \tag{2.6}$$

where

$$\begin{cases} \underline{\bar{x}}_i &= \begin{bmatrix} \underline{x}_i \\ \underline{\hat{x}}_i \end{bmatrix}, \quad \underline{\bar{u}}_i = \begin{bmatrix} \underline{u}_i \\ \underline{v}_i \end{bmatrix}, \\ \bar{\mathbf{F}}_i &= \begin{bmatrix} \mathbf{F}_i & \mathbf{0} \\ \mathbf{K}_i\mathbf{H}_i & \mathbf{A}_i - \mathbf{K}_i\mathbf{H}_i \end{bmatrix}, \quad \bar{\mathbf{G}}_i = \begin{bmatrix} \mathbf{G}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_i \end{bmatrix}, \\ \bar{\mathbf{C}}_i &= \begin{bmatrix} \mathbf{C}_{1,i} \\ \mathbf{K}_i\mathbf{C}_{2,i} \end{bmatrix}, \quad \bar{\mathbf{E}}_i = [\mathbf{E}_i \ \ \mathbf{0}]. \end{cases}$$

Note that the state vector of the cascade, $\underline{\bar{x}}_i$, contains both $\underline{x}_i$ (the states of the model) and the estimates $\underline{\hat{x}}_i$, and hence the dimension of the state vector is doubled. The Lyapunov equation that governs the evolution of the covariance matrix $\mathbf{\Sigma}_i = \mathcal{E}\{\underline{\bar{x}}_i\underline{\bar{x}}_i^T\}$ can be written as

$$\mathbf{\Sigma}_{i+1} = \left[\bar{\mathbf{F}}_i + \bar{\mathbf{C}}_i\mathbf{Z}_i\bar{\mathbf{E}}_i\right]\mathbf{\Sigma}_i\left[\bar{\mathbf{F}}_i + \bar{\mathbf{C}}_i\mathbf{Z}_i\bar{\mathbf{E}}_i\right]^T + \bar{\mathbf{G}}_i\mathbf{W}_i\bar{\mathbf{G}}_i^T, \tag{2.7}$$

where $\mathbf{W}_i = \text{blockdiag}\,(\mathbf{Q}_i, \mathbf{R}_i)$. The error covariance $\mathbf{P}_{i+1}$ can be obtained from (2.7) by premultiplying $[\mathbf{I} \quad -\mathbf{I}]$ and postmultiplying $[\mathbf{I} \quad -\mathbf{I}]^T$; i.e.,

$$\mathbf{P}_{i+1} = \hat{\mathbf{F}}_i \boldsymbol{\Sigma}_i \hat{\mathbf{F}}_i^T + \mathbf{G}_i \mathbf{Q}_i \mathbf{G}_i^T + \mathbf{K}_i \mathbf{R}_i \mathbf{K}_i^T, \tag{2.8}$$

where

$$\hat{\mathbf{F}}_i = \left[\begin{array}{cc} (\mathbf{F}_i + \mathbf{C}_{1,i}\mathbf{Z}_i\mathbf{E}_i - \mathbf{K}_i\mathbf{H}_i - \mathbf{K}_i\mathbf{C}_{2,i}\mathbf{Z}_i\mathbf{E}_i) & (\mathbf{K}_i\mathbf{H}_i - \mathbf{A}_i) \end{array} \right].$$

Now the finite-horizon robust state estimator problem can be stated as follows.

PROBLEM. *At each stage i, choose the filtering matrices $\{\mathbf{A}_j\}_{j=0}^i$ and $\{\mathbf{K}_j\}_{j=0}^i$ so as to minimize the worst-case weighted error covariance matrix $\mathbf{DP}_{i+1}$; i.e.,*

$$\min_{\substack{\mathbf{K}_j, \mathbf{A}_j \\ \forall j \le i}} \quad \max_{\substack{\Theta_j \in \Omega_j \\ \forall j \le i}} \text{Tr}\,(\mathbf{DP}_{i+1}), \tag{2.9}$$

or equivalently,

$$\min_{\substack{\mathbf{K}_j, \mathbf{A}_j \\ \forall j \le i}} \quad \max_{\substack{\Theta_j \in \Omega_j \\ \forall j \le i}} \text{Tr}\left(\mathbf{D}\,[\mathbf{I} \quad -\mathbf{I}]\,\boldsymbol{\Sigma}_{i+1}\,[\mathbf{I} \quad -\mathbf{I}]^T\right), \tag{2.10}$$

where $\text{Tr}\,(\cdot)$ denotes the trace of a matrix $(\cdot)$ and $\mathbf{D} \in \mathcal{R}^{n \times n}$ is a positive semidefinite weighting matrix.

We have stated the robust state estimation problem in a rather general weighted form which includes many special cases. If we wish to estimate $\{\underline{x}_i\}$, choosing $\mathbf{D} = \mathbf{I}$ will suffice, whereas to estimate $\{\underline{s}_i = \mathbf{L}_i \underline{x}_i\}$, choosing $\mathbf{D} = \mathbf{L}_i \mathbf{L}_i^T$ will suffice. We can also weight the estimation accuracy of the states as desired, or add additional terms to $\mathbf{D}$, as long as it remains positive semidefinite. As we will observe later in section 4, adding additional terms to $\mathbf{D}$ may improve the numerical stability of the finite-horizon filtering solutions.

The above minimax formulation is intended to incorporate robustness into the filter solution. In particular, $\text{Tr}\,(\mathbf{DP}_{i+1})$, as recursively defined by (2.8), depends on all the uncertainties $\Theta_0, \ldots, \Theta_i$ as well as on the filtering matrices $\mathbf{K}_0, \mathbf{A}_0, \ldots, \mathbf{K}_i, \mathbf{A}_i$. The maximum weighted trace of $\mathbf{P}_{i+1}$,

$$\max_{\substack{\Theta_j \in \Omega_j \\ \forall j \le i}} \text{Tr}\,(\mathbf{DP}_{i+1}),$$

represents the worst-case weighted error covariance when subject to the prescribed uncertainties. Therefore, the goal of robust filter design is to select the filtering matrices so that the worst-case weighted error covariance is minimized.

As given by (2.9) or (2.10), the robust filter design problem is nonlinear and nonsmooth and hence is computationally difficult. Furthermore, the problem apparently lacks convexity, which is essential in the development of computationally efficient algorithms. A further difficulty with the formulation (2.9) or (2.10) is that it is nonrecursive, in the sense that the problem dimension increases linearly in i. This nonrecursive feature makes it necessary to solve from scratch for the filtering matrices $\mathbf{K}_0, \mathbf{A}_0, \ldots, \mathbf{K}_i, \mathbf{A}_i$ at each stage i, which is clearly undesirable and impractical.

In practice, we typically fix $\mathbf{K}_0, \mathbf{A}_0, \ldots, \mathbf{K}_{i-1}, \mathbf{A}_{i-1}$ at stage i and solve only for $\mathbf{K}_i, \mathbf{A}_i$. However, such simplification only partially fixes the problem since the uncertainties $\Theta_0, \ldots, \Theta_i$ still enter into the maximization of $\text{Tr}\,(\mathbf{DP}_{i+1})$, indicating that the problem dimension still increases linearly with i. Our objective is to reformulate

problem (2.9) in a recursive way such that at each stage i we have only to determine $\mathbf{K}_i, \mathbf{A}_i$ by solving a subproblem with a fixed dimension (i.e., independent of i).

To reformulate problem (2.9), we consider a sequence of matrices

$$\{\mathbf{\Gamma}_{i+1}(\mathbf{K}_i, \mathbf{A}_i) : \ i = 1, 2, \ldots\},$$

which are *not* dependent on the uncertainties $\{\Theta_i : \ i = 1, 2, \ldots\}$. These matrices will serve as upper bounds for the covariance matrices $\{\mathbf{\Sigma}_{i+1} : \ i = 1, 2, \ldots\}$ which *are* dependent on the uncertainty vectors $\{\Theta_i : \ i = 1, 2, \ldots\}$, as well as on $\mathbf{K}_i$ and $\mathbf{A}_i$. In particular, we will have

$$\mathbf{\Gamma}_{i+1}(\mathbf{K}_i, \mathbf{A}_i) \geq \mathbf{\Sigma}_{i+1} \qquad \forall \Theta_i \in \Omega_i, \quad i = 1, 2, \ldots. \tag{2.11}$$

There are, of course, many choices for an upper bound $\mathbf{\Gamma}_{i+1}(\mathbf{K}_i, \mathbf{A}_i)$ that will satisfy (2.11). Our objective should be to choose the one which, together with some $\mathbf{K}_i$ and $\mathbf{A}_i$, will yield the minimum weighted error covariance $\mathbf{DP}_{i+1}$. By the relation

$$\mathbf{P}_{i+1} = \begin{bmatrix} \mathbf{I} & -\mathbf{I} \end{bmatrix} \mathbf{\Sigma}_{i+1} \begin{bmatrix} \mathbf{I} & -\mathbf{I} \end{bmatrix}^T,$$

we see that an upper bound on $\mathbf{\Sigma}_{i+1}$ naturally leads to an upper bound on $\mathbf{P}_{i+1}$. Thus we can approximately minimize $\mathbf{DP}_{i+1}$ by minimizing the trace of the matrix

$$\mathbf{D} \begin{bmatrix} \mathbf{I} & -\mathbf{I} \end{bmatrix} \mathbf{\Gamma}_{i+1}(\mathbf{K}_i, \mathbf{A}_i) \begin{bmatrix} \mathbf{I} & -\mathbf{I} \end{bmatrix}^T,$$

which is an upper bound of $\mathbf{DP}_{i+1}$. In particular, we choose $\mathbf{\Gamma}_{i+1}$, $\mathbf{K}_i$, and $\mathbf{A}_i$ to

$$\begin{aligned} &\text{minimize} \quad \operatorname{Tr}\left(\mathbf{D} \begin{bmatrix} \mathbf{I} & -\mathbf{I} \end{bmatrix} \mathbf{\Gamma}_{i+1} \begin{bmatrix} \mathbf{I} & -\mathbf{I} \end{bmatrix}^T\right) \\ &\text{subject to} \quad \mathbf{\Gamma}_{i+1},\ \mathbf{K}_i,\ \mathbf{A}_i \text{ satisfying (2.11).} \end{aligned} \tag{2.12}$$

The optimization problem (2.12) involves the constraint (2.11), which involves all of the uncertainty vectors $\{\Theta_i : \ i = 1, 2, \ldots\}$ and $\{\mathbf{K}_i, \mathbf{A}_i : \ i = 1, 2, \ldots\}$, thus making the amount of computation increase with i. To resolve this issue of dimensionality increase, we shall define the constraint recursively as follows. Specifically, let $b > 0$ be a chosen scalar bound and let $\bar{\mathbf{\Sigma}}_0 = \mathbf{\Sigma}_0$. For $i \geq 0$, suppose $\bar{\mathbf{\Sigma}}_i$, an upper bound on $\mathbf{\Sigma}_i$, has been computed and is already available. Consider the following minimization problem in the matrix variables $\{\mathbf{\Gamma}_{i+1}, \mathbf{K}_i, \mathbf{A}_i\}$:

$$\begin{aligned} &\text{minimize} \quad \operatorname{Tr}\left(\mathbf{D} \begin{bmatrix} \mathbf{I} & -\mathbf{I} \end{bmatrix} \mathbf{\Gamma}_{i+1} \begin{bmatrix} \mathbf{I} & -\mathbf{I} \end{bmatrix}^T\right) \\ &\text{subject to} \quad \mathbf{\Gamma}_{i+1} \geq \left[\bar{\mathbf{F}}_i + \bar{\mathbf{C}}_i \mathbf{Z}_i \bar{\mathbf{E}}_i\right] \bar{\mathbf{\Sigma}}_i \left[\bar{\mathbf{F}}_i + \bar{\mathbf{C}}_i \mathbf{Z}_i \bar{\mathbf{E}}_i\right]^T + \bar{\mathbf{G}}_i \mathbf{W}_i \bar{\mathbf{G}}_i^T \qquad \forall \Theta_i \in \Omega_i, \\ &\qquad\qquad \operatorname{Tr}(\mathbf{\Gamma}_{i+1}) \leq b. \end{aligned} \tag{2.13}$$

We choose $\bar{\mathbf{\Sigma}}_{i+1}$ to be the optimal value of $\mathbf{\Gamma}_{i+1}$ in (2.13). Therefore our reformulation of (2.9) can now be stated as the following.

REFORMULATION OF THE ROBUST FILTERING PROBLEM. *Let* $\bar{\mathbf{\Sigma}}_0 = \mathbf{\Sigma}_0$. *For each* $i \geq 0$ *compute, recursively, the matrix* $\bar{\mathbf{\Sigma}}_{i+1}$ *and the robust filtering matrices* $\mathbf{A}_i$ *and* $\mathbf{K}_i$ *as the minimizing solution of* (2.13).

We remark that the second constraint in (2.13), $\operatorname{Tr}(\mathbf{\Gamma}_{i+1}) \leq b$, is used to ensure that the matrix $\mathbf{\Gamma}_{i+1}$ is bounded. This is important because otherwise the optimal solution of (2.13), $\bar{\mathbf{\Sigma}}_{i+1}$, may become progressively ill-conditioned as i becomes large.

An alternative way of preventing ill-conditioning is to impose the following structure on $\mathbf{\Gamma}_{i+1}$,

$$\mathbf{\Gamma}_{i+1} = \begin{bmatrix} \bar{\mathbf{\Gamma}} + \hat{\mathbf{\Gamma}} & \bar{\mathbf{\Gamma}} \\ \bar{\mathbf{\Gamma}} & \bar{\mathbf{\Gamma}} \end{bmatrix} \qquad \text{for some symmetric matrices } \bar{\mathbf{\Gamma}},\ \hat{\mathbf{\Gamma}}, \tag{2.14}$$

and to use the following constraint:

$$\mathrm{Tr}\,(\hat{\mathbf{\Gamma}}) \geq \beta\ \mathrm{Tr}\,(\bar{\mathbf{\Gamma}}), \tag{2.15}$$

where $\beta > 0$ is a constant. The above structure (2.14) for $\mathbf{\Gamma}_{i+1}$ mimics the structure of the joint covariance matrix of the state of a system and its optimal estimate in the Kalman sense, and is maintained in [13]. The bound (2.15) is used to ensure that the condition number of $\mathbf{\Gamma}_{i+1}$ does not become unbounded when $\bar{\mathbf{\Gamma}}$ and $\hat{\mathbf{\Gamma}}$ become large. Indeed, notice that

$$\mathbf{\Gamma}_{i+1} = \begin{bmatrix} \bar{\mathbf{\Gamma}} + \hat{\mathbf{\Gamma}} & \bar{\mathbf{\Gamma}} \\ \bar{\mathbf{\Gamma}} & \bar{\mathbf{\Gamma}} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{\Gamma}} & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{\Gamma}} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{I} & \mathbf{I} \end{bmatrix},$$

so we only need to bound the condition number for the matrix blockdiag$\{\hat{\mathbf{\Gamma}}, \bar{\mathbf{\Gamma}}\}$. By the above factorization of $\mathbf{\Gamma}_{i+1}$ and the fact that the right-hand side of the first constraint in (2.13) is bounded from below by a positive definite matrix, we obtain that blockdiag$\{\hat{\mathbf{\Gamma}}, \bar{\mathbf{\Gamma}}\}$ is bounded from below by a positive definite matrix. Thus, the smallest eigenvalue of the matrix blockdiag$\{\hat{\mathbf{\Gamma}}, \bar{\mathbf{\Gamma}}\}$ is bounded away from zero. In the meantime, the constraint (2.15) and the fact that we are minimizing $\hat{\mathbf{\Gamma}}$ implies that the largest eigenvalue of the matrix blockdiag$\{\hat{\mathbf{\Gamma}}, \bar{\mathbf{\Gamma}}\}$ is also bounded. This implies the boundedness of the condition number of $\mathbf{\Gamma}_{i+1}$ at optimal solution.

As a result of the above discussion, we have the following alternative formulation (to (2.13)):

$$\begin{aligned} \text{minimize} \quad & \mathrm{Tr}\left(\mathbf{D} \begin{bmatrix} \mathbf{I} & -\mathbf{I} \end{bmatrix} \mathbf{\Gamma}_{i+1} \begin{bmatrix} \mathbf{I} & -\mathbf{I} \end{bmatrix}^T\right) \\ \text{subject to} \quad & \mathbf{\Gamma}_{i+1} \geq \left[\bar{\mathbf{F}}_i + \bar{\mathbf{C}}_i \mathbf{Z}_i \bar{\mathbf{E}}_i\right] \bar{\mathbf{\Sigma}}_i \left[\bar{\mathbf{F}}_i + \bar{\mathbf{C}}_i \mathbf{Z}_i \bar{\mathbf{E}}_i\right]^T + \bar{\mathbf{G}}_i \mathbf{W}_i \bar{\mathbf{G}}_i^T \qquad \forall \Theta_i \in \Omega_i, \\ & \mathbf{\Gamma}_{i+1} \text{ satisfying (2.14) and (2.15).} \end{aligned} \tag{2.16}$$

In the remainder of this paper, we will focus on the first formulation (2.12), but the second formulation (2.16) can also be treated in an analogous fashion.

We point out that the dimension of problem (2.13) is fixed rather than growing linearly with i. Moreover, it will be shown that (2.13) is convex and can be reformulated as a semidefinite program. The latter can be solved very efficiently via interior point methods [14, 10, 12]. Before we explain how to solve (2.13), we need to show that $\bar{\mathbf{\Sigma}}_i$ defined by (2.13) does provide an upper bound for $\mathbf{\Sigma}_i$ for all $i \geq 0$. We have the following theorem.

THEOREM 2.1. *Let $\bar{\mathbf{\Sigma}}_0 = \mathbf{\Sigma}_0$. For $i \geq 1$, let $\bar{\mathbf{\Sigma}}_i$ be defined as in* (2.13)*. Then there holds*

$$\bar{\mathbf{\Sigma}}_i \geq \mathbf{\Sigma}_i \qquad \forall \Theta_j \in \Omega_j, \quad j = 1, 2, \ldots, i-1. \tag{2.17}$$

Proof. The theorem can be proved by mathematical induction. In particular, for $i = 0$ we have $\bar{\mathbf{\Sigma}}_0 = \mathbf{\Sigma}_0$. Suppose that (2.17) holds for $i = k$. Since $\bar{\mathbf{\Sigma}}_{k+1}$ is the optimal solution of (2.13), it follows from the constraint of (2.13) that

$$\bar{\mathbf{\Sigma}}_{k+1} \geq \left[\bar{\mathbf{F}}_k + \bar{\mathbf{C}}_k \mathbf{Z}_k \bar{\mathbf{E}}_k\right] \bar{\mathbf{\Sigma}}_k \left[\bar{\mathbf{F}}_k + \bar{\mathbf{C}}_k \mathbf{Z}_k \bar{\mathbf{E}}_k\right]^T + \bar{\mathbf{G}}_k \mathbf{W}_k \bar{\mathbf{G}}_k^T \qquad \forall \Theta_k \in \Omega_k. \tag{2.18}$$

By the inductive hypothesis we have

$$\bar{\mathbf{\Sigma}}_k \geq \mathbf{\Sigma}_k \qquad \forall \Theta_j \in \Omega_j, \quad j = 1, 2, \ldots, (k-1).$$

Combining this with (2.18), we obtain

$$\begin{aligned}\bar{\mathbf{\Sigma}}_{k+1} &\geq \left[\bar{\mathbf{F}}_k + \bar{\mathbf{C}}_k \mathbf{Z}_k \bar{\mathbf{E}}_k\right] \mathbf{\Sigma}_k \left[\bar{\mathbf{F}}_k + \bar{\mathbf{C}}_k \mathbf{Z}_k \bar{\mathbf{E}}_k\right]^T + \bar{\mathbf{G}}_k \mathbf{W}_k \bar{\mathbf{G}}_k^T \\ &= \mathbf{\Sigma}_{k+1} \qquad\qquad\qquad \forall \Theta_j \in \Omega_j, \quad j = 1, 2, \ldots, k,\end{aligned}$$

where the last step is due to (2.7) for the particular value of Θ_j which represents the actual error in the model. This completes the induction proof. ∎

In common with the existing approaches to the finite-horizon robust filtering problem, we do not have a sufficient condition for the convergence of the estimator $\bar{\mathbf{\Sigma}}_i$ as i tends to infinity. However, we now provide some necessary conditions. (These conditions are analogous to those in [13].)

THEOREM 2.2. *Suppose the system* (2.1)–(2.4) *is time-invariant in the sense that the data matrices* $\mathbf{H}_i$, $\mathbf{C}_{1,i}$, $\mathbf{C}_{2,i}$, $\mathbf{G}_i$, $\mathbf{E}_i$, $\bar{\mathbf{R}}_i$, *and* $\bar{\mathbf{Q}}_i$ *are fixed and independent of* i. *Then the solution* $\bar{\mathbf{\Sigma}}_i$ *converges to some* $\bar{\mathbf{\Sigma}}$ *only if the set of uncertain systems* (2.1)–(2.2) *is quadratically stable.*

Proof. Let $\underline{u}_i$ and $\underline{v}_i$ be zero. By constraint (2.13) and the fact that $\bar{\mathbf{\Sigma}}_i \to \bar{\mathbf{\Sigma}}$, we have

$$\bar{\mathbf{\Sigma}} \geq \left[\bar{\mathbf{F}} + \bar{\mathbf{C}} \mathbf{Z}_i \bar{\mathbf{E}}\right] \bar{\mathbf{\Sigma}} \left[\bar{\mathbf{F}} + \bar{\mathbf{C}} \mathbf{Z}_i \bar{\mathbf{E}}\right]^T \qquad \forall \mathbf{Z}_i \text{ with } \|\mathbf{Z}_i\| \leq 1.$$

This shows that the augmented linear system (2.6) is quadratically stable. This is because the above relation easily implies that the quadratic Lyapunov function $V(\underline{\bar{x}}, i) = -\underline{\bar{x}}_i^T \bar{\mathbf{\Sigma}} \underline{\bar{x}}_i \geq 0$ and that, for all admissible systems, $V(\underline{\bar{x}}, i+1) \leq V(\underline{\bar{x}}, i)$ if the process noise $\underline{\bar{u}}_i = 0$. By construction, $\underline{x}_i$ is a component of $\underline{\bar{x}}_i$; therefore the quadratic stability of (2.6) (in this time-invariant case) implies the quadratic stability of (2.1)–(2.2) for all admissible systems. ∎

3. Robust SDP solution. In this section, we shall develop an SDP [14] formulation for the robust state estimator problem (in particular, the problem (2.13)). This will then allow for efficient numerical solutions via recent interior point methods. We begin by noting that the finite-horizon robust state estimator problem (2.13) has a constraint of the form

$$\mathbf{\Gamma}_{i+1} \geq \left[\bar{\mathbf{F}}_i + \bar{\mathbf{C}}_i \mathbf{Z}_i \bar{\mathbf{E}}_i\right] \bar{\mathbf{\Sigma}}_i \left[\bar{\mathbf{F}}_i + \bar{\mathbf{C}}_i \mathbf{Z}_i \bar{\mathbf{E}}_i\right]^T + \bar{\mathbf{G}}_i \mathbf{W}_i \bar{\mathbf{G}}_i^T \ \ \forall \Theta_i = (\Delta\mathbf{Q}_i, \Delta\mathbf{R}_i, \mathbf{Z}_i) \in \Omega_i, \tag{3.1}$$

which contains an uncertainty vector $\Theta_i = (\Delta\mathbf{Q}_i, \Delta\mathbf{R}_i, \mathbf{Z}_i)$. Recall that $\mathbf{W}_i = \text{blockdiag}(\bar{\mathbf{Q}}_i + \Delta\mathbf{Q}_i, \bar{\mathbf{R}}_i + \Delta\mathbf{R}_i)$ and that by (2.4) we have

$$-\epsilon\mathbf{I} \leq \Delta\mathbf{Q}_i \leq \epsilon\mathbf{I}, \quad -\epsilon\mathbf{I} \leq \Delta\mathbf{R}_i \leq \epsilon\mathbf{I}.$$

Therefore, by choosing the upper bound for $\mathbf{W}_i$, the constraint (3.1) holds for all $\Theta_i = (\Delta\mathbf{Q}_i, \Delta\mathbf{R}_i, \mathbf{Z}_i) \in \Omega_i$ if and only if the following holds:

$$\mathbf{\Gamma}_{i+1} \geq \left[\bar{\mathbf{F}}_i + \bar{\mathbf{C}}_i \mathbf{Z}_i \bar{\mathbf{E}}_i\right] \bar{\mathbf{\Sigma}}_i \left[\bar{\mathbf{F}}_i + \bar{\mathbf{C}}_i \mathbf{Z}_i \bar{\mathbf{E}}_i\right]^T + \bar{\mathbf{G}}_i \bar{\mathbf{W}}_i \bar{\mathbf{G}}_i^T \qquad \forall \mathbf{Z}_i \text{ with } \|\mathbf{Z}_i\| \leq 1,$$

where

$$\bar{\mathbf{W}}_i = \text{blockdiag}(\bar{\mathbf{Q}}_i + \epsilon\mathbf{I}, \bar{\mathbf{R}}_i + \epsilon\mathbf{I}).$$

We rearrange the above inequality as follows:

$$\mathbf{\Gamma}_{i+1} - \left[\bar{\mathbf{F}}_i + \bar{\mathbf{C}}_i\mathbf{Z}_i\bar{\mathbf{E}}_i \;\; \bar{\mathbf{G}}_i\right] \begin{bmatrix} \bar{\mathbf{\Sigma}}_i & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{W}}_i \end{bmatrix} \left[\bar{\mathbf{F}}_i + \bar{\mathbf{C}}_i\mathbf{Z}_i\bar{\mathbf{E}}_i \;\; \bar{\mathbf{G}}_i\right]^T \geq \mathbf{0} \quad \forall \mathbf{Z}_i \text{ with } \|\mathbf{Z}_i\| \leq 1.$$

Using the Schur complement, the above constraint is equivalent to

$$\begin{bmatrix} \bar{\mathbf{\Sigma}}_i^{-1} & \mathbf{0} & (\bar{\mathbf{F}}_i + \bar{\mathbf{C}}_i\mathbf{Z}_i\bar{\mathbf{E}}_i)^T \\ \mathbf{0} & \bar{\mathbf{W}}_i^{-1} & \bar{\mathbf{G}}_i^T \\ (\bar{\mathbf{F}}_i + \bar{\mathbf{C}}_i\mathbf{Z}_i\bar{\mathbf{E}}_i) & \bar{\mathbf{G}}_i & \mathbf{\Gamma}_{i+1} \end{bmatrix} \geq \mathbf{0} \quad \forall \mathbf{Z}_i \text{ with } \|\mathbf{Z}_i\| \leq 1. \tag{3.2}$$

Note that both $\bar{\mathbf{\Sigma}}_i$ and $\bar{\mathbf{W}}_i$ are positive definite and hence invertible.

For each fixed $\mathbf{Z}_i$ with $\|\mathbf{Z}_i\| \leq 1$, the above constraint (3.2) is a so-called linear matrix inequality (LMI) in the matrix variables $\{\mathbf{\Gamma}_{i+1},\ \mathbf{A}_i,\ \mathbf{K}_i\}$ which is convex. (Recall that the matrix variables $\{\mathbf{A}_i,\ \mathbf{K}_i\}$ are buried, linearly, in $\bar{\mathbf{F}}_i$, $\bar{\mathbf{G}}_i$, and $\bar{\mathbf{C}}_i$.) Thus the feasible region described by the above constraint is the intersection of convex regions described by an infinite number of linear matrix inequalities parameterized by $\mathbf{Z}_i$. This implies that the feasible region of (2.13) is convex. It is now clear that the original robust filtering problem (2.13) is equivalent to

$$\begin{aligned} &\text{minimize} && \operatorname{Tr}\left(\mathbf{D}\begin{bmatrix} \mathbf{I} & -\mathbf{I} \end{bmatrix}\mathbf{\Gamma}_{i+1}\begin{bmatrix} \mathbf{I} & -\mathbf{I} \end{bmatrix}^T\right) \\ &\text{subject to} && \{\mathbf{\Gamma}_{i+1},\ \mathbf{A}_i,\ \mathbf{K}_i\} \text{ satisfying (3.2)}, \\ & && \operatorname{Tr}(\mathbf{\Gamma}_{i+1}) \leq b. \end{aligned} \tag{3.3}$$

The formulation (3.3) is given as an SDP, except that the data matrices are subject to uncertainty $\mathbf{Z}_i$. Therefore it cannot be solved by standard SDP methods. The constraints in (3.3) imply that the solution must remain feasible for all allowable perturbations. This is precisely the intent of a robust filter solution. An SDP problem for which the data matrices are uncertain is called a robust SDP. In the next subsection, we introduce a technique for converting a robust SDP into a standard SDP, which can then be solved efficiently by the recent interior point methods.

3.1. The robust SDP. SDP is a convex optimization problem and can be solved in polynomial time using efficient algorithms such as the primal-dual interior point methods [14, 10, 12]. An SDP consists of minimizing a linear objective subject to an LMI constraint,

$$\begin{aligned} &\text{minimize} && \underline{c}^T\underline{\alpha} \\ &\text{subject to} && \mathbf{B}(\underline{\alpha}) = \mathbf{B}_0 + \sum_{k=1}^{q} \alpha_k\mathbf{B}_k \geq \mathbf{0}, \end{aligned}$$

where $\underline{c} \in \mathcal{R}^q$, $\underline{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_q)^T$, and the symmetric matrices $\mathbf{B}_k = \mathbf{B}_k^T \in \mathcal{R}^{l\times l}$, $k = 0, \ldots, q$, are some given data matrices. In our case, these data matrices are subject to uncertainty. We can incorporate some linear uncertainty into $\mathbf{B}(\underline{\alpha})$ in the following way. Let $\mathbf{B}(\underline{\alpha}, \Delta)$ be a symmetric matrix-valued function of two variables $\underline{\alpha}$ and Δ of the form

$$\mathbf{B}(\underline{\alpha}, \Delta) = \mathbf{B}(\underline{\alpha}) + \mathbf{N}\Delta\mathbf{M}(\underline{\alpha}) + \mathbf{M}(\underline{\alpha})^T\Delta^T\mathbf{N}^T, \tag{3.4}$$

where $\mathbf{B}(\underline{\alpha})$ is defined in (3.1), $\mathbf{N}$ and $\mathbf{M}(\underline{\alpha})$ are given matrices, Δ is a perturbation which is unknown but bounded. We define the robust feasible set by

$$\mathcal{A} = \{\underline{\alpha} \in \mathcal{R}^q \mid \mathbf{B}(\underline{\alpha}, \Delta) \geq 0 \text{ for every } \Delta \text{ with } \|\Delta\| \leq 1\}.$$

The robust SDP is then defined as

$$\begin{array}{ll} \text{minimize} & \underline{c}^T \underline{\alpha} \\ \text{subject to} & \underline{\alpha} \in \mathcal{A}. \end{array} \tag{3.5}$$

The following lemma shows how such a robust SDP can be solved using a conventional SDP. It is a simple corollary of a classical result on quadratic inequalities referred to as the $\mathcal{S}$-procedure, and its proof is detailed in [2].

LEMMA 3.1. *Let $\mathbf{B} = \mathbf{B}^T$, $\mathbf{N}$, and $\mathbf{M}$ be real matrices of appropriate size. We have*

$$\mathbf{B} + \mathbf{N}\Delta\mathbf{M} + \mathbf{M}^T\Delta^T\mathbf{N}^T \geq \mathbf{0} \tag{3.6}$$

for every Δ, $\|\Delta\| \leq 1$, if and only if there exists a scalar ρ such that

$$\begin{bmatrix} \mathbf{B} - \rho\mathbf{N}\mathbf{N}^T & \mathbf{M}^T \\ \mathbf{M} & \rho\mathbf{I} \end{bmatrix} \geq \mathbf{0}. \tag{3.7}$$

As a consequence, the robust SDP (3.5) *can be formulated as the following standard SDP in variables $\underline{\alpha}$ and ρ:*

$$\begin{array}{ll} \textit{minimize} & \underline{c}^T \underline{\alpha} \\ \textit{subject to} & \begin{bmatrix} \mathbf{B}(\underline{\alpha}) - \rho\mathbf{N}\mathbf{N}^T & \mathbf{M}(\underline{\alpha})^T \\ \mathbf{M}(\underline{\alpha}) & \rho\mathbf{I} \end{bmatrix} \geq \mathbf{0}. \end{array} \tag{3.8}$$

We now return to the problem in (3.3) and factorize the LMI constraint matrix (3.2) according to the structure in (3.4). In such a factorization, the decision variable $\underline{\alpha}$ in (3.4) will correspond to a concatenation of the elements of the matrix variables $\mathbf{\Gamma}_{i+1}$, $\mathbf{A}_i$, and $\mathbf{K}_i$ in (3.2), and the perturbation Δ in (3.4) will correspond to $\mathbf{Z}_i$ in (3.2). The factorization is given by

$$\mathbf{B}(\underline{\alpha}) = \begin{bmatrix} \bar{\mathbf{\Sigma}}_i^{-1} & \mathbf{0} & \bar{\mathbf{F}}_i^T \\ \mathbf{0} & \bar{\mathbf{W}}_i^{-1} & \bar{\mathbf{G}}_i^T \\ \bar{\mathbf{F}}_i & \bar{\mathbf{G}}_i & \mathbf{\Gamma}_{i+1} \end{bmatrix}, \tag{3.9}$$

where

$$\bar{\mathbf{F}}_i = \begin{bmatrix} \mathbf{F}_i & \mathbf{0} \\ \mathbf{K}_i\mathbf{H}_i & \mathbf{A}_i - \mathbf{K}_i\mathbf{H}_i \end{bmatrix}, \ \bar{\mathbf{G}}_i = \begin{bmatrix} \mathbf{G}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_i \end{bmatrix}, \ \bar{\mathbf{W}}_i = \begin{bmatrix} \bar{\mathbf{Q}}_i + \epsilon\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{R}}_i + \epsilon\mathbf{I} \end{bmatrix},$$

and

$$\mathbf{N}\Delta\mathbf{M}(\underline{\alpha}) + \mathbf{M}(\underline{\alpha})^T\Delta^T\mathbf{N}^T = \begin{bmatrix} \mathbf{0} & \mathbf{0} & (\bar{\mathbf{C}}_i\mathbf{Z}_i\bar{\mathbf{E}}_i)^T \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \bar{\mathbf{C}}_i\mathbf{Z}_i\bar{\mathbf{E}}_i & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{3.10}$$

with

$$\bar{\mathbf{C}}_i\mathbf{Z}_i\bar{\mathbf{E}}_i = \begin{bmatrix} \mathbf{C}_{1,i}\mathbf{Z}_i\mathbf{E}_i & \mathbf{0} \\ \mathbf{K}_i\mathbf{C}_{2,i}\mathbf{Z}_i\mathbf{E}_i & \mathbf{0} \end{bmatrix}.$$

The matrices $\mathbf{N}$ and $\mathbf{M}(\underline{\alpha})$ are given by

$$\mathbf{M}(\underline{\alpha}) = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{C}_{1,i}^T & \mathbf{C}_{2,i}^T \mathbf{K}_i^T \end{bmatrix},$$

$$\mathbf{N} = \begin{bmatrix} \mathbf{E}_i^T \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}. \tag{3.11}$$

Now we are in a position to apply Lemma 3.1 to convert the robust SDP (3.3) into the following standard SDP in the variables $\mathbf{\Gamma}_{i+1}$, $\mathbf{A}_i$, $\mathbf{K}_i$, and ρ:

$$\begin{aligned} &\text{minimize} && \operatorname{Tr}\left(\mathbf{D}\begin{bmatrix} \mathbf{I} & -\mathbf{I} \end{bmatrix}\mathbf{\Gamma}_{i+1}\begin{bmatrix} \mathbf{I} & -\mathbf{I} \end{bmatrix}^T\right) \\ &\text{subject to} && \begin{bmatrix} \mathbf{B}(\underline{\alpha}) - \rho\mathbf{N}\mathbf{N}^T & \mathbf{M}(\underline{\alpha})^T \\ \mathbf{M}(\underline{\alpha}) & \rho\mathbf{I} \end{bmatrix} \geq \mathbf{0}, \\ & && \operatorname{Tr}(\mathbf{\Gamma}_{i+1}) \leq b, \end{aligned} \tag{3.12}$$

where the variable $\underline{\alpha}$ contains columns of the matrices $\mathbf{\Gamma}_{i+1}$, $\mathbf{K}_i$, and $\mathbf{A}_i$, and the matrices $\mathbf{B}(\underline{\alpha})$, $\mathbf{N}$, and $\mathbf{M}(\underline{\alpha})$ are given by (3.9) and (3.11), respectively.

Note that, for each i, problem (3.12) is fixed in dimension (i.e., does not grow with i). It is a standard SDP problem which has a unique solution and satisfies the usual regularity condition, provided that the primal and dual of (3.12) are strictly feasible and for every $\underline{\alpha}$, $\mathbf{M}(\underline{\alpha}) \neq \mathbf{0}$ and $\begin{bmatrix} \mathbf{N} & \mathbf{M}^T(\underline{\alpha}) \end{bmatrix}^T$ is full column-rank. As such, the problem can be solved very efficiently by an interior point method; in particular, by the homogeneous self-dual method [10, 12]. In our computational experience, the number of iterations required to solve each SDP is fixed (no more than 8), and therefore the proposed technique can be regarded as a recursive filtering method.

To make a formal comparison of the computational complexity of our robust filtering method with those of [15, 13], we need to recall the notations of our model (2.1): n denotes the number of states, m denotes the number of inputs, and p denotes the number of measured outputs. Xie's method [15] is a "one-shot" method, and hence the robust observer matrix is calculated only once. The cost of this computation is $O((n+p)^3)$. However, Xie's method [15] works only for time-invariant systems. On the other hand, Theodor's method [13] is iterative. The cost per iteration is $O((n+p)^3 + n^2 m)$. Our method is also iterative. Using a general purpose interior point SDP solver requires $O((n+m+p)^{5/2}(n^2+np)^2)$ per filtering iteration. It is interesting to examine the above costs as the number of states in the model, n, grows. In that case, the total computational cost of Xie's method [15] is $O(n^3)$, while the cost per (filtering) iteration of Theodor's method [13] and of our proposed method are $O(n^3)$ and $O(n^{6.5})$, respectively. It is also interesting to examine the above costs as the number of measured outputs, p, grows. In that case, the total cost of Xie's method [15] is $O(p^3)$, while the cost per (filtering) iteration of Theodor's method [13] and of our method are $O(p^3)$ and $O(p^{4.5})$, respectively. We believe it is possible to reduce the complexity per iteration for our method by exploiting the sparsity structure present in our problem. This is an interesting issue for future investigation.

We now make an observation regarding the scaling of the matrices $\bar{\mathbf{C}}_i$ and $\bar{\mathbf{E}}_i$. In particular, these two matrices can be scaled and replaced by $\bar{\mathbf{C}}_i/\mu$ and $\mu\bar{\mathbf{E}}_i$, respectively. Such a scaling does not change the formulation of (3.3), nor does it affect the formulation of (3.12), because the latter is completely equivalent to the former. This

shows that the solutions to our reformulated robust filtering problem are independent of the scaling factor μ. This property is in contrast to the robust filter proposed in [13], where the solutions are "highly sensitive" [13] to the choice of μ. The scale invariance of our method with the choice of μ is a clear advantage.

However, our method also has a disadvantage in that it is sensitive to the choice of b in the second constraint in (3.12), $\mathrm{Tr}\,(\mathbf{\Gamma}_{i+1}) \leq b$. This constraint is used to ensure that the matrix $\mathbf{\Gamma}_{i+1}$ is bounded. This is important because otherwise the optimal solution of (3.12), $\bar{\mathbf{\Sigma}}_{i+1}$, may become progressively ill-conditioned as i becomes large. This phenomenon has been observed in computer simulations. In general, large values of b will allow the matrices $\{\bar{\mathbf{\Sigma}}_i : \ i = 1, 2, \ldots\}$ to become rather ill-conditioned, while small values of b may render the subproblem (3.12) infeasible. The same remark applies to the alternative formulation (2.16), where a value of $\beta > 0$ needs to be selected. Through computer experiments we found that both formulations led to filters with similar behavior and performance.

4. Numerical examples. In this section, the performance of the proposed robust state-estimation method is illustrated via simulation results. Two numerical examples are given here; the first one is the same problem as that used in [13, 15], and the second one is a target-tracking problem.

4.1. Example 1. In this example the following discrete-time linear uncertain state-space model is used:

$$\begin{aligned} \underline{x}_{i+1} &= \begin{bmatrix} 0 & -0.5 \\ 1 & 1+\delta \end{bmatrix} \underline{x}_i + \begin{bmatrix} -6 \\ 1 \end{bmatrix} u_i, \quad |\delta| < 0.3, \\ y_i &= [\,-100 \quad 10\,]\,\underline{x}_i + v_i, \\ s_i &= [\,1 \quad 0\,]\,\underline{x}_i, \end{aligned} \tag{4.1}$$

where u_i and v_i are uncorrelated zero-mean white noise signals with variances $\bar{Q} = 1$ and $\bar{R} = 1$, respectively. The value of ϵ in (2.4) is set to zero, so that there is no uncertainty in the knowledge of noise statistics. The uncertainty in (4.1) is described by the matrices

$$\mathbf{C}_1 = [0 \ \ 10]^T, \quad \mathbf{C}_2 = 0, \quad \mathbf{E} = [0 \ \ 0.03]$$

and the scalar parameter z, $|z| \leq 1$.

To determine the robust filter at each instant i, we use the MATLAB toolbox `SeDuMi` [12] to solve the robust SDP (3.12). This code requires no initialization since it is based on the self-dual formulation of the SDP. Solving the SDP (3.12) at each instant i with $b = 900$ and $\mathbf{D} = \mathrm{diag}(1, 5)$ yields a robust state estimator [of the form (2.5)] which converges to

$$\mathbf{A} = \begin{bmatrix} -0.1711 & -0.4624 \\ 1.4080 & 1.1786 \end{bmatrix} \quad \text{and} \quad \mathbf{K} = \begin{bmatrix} -0.0051 \\ 0.0047 \end{bmatrix}.$$

Note that for stability reasons the estimator (as seen in $\mathbf{D}$) weights the second component of $\underline{x}_i$ more heavily than the first component even though the goal is to estimate the first component of $\underline{x}_i$. In our simulation studies, the proposed technique is compared with the Kalman and H^∞ filters and the robust filters of [13, 15]. For this purpose, steady-state Kalman and H^∞ filters are designed for the nominal system of (4.1), i.e., $\delta = 0$. We then apply these filters to system (4.1) with $\delta = 0$, $\delta = 0.3$,

TABLE 1
Steady-state estimation error variances for different filters (results are averaged over 100 *runs).*

Filter	$\delta = -0.3$	$\delta = 0$	$\delta = 0.3$
Nominal Kalman filter	551.2	36.0	8352.8
Nominal H^∞ filter	96.0	47.2	893.9
The robust filter of [13]	51.4	51.3	54.4
The robust filter of [15]	64.0	61.4	64.4
The robust filter of [3]	51.5	49.1	53.8
Proposed robust filter	46.2	45.6	51.9

TABLE 2
Steady-state estimation error variances for our method and the method of [3].

Filter	$\delta = -0.09$	$\delta = 0$	$\delta = 0.09$
The robust filter of [3]	37.75	38.19	41.47
Proposed robust filter	37.38	37.78	40.31

and $\delta = -0.3$. The steady-state estimation error variances (i.e., $\mathcal{E}\{(s_i - \hat{s}_i)^2\}$ for sufficiently large i) for the filters are displayed in Table 1. It is clear from the table that the proposed robust filter performs far better than the nominal Kalman and H^∞ filters in the presence of modelling error.

Both our method and the methods of [13, 15] require the tuning of a certain parameter. In our case, we need to adjust the parameter b in order to prevent the iterates from becoming ill-conditioned, and the diagonal elements of $\mathbf{D}$ in order to get the best estimator. The methods of [13, 15] require the adjustment of the factor μ in the scaling of $\mathbf{C}_i/\mu$ and $\mu\mathbf{E}_i$. Our experiments suggest that our method works for $b \in [880, 5000]$, while the method of [13] converges for $\mu \in (0, 1.703]$ and diverges for values outside this range. The best performance is achieved with $\mu = 1.703$. (Note that the authors of [13] reported their choice of $\mu = 2.2$, but our own implementation of their method showed that this value of μ leads to divergence instead.)

The filter performance for the robust filter of [15] stated in Table 1 is quoted from [13]. We should point out that we could not reproduce the design of the robust filter [13] using their design method. With our own (simple) MATLAB implementation of their method, we could only produce a filter with $\mu = 1.703$, whose error covariances are 51.4, 51.3, and 54.4, rather than 46.6, 45.2, and 54.1 (as claimed in [13]) for model errors of $\delta = -0.3$, 0, and 0.3, respectively. From Table 1, we can see that the performance of the robust filters [13, 15] are inferior to the filter designed by the robust SDP method: the worst-case performance (for $\delta = -0.3,\ 0,\ 0.3$) is 51.9 for our proposed robust filter, and is 54.2 and 64.4, respectively, for the robust filters of [13] and [15]. From this example, it appears that our robust filter design is slightly superior.

Recently our approach has been further extended by Fu, de Souza, and Luo [3], who introduced multiple scaling factors in the SDP formulation and showed performance improvement when compared to the single-scaling-factor case. It should be pointed out that the single-scaling-factor case of [3] corresponds to the algorithm considered in this paper, except that we have an additional boundedness constraint $\mathrm{Tr}\,(\mathbf{\Gamma}_{i+1}) \leq b$ in our SDP subproblem (3.12). We simulated the single-scaling-factor case of [3] in Table 1 for comparison. From the simulation results, our method is slightly superior to the method of [3] in the single-scaling-factor case. This is due to the differences in the way the ill-conditioning of the bound on the covariance matrix is handled. The simulation results stated in [3] are for $\mathbf{C}_1 = [0\ \ 3]^T$ (instead of

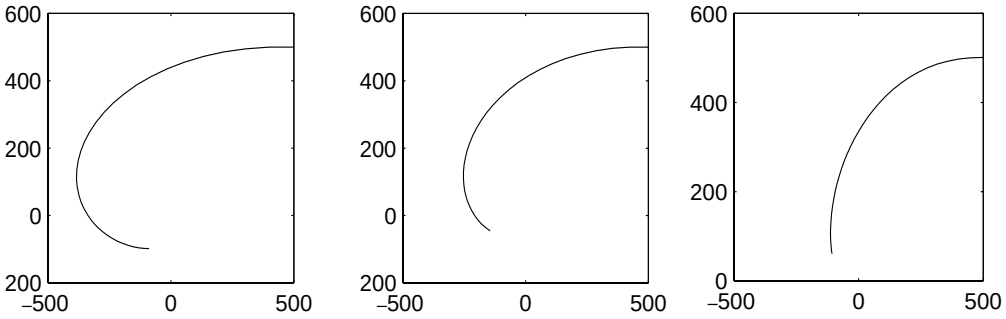


FIG. 1. *Trajectories of the target-tracking model with uncertainty* $\delta = -0.05$ *(left),* $\delta = 0$ *(middle),* $\delta = 0.05$ *(right).*

$\mathbf{C}_1 = [0\ \ 10]^T$), which means that the simulated cases in [3] have only 30% of the uncertainty considered in Table 1. We also compared our method with the method of [3] for the case $\mathbf{C}_1 = [0\ \ 3]^T$, and the simulation results show that our method is still slightly superior to the method of [3] (Table 2).

4.2. A tracking example. In this example a target-tracking case is considered. The discrete-time state-space model is given by

$$\begin{aligned} \underline{x}_{i+1} &= \begin{bmatrix} 0.95 & -0.1+\delta \\ 0.05 & 0.95 \end{bmatrix} \underline{x}_i + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_i, \quad |\delta| < 0.05, \\ y_i &= [\,1 \quad 0\,]\,\underline{x}_i + v_i, \\ s_i &= [\,1 \quad 0\,]\,\underline{x}_i, \end{aligned} \tag{4.2}$$

where u_i and v_i are uncorrelated zero-mean white noise signals with variances $\bar{Q} = 1$ and $\bar{R} = 1$, respectively. The value of ϵ in (2.4) is set to zero, so that there is no uncertainty in the knowledge of noise statistics. The uncertainty in (4.2) is described by the matrices

$$\mathbf{C}_1 = [0.05\ \ 0]^T, \quad \mathbf{C}_2 = 0, \quad \mathbf{E} = [0\ \ 1]$$

and the uncertainty parameter z, $|z| \le 1$.

In this model, the state vector $\underline{x}_i$ represents the position of a target in a two-dimensional coordinate system, and the observation y_i is a noise-corrupted version of the first coordinate. The target is making a counter-clockwise turn starting from the position $\underline{x}_0 = [500, 500]^T$. The unknown parameter δ describes the uncertainty in the turning rate of the trajectory. Three possible trajectories from this model are shown in Figure 1.

Solving the SDP (3.12) for each value of i, with $b = 1100$ and $\mathbf{D} = \mathrm{diag}(1, 7)$, yields a robust state estimator (of the form (2.5)) which converges to

$$\mathbf{A} = \begin{bmatrix} 0.9500 & -0.1016 \\ 0.0500 & 0.9644 \end{bmatrix} \quad \text{and} \quad \mathbf{K} = \begin{bmatrix} 0.7560 \\ 0.0130 \end{bmatrix}.$$

TABLE 3

Steady-state estimation error variances for different filters for the tracking problem (results are averaged over 100 *runs).*

Filter	$\delta = -0.05$	$\delta = 0$	$\delta = 0.05$
Nominal Kalman filter	6425.2	1.4	11404.0
The robust filter of [13]	199.7	53.6	703.5
The robust filter of [15]	1309.6	666.9	549.2
Proposed robust filter	187.9	52.8	693.4

We have compared our method with the methods of [13, 15], as well as the nominal Kalman filter. The result is shown in Table 3.

From the simulation results, it appears that the filter designed by our method is superior to the filters obtained via the methods of [13] and [15]. In designing the filters by the methods of [13, 15], we have adjusted their corresponding adjustable parameters (e.g., the parameter μ in the scaling of $\mathbf{C}_i/\mu$ and $\mu\mathbf{E}_i$) and picked the filters which generate the best performance guarantees. The method of [15] requires that an additional parameter, denoted ϵ in [15], be tuned. We tuned this parameter to a value of 10 in our implementation. Note that, in the presence of uncertainty, the nominal Kalman filter performs far worse than the robust filters, as expected.

We have also compared our robust filter design to the robust filters of [13, 15] in higher-dimensional cases. We found that the relative steady-state performance of these filters is similar to that in the above examples. From the computational standpoint, our method is quite efficient, as the SDP solved at each instant has a fixed dimension, and the interior point method used to solve it is fast. However, our method does incur a greater per-sample computational cost than methods based on *analytic* recursions, such as the Kalman filter and the robust Kalman filter in [13]. (The robust filter in [15] is a "one-shot" filter which does not vary with i.) For example, on a 200MHz Pentium Pro workstation, using a general purpose SDP solver [12] under the MATLAB environment, the per-sample computation time of our method in the above examples was around 1s, whereas that of the method in [13] was around 5ms. (Recall, however, that the performance of the method in [13] is "highly sensitive" to the parameter which must be tuned.) In future work, it will be useful to design special purpose interior point algorithms which exploit the matrix structure of the SDP in (3.12) to reduce the per-sample computational complexity of our new method. Such a reduction of computational complexity is essential if one is to implement the proposed robust filtering algorithm on a DSP (digital signal processing) chip for a real-time filtering application.

5. Conclusions. In this paper, we have proposed a new state estimator for linear uncertain systems. The method is robust to norm-bounded parameter uncertainties on the system model as well as to bounded uncertainties on the noise statistics. In the new technique, the estimation problem was formulated as a convex optimization problem, which is then solved using the recent primal-dual self-dual interior point method. This requires at most 8 iterations (or matrix inversions) and therefore can be regarded as a recursive filtering method. The formulation guarantees the existence of robust solutions via a semidefinite program and, under some conditions, the solution to that semidefinite program is unique. The proposed technique compared favorably with the well-known Kalman and H^∞ filters and the "robust" filters of [13, 15]. When applied to the problem of target tracking, the new method has led to a significant improvement in tracking performance.

REFERENCES

[1] V. D. Blondel and J. N. Tstitsiklis, *A survey of computational complexity results in systems and control*, Automatica J. IFAC, 36 (2000), pp. 1249–1274.

[2] L. El Ghaoui and H. Lebret, *Robust solution to least-squares problems with uncertain data*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 1035–1064.

[3] M. Fu, C. E. de Souza, and Z.-Q. Luo, *Finite horizon robust Kalman filter design*, IEEE Trans. Signal Processing, to appear.

[4] M. Green and D. J. N. Limebeer, *Linear Robust Control*, Prentice–Hall, Englewood Cliffs, NJ, 1995.

[5] B. Hassibi, A. H. Sayed, and T. Kailath, *Linear estimation in Krein spaces—Part* I*: Theory & Part* II*: Applications*, IEEE Trans. Automat. Control, 41 (1996), pp. 18–49.

[6] S. Haykin, *Adaptive Filter Theory*, 3rd ed., Prentice–Hall, Englewood Cliffs, NJ, 1996.

[7] D. Hinrichsen and A. J. Pritchard, *Robustness measures for linear systems with application to stability radii of Hurwitz and Schur polynomials*, Internat. J. Control, 55 (1992), pp. 809–844.

[8] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*, Prentice–Hall, Upper Saddle River, NJ, 2000.

[9] H. Li and M. Fu, *A linear matrix inequality approach to robust H_∞ filtering*, IEEE Trans. Signal Process., 45 (1997), pp. 2238–2250.

[10] Z.-Q. Luo, J. F. Sturm, and S. Zhang, *Conic convex programming and self-dual embedding*, Optim. Methods Softw., 14 (2000), pp. 169–218.

[11] A. H. Sayed and T. Kailath, *A state-space approach to adaptive RLS filtering*, IEEE Signal Processing Magazine, 11 (1994), pp. 18–60.

[12] J. F. Sturm, *Using SeDuMi* 1.02*, a MATLAB toolbox for optimization over symmetric cones*, Optim. Methods Softw., 11–12 (1999), pp. 625–653. Available online at http://fewcal.kub.nl/sturm/software/sedumi.html.

[13] Y. Theodor and U. Shaked, *Robust discrete-time minimum-variance filtering*, IEEE Trans. Signal Process., 44 (1996), pp. 181–189.

[14] L. Vandenberghe and S. Boyd, *Semidefinite programming*, SIAM Rev., 38 (1996), pp. 49–95.

[15] L. Xie, Y. C. Soh, and C. E. de Souza, *Robust Kalman filtering for uncertain discrete-time systems*, IEEE Trans. Automat. Control, 39 (1994), pp. 1310–1314.

[16] L. Xie, C. E. de Souza, and M. Fu, *H_∞ estimation for linear discrete-time uncertain systems*, Internat. J. Robust Nonlinear Control, 1 (1991), pp. 111–123.

[17] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*, Prentice–Hall, Englewood Cliffs, NJ, 1996.

SIAM J. OPTIM.
Vol. 12, No. 3, pp. 756–769

AN EXPLICIT EQUIVALENT POSITIVE SEMIDEFINITE PROGRAM FOR NONLINEAR 0-1 PROGRAMS*

JEAN B. LASSERRE†

Abstract. We consider the general nonlinear optimization problem in 0-1 variables and provide an explicit equivalent positive semidefinite program in $2^n - 1$ variables. The optimal values of both problems are identical. From every optimal solution of the former, one easily finds an optimal solution of the latter, and conversely, from every solution of the latter, one may construct an optimal solution of the former. For illustration, the equivalent positive semidefinite program is explicated for quadratic 0-1 programs and MAX-CUT in $\mathbb{R}^3$. For unconstrained 0-1 programs, a special representation in terms of a weighted sum of squares is provided.

Key words. integer programming, semidefinite programming, moment problem, real algebraic geometry

AMS subject classifications. 90C22, 90C2, 90C27, 90C10

PII. S1052623400380079

1. Introduction. This paper is concerned with the general nonlinear problem in 0-1 variables

$$\mathbb{P} \rightarrow p^* := \min_{x \in \{0,1\}^n} \{g_0(x) \,|\, g_k(x) \geq 0,\ k = 1, \ldots, m\}, \tag{1.1}$$

where all the $g_k(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ are real-valued polynomials of degree $2v_k - 1$ if odd, or $2v_k$ if even. Equality constraints are allowed via two opposite inequalities. This general formulation encompasses 0-1 linear and nonlinear programs, among them the quadratic assignment problem. In the MAX-CUT problem, the discrete set $\{0,1\}^n$ is replaced by $\{-1,1\}^n$.

In our recent work [7], and for general optimization problems involving polynomials, we have provided a sequence $\{\mathbb{Q}_i\}$ of positive semidefinite (psd) programs (or semidefinite program (SDP) relaxations) with the property that $\inf \mathbb{Q}_i \uparrow p^*$ as $i \rightarrow \infty$, under a certain assumption on the semialgebraic constraint set $\{g_k(x) \geq 0,\ k = 1, \ldots, m\}$. The approach was based on recent results in real algebraic geometry on the representation of polynomials, positive on a compact semialgebraic set, a theory dual to the theory of *moments*. For general 0-1 programs—that is, with the additional constraint $x \in \{0,1\}^n$—we have shown in Lasserre [8] that this assumption on the constraint set is automatically satisfied and the SDP relaxations $\{\mathbb{Q}_i\}$ simplify to a specific form with at most $2^n - 1$ variables, no matter how large i might be. The approach followed in [7] and [8] is different in spirit from the *lift and project* iterative procedure of Lovász and Schrijver [9] for 0-1 programs (see also extensions in Kojima and Tunçel [5]), which requires that a weak separation oracle be available for the homogeneous cone associated with the constraints. In the *lift and project* procedure, the description of the convex hull of the constraint set is *implicit* (via successive projections), whereas we provide in this paper an explicit description of the equivalent psd program, with a simple interpretation. Although different, our approach is closer in spirit to the successive linear program (LP) relaxations in the RLT (reformulation

*Received by the editors October 26, 2000; accepted for publication (in revised form) July 15, 2001; published electronically February 8, 2002.
http://www.siam.org/journals/siopt/12-3/38007.html
†LAAS-CNRS, 7 Avenue du Colonel Roche, 31077 Toulouse Cédex, France (lasserre@laas.fr).

linearization technique) procedure of Sherali and Adams [12] for 0-1 linear programs, in which each of the linear original constraints is multiplied by suitable polynomials of the form $\Pi_{i\in J_1} x_i \Pi_{j\in J_2}(1-x_j)$ and then linearized in a higher dimension space via several changes of variables to obtain an LP. The last relaxation in the hierarchy of RLT produces the convex hull of the feasible set. This also extends to a special class of 0-1 polynomial programs and mixed integer programs (see Sherali and Adams [12, 13]).

The contribution of the present paper is threefold.

(a) First, we show that in addition to the asymptotic convergence already proved in [7, 8], the sequence of SDP relaxations $\{\mathbb{Q}_i\}$ is in fact *finite*; that is, the optimal value p^* is also $\min \mathbb{Q}_i$ for all $i \geq n + v$ with $v := \max_k v_k$. Moreover, every optimal solution y^* of $\mathbb{Q}_i$ is the (finite) vector of moments of some probability measure supported on optimal solutions of $\mathbb{P}$. Therefore, every 0-1 program is in fact *equivalent* to a continuous psd program in $2^n - 1$ variables for which an explicit form as well as a simple interpretation are available. The projection of the feasible set defined by the *linear matrix inequality* (LMI) constraints of this psd program onto the subspace spanned by the first n variables provides the convex hull of the original constraint set. Note that the result holds for *arbitrary* 0-1 constrained programs, that is, with arbitrary polynomial criteria and constraints. (No weak separation oracle is needed as in the lift and project procedure [9].) This is because the theory of representation of polynomials positive on a compact semialgebraic set and its dual theory of moments make no assumption on the semialgebraic set, except compactness (it can be nonconvex, disconnected). For illustration purposes, we provide the equivalent psd programs for quadratic 0-1 programs and MAX-CUT problems in $\mathbb{R}^3$.

(b) As a by-product, for unconstrained 0-1 problems $\mathbb{P}$, we show that with $g_0(x)$ an arbitrary polynomial, $g_0(x) - p^*$ can be written as a sum of squares of degree less than $n + v$, weighted by the polynomials $x_k^2 - x_k$ defining the integrality constraints. For instance, for an arbitrary quadratic form $x'Qx$ we obtain that

$$x'Qx - p^* = \sum_j q_j(x)^2 + \sum_{k=1}^{n} (x_k^2 - x_k) \left[\sum_j u_{kj}(x)^2 - \sum_j v_{kj}(x)^2 \right] \tag{1.2}$$

for some polynomials $\{q_j(x)\}$ of degree at most $n+1$ and some polynomials $\{u_{kj}(x)\}$, $\{v_{kj}(x)\}$ of degree at most n. A similar result also holds for MAX-CUT problems (in (1.2) replace $(x_k^2 - x_k)$ by $(x_k^2 - 1)$). Hence, getting an optimal solution of an unconstrained 0-1 program at a relaxation of order less than n depends on our ability to represent $g_0(x) - p^*$ as in (1.2), but with polynomials of degree less than n.

(c) For practical computational purposes, the preceding results are of little value, for the number of variables grows exponentially with the size of the problem. Fortunately, in many cases, the optimal value is also the optimal value of some $\mathbb{Q}_i$ for $i \ll n$. For instance, on a sample of 50 randomly generated MAX-CUT problems in $\mathbb{R}^{10}$, the optimal value p^* was always obtained at the $\mathbb{Q}_2$ relaxation (in which case $\mathbb{Q}_2$ is a psd program with "only" 385 variables, compared with $2^{10} - 1 = 1023$). However, when solving $\mathbb{Q}_i$, one would like to determine whether the optimal value p^* is indeed achieved in the case of multiple optimal solutions of $\mathbb{P}$ (as in MAX-CUT problems where both x^* and $-x^*$ are solutions); it is not easy to check by a direct inspection of an optimal solution y^* of $\mathbb{Q}_i$ whether y^* is the vector of moments of some probability measure supported on optimal solutions of $\mathbb{P}$. Our next contribution is to provide a test at an optimal solution of $\mathbb{Q}_i$ to detect whether $p^* = \min \mathbb{Q}_i$. This test amounts to

checking whether two moment matrices have same rank. The proof relies on a recent deep result of Curto and Fialkow [2] on the $\mathbb{K}$-moment problem.

In a sense, one may say that nonlinear 0-1 optimization problems are "easier" than nonconvex continuous optimization problems. For the former, an equivalent psd program is available and the sequence of SDP relaxations is finite, whereas for the latter only asymptotic convergence is ensured (with, however, finite termination in many cases, that is, when the polynomial $g_0(x) - p^*$ has a certain representation in terms of weighted squares (see Lasserre [7])).

2. Notation and definitions. We adopt the notation in Lasserre [7], which, for the sake of clarity, we reproduce here.

Given any two real-valued symmetric matrices A, B, let $\langle A, B\rangle$ denote the usual scalar product trace(AB) and let $A \succeq B$ (resp., $A \succ B$) stand for $A - B$ positive semidefinite (resp., $A - B$ positive definite). Let

$$1, x_1, x_2, \ldots, x_n, x_1^2, x_1x_2, \ldots, x_1x_n, x_2^2, x_2x_3, \ldots, x_n^2, \ldots, x_1^r, \ldots, x_n^r \tag{2.1}$$

be a basis for the vector space of real-valued polynomials of degree at most r, and let $s(r)$ be its dimension. Then, an r-degree polynomial $p(x) : \mathbb{R}^n \to \mathbb{R}$ is written

$$p(x) = \sum_{\alpha} p_\alpha x^\alpha, \qquad x \in \mathbb{R}^n,$$

where

$$x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}, \quad \text{with } \sum_{i=1}^n \alpha_i = k,$$

is a monomial of degree k with coefficient p_α. Denote by $p = \{p_\alpha\} \in \mathbb{R}^{s(r)}$ the coefficients of the polynomial $p(x)$ in the basis (2.1). Hence, the respective vectors of coefficients of the polynomials $g_i(x)$, $i = 0, 1, \ldots, m$, in (1.1) are denoted $\{(g_i)_\alpha\} = g_i \in \mathbb{R}^{s(w_i)}$, $i = 0, 1, \ldots, m$, if w_i is the degree of g_i.

We next define the important notions of moment matrix and localizing matrix.

2.1. Moment matrix. Given an $s(2r)$-sequence $(1, y_1, \ldots)$, let $M_r(y)$ be the *moment* matrix of dimension $s(r)$ (denoted $M(r)$ in Curto and Fialkow [2]), with rows and columns labelled by (2.1). For illustration purposes, consider the two-dimensional case. The moment matrix $M_r(y)$ is the block matrix $\{M_{i,j}(y)\}_{0 \le i,j \le r}$ defined by

$$M_{i,j}(y) = \begin{bmatrix} y_{i+j,0} & y_{i+j-1,1} & \cdots & y_{i,j} \\ y_{i+j-1,1} & y_{i+j-2,2} & \cdots & y_{i-1,j+1} \\ \cdots & \cdots & \cdots & \cdots \\ y_{j,i} & y_{i+j-1,1} & \cdots & y_{0,i+j} \end{bmatrix}. \tag{2.2}$$

Thus, with $n = 2$ and $r = 2$, one obtains

$$M_2(y) = \left[\begin{array}{c|cc|cc|ccc} 1 & y_{10} & y_{01} & y_{20} & y_{11} & y_{02} \\ \hline y_{10} & y_{20} & y_{11} & y_{30} & y_{21} & y_{12} \\ y_{01} & y_{11} & y_{02} & y_{21} & y_{12} & y_{03} \\ \hline y_{20} & y_{30} & y_{21} & y_{40} & y_{31} & y_{22} \\ y_{11} & y_{21} & y_{12} & y_{31} & y_{22} & y_{13} \\ y_{02} & y_{12} & y_{03} & y_{22} & y_{13} & y_{04} \end{array}\right].$$

Another more intuitive way of constructing $M_r(y)$ is as follows. If $M_r(y)(1,i) = y_\alpha$ and $M_r(y)(j,1) = y_\beta$, then

$$M_r(y)(i,j) \,=\, y_{\alpha+\beta}, \quad \text{with } \alpha+\beta \,=\, (\alpha_1+\beta_1,\ldots,\alpha_n+\beta_n). \tag{2.3}$$

$M_r(y)$ defines a bilinear form $\langle .,. \rangle_y$, on the space $\mathcal{A}_r$ of real-valued polynomials of degree at most r, by

$$\langle q(x), v(x) \rangle_y \,:=\, \langle q, M_r(y) v \rangle, \quad q(x), v(x) \in \mathcal{A}_r,$$

and if y is a sequence of moments of some measure μ_y, then

$$\langle q, M_r(y) q \rangle \,=\, \int q(x)^2 \, \mu_y(dx) \,\geq\, 0, \tag{2.4}$$

so that $M_r(y) \succeq 0$.

2.2. Localizing matrix. If the entry (i,j) of the matrix $M_r(y)$ is y_β, let $\beta(i,j)$ denote the subscript β of y_β. Next, given a polynomial $\theta(x) : \mathbb{R}^n \to \mathbb{R}$ with coefficient vector θ, we define the matrix $M_r(\theta y)$ by

$$M_r(\theta y)(i,j) \,=\, \sum_\alpha \theta_\alpha y_{\{\beta(i,j)+\alpha\}}. \tag{2.5}$$

For instance, with

$$M_1(y) = \begin{bmatrix} 1 & y_{10} & y_{01} \\ y_{10} & y_{20} & y_{11} \\ y_{01} & y_{11} & y_{02} \end{bmatrix} \text{ and } x \,\mapsto\, \theta(x) \,=\, a - x_1^2 - x_2^2,$$

we obtain

$$M_1(\theta y) = \begin{bmatrix} a - y_{20} - y_{02}, & a y_{10} - y_{30} - y_{12}, & a y_{01} - y_{21} - y_{03} \\ a y_{10} - y_{30} - y_{12}, & a y_{20} - y_{40} - y_{22}, & a y_{11} - y_{31} - y_{13} \\ a y_{01} - y_{21} - y_{03}, & a y_{11} - y_{31} - y_{13}, & a y_{02} - y_{22} - y_{04} \end{bmatrix}.$$

In a manner similar to what we have in (2.4), if y is a sequence of moments of some measure μ_y, then

$$\langle q, M_r(\theta y) q \rangle \,=\, \int \theta(x) q(x)^2 \, \mu_y(dx) \tag{2.6}$$

for every polynomial $q(x) : \mathbb{R}^n \to \mathbb{R}$ with coefficient vector $q \in \mathbb{R}^{s(r)}$. Therefore, $M_r(\theta y) \succeq 0$ whenever μ_y has its support contained in the set $\{\theta(x) \geq 0\}$. In Curto and Fialkow [2], $M_r(\theta y)$ is called a *localizing* matrix (denoted by $M_\theta(r+v)$ if $\deg \theta = 2v$ or $2v-1$).

The theory of *moments* identifies those sequences $y = (y_1, \ldots)$ with $M_r(y) \succeq 0$ that are moment-sequences. The $\mathbb{K}$-moment problem identifies those sequences y that are moment-sequences of a measure with support contained in the semialgebraic set $\mathbb{K}$. In duality with the theory of moments is the theory of representation of positive polynomials, which dates back to Hilbert's 17th problem. For details and recent results, the interested reader is referred to Curto and Fialkow [1, 2], Jacobi [3], Jacobi and Prestel [4], Simon [14], Schmüdgen [11], and the many references therein.

3. Main result. Consider the 0-1 optimization problem $\mathbb{P}$ in (1.1) where the $g_i(x)$ are all real-valued polynomials, $i = 0, \ldots, m$. Let

$$\mathbb{K} := \{x \in \{0,1\}^n \,|\, g_i(x) \geq 0,\ i = 1, \ldots, m\}$$

be the feasible set.

For the sake of simplicity in later proofs, we treat the 0-1 integrality constraints $x_i^2 = x_i$ as two opposite inequalities $g_{m+i}(x) = x_i^2 - x_i \geq 0$ and $g_{m+n+i}(x) = x_i - x_i^2 \geq 0$, and we redefine the set $\mathbb{K}$ to be

$$\mathbb{K} = \{g_i(x) \geq 0,\ i = 1, \ldots, m+2n\}. \tag{3.1}$$

However, in view of the special form of the constraints $g_{m+k}(x) \geq 0$, $k = 1, \ldots, 2n$, we will provide a simpler form of the SDP relaxations $\{\mathbb{Q}_i\}$ below.

Depending on its parity, let $w_k := 2v_k$ or $w_k := 2v_k - 1$ be the degree of the polynomial $g_k(x)$, $k = 1, \ldots, m+2n$. When needed below, for $i \geq \max_k w_k$, the vectors $g_k \in \mathbb{R}^{s(w_k)}$ are extended to vectors of $\mathbb{R}^{s(i)}$ by completing them with zeros. As we minimize g_0, we may and will assume that its constant term is zero, that is, $g_0(0) = 0$.

For optimization purposes, we could use the integrality constraints $x_k^2 = x$, $k = 1, \ldots, n$, to simplify the polynomials g_k, $k = 0, \ldots, m$. However, for the *representation* results of section 3.2, we need to consider the polynomials g_k, $k = 0, \ldots, m$, as given in their original form.

For $i \geq \max_k v_k$, consider the following family $\{\mathbb{Q}_i\}$ of psd programs:

$$\mathbb{Q}_i \begin{cases} \min\limits_y \sum\limits_\alpha (g_0)_\alpha y_\alpha \\ \qquad M_i(y) \succeq 0, \\ \quad M_{i-v_k}(g_k y) \succeq 0, \quad k = 1, \ldots, m+2n, \end{cases} \tag{3.2}$$

with respective dual problems

$$\mathbb{Q}_i^* \begin{cases} \min\limits_{X, Z_k \succeq 0} -X(1,1) - \sum\limits_{k=1}^{m+2n} g_k(0) Z_k(1,1) \\ \qquad \langle X, B_\alpha \rangle + \sum\limits_{k=1}^{m+2n} \langle Z_k, C_\alpha^k \rangle = (g_0)_\alpha \ \forall \alpha \neq 0, \end{cases} \tag{3.3}$$

where we have written

$$M_i(y) = \sum_\alpha B_\alpha y_\alpha, \quad M_{i-v_k}(g_k y) = \sum_\alpha C_\alpha^k y_\alpha, \quad k = 1, \ldots, m+2n,$$

for appropriate real-valued symmetric matrices B_α, C_α^k, $k = 1, \ldots, m+2n$.

Note that the localizing matrices $M_{i-v_k}(g_k y)$ are easily obtained from the data $\{g_k(x)\}$ of the problem by (2.5).

Interpretation of $\mathbb{Q}_i$. The LMI constraints of $\mathbb{Q}_i$ state necessary conditions for y to be the vector of moments up to order $2i$, of some probability measure μ_y with support contained in $\mathbb{K}$. This clearly implies that $\inf \mathbb{Q}_i \leq p^*$, as the vector of moments of the Dirac measure at a feasible point of $\mathbb{P}$ is feasible for $\mathbb{Q}_i$.

Interpretation of $\mathbb{Q}_i^*$. Let $X, Z_k \succeq 0$ be a feasible solution of $\mathbb{Q}_i^*$ with value ρ. Write

$$X = \sum_j t_j t_j' \quad \text{and } Z_k = \sum_j t_{kj} t_{kj}', \quad k = 1, \ldots, m+2n.$$

Then, from the feasibility of (X, Z_k) in $\mathbb{Q}_i^*$, it was shown in Lasserre [7, 8] that

$$g_0(x) - \rho = \sum_j t_j(x)^2 + \sum_{k=1}^{m+2n} g_k(x) \left[\sum_j t_{kj}(x)^2 \right], \tag{3.4}$$

where the polynomials $\{t_j(x)\}$ and $\{t_{kj}(x)\}$ have respective coefficient vectors $\{t_j\}$ and $\{t_{kj}\}$ in the basis (2.1).

As $\rho \le p^*$, $g_0(x) - \rho$ is nonnegative on $\mathbb{K}$ (strictly positive if $\rho < p^*$); one recognizes in (3.4) a decomposition into a weighted sum of squares of the polynomial $g_0(x) - p^*$, strictly positive on $\mathbb{K}$, as in the theory of representation of polynomials, strictly positive on a compact semialgebraic set $\mathbb{K}$ (see, e.g., Schmüdgen [11], Putinar [10], Jacobi [3], Jacobi and Prestel [4]). Indeed, when the set $\mathbb{K}$ has a certain property (satisfied here), the "linear" representation (3.4) holds (see Lasserre [7, 8]).

Hence, both programs $\mathbb{Q}_i$ and $\mathbb{Q}_i^*$ illustrate the duality between the theory of moments and the theory of positive polynomials. Among other results, it has been shown in Lasserre [8, Theorem 3.3] (see also Lasserre [7, Theorem 4.2]) that

$$\inf \mathbb{Q}_i \uparrow p^* \quad \text{as } i \to \infty. \tag{3.5}$$

In view of the construction of the localizing matrices in (2.5) and the form of the polynomials g_k for $k > m$, the constraints $M_{i-1}(g_{m+k}y) \succeq 0$ and $M_{i-1}(g_{m+n+k}y) \succeq 0$ for $k = 1, \ldots, n$ (equivalently, $M_{i-1}(g_{m+k}y) = 0$) simply state that the variable y_α with $\alpha = (\alpha_1, \ldots, \alpha_n)$ can be replaced with the variable y_β with $\beta_i := 1$ whenever $\alpha_i \ge 1$. Therefore, a simpler form of $\mathbb{Q}_i$ is obtained as follows.

Ignore the constraints $M_{i-v_k}(g_k y) \succeq 0$ for $k = m+1, \ldots, m+2n$ corresponding to the integral constraints, and make the above substitution $y_\alpha \to y_\beta$ in the matrices $M_i(y)$ and $M_{i-v_k}(g_k(y))$, $k = 1, \ldots, m$. For instance, in $\mathbb{R}^2$ ($n = 2$), the matrix

$$M_2(y) = \left[\begin{array}{c|cc|cc|ccc}
1 & y_{10} & y_{01} & y_{20} & y_{11} & y_{02} \\
\hline
y_{10} & y_{10} & y_{11} & y_{30} & y_{21} & y_{12} \\
y_{01} & y_{11} & y_{02} & y_{21} & y_{12} & y_{03} \\
\hline
y_{20} & y_{30} & y_{21} & y_{40} & y_{31} & y_{22} \\
y_{11} & y_{21} & y_{12} & y_{3,1} & y_{22} & y_{13} \\
y_{02} & y_{12} & y_{03} & y_{22} & y_{13} & y_{04}
\end{array}\right] \tag{3.6}$$

is replaced with

$$\widehat{M_2}(y) = \left[\begin{array}{c|cc|cc|ccc}
1 & y_{10} & y_{01} & y_{10} & y_{11} & y_{01} \\
\hline
y_{10} & y_{10} & y_{11} & y_{10} & y_{11} & y_{11} \\
y_{01} & y_{11} & y_{01} & y_{11} & y_{11} & y_{01} \\
\hline
y_{10} & y_{10} & y_{11} & y_{10} & y_{11} & y_{11} \\
y_{11} & y_{11} & y_{11} & y_{11} & y_{11} & y_{11} \\
y_{01} & y_{11} & y_{01} & y_{11} & y_{11} & y_{01}
\end{array}\right], \tag{3.7}$$

and only the variables y_{10}, y_{01}, y_{11} appear in all the relaxations $\mathbb{Q}_i$. Interpreted in terms of polynomials, the integrality constraints $x^2 = x$ imply that a monomial $x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$ can be replaced by

$$x_1^{\beta_1} x_2^{\beta_2} \cdots x_n^{\beta_n} \quad \text{with } \beta_i = \begin{cases} 0 \text{ if } \alpha_i = 0, \\ 1 \text{ if } \alpha_i \ge 1. \end{cases} \tag{3.8}$$

Therefore, there are no more than $2^n - 1$ variables y_β (the number of monomials of degree at most n), and the relaxation $\mathbb{Q}_i$ has the simplified form

$$\mathbb{Q}_i \begin{cases} \min\limits_y \sum\limits_\alpha (g_0)_\alpha y_\alpha \\ \qquad \widehat{M}_i(y) \succeq 0, \\ \widehat{M}_{i-v_k}(g_k y) \succeq 0, \quad k = 1, \ldots, m, \end{cases} \tag{3.9}$$

where the LMI constraints associated with the integral constraints are removed and the moment matrix $M_i(y)$ as well as the localizing matrices $M_{i-v_k}(g_k y)$, $k = 1, \ldots, m$, have incorporated the substitutions $y_\alpha \to y_\beta$ indicated above and are denoted $\widehat{M}_i(y)$ and $\widehat{M}_{i-v_k}(g_k y)$.

We begin with the following crucial result.

PROPOSITION 3.1. (a) *All the relaxations $\mathbb{Q}_i$ involve at most $2^n - 1$ variables y_α.*

(b) *For all the relaxations $\mathbb{Q}_i$ with $i > n$, one has*

$$\operatorname{rank} M_i(y) = \operatorname{rank} M_n(y). \tag{3.10}$$

Proof. Part (a) is just a consequence of the comment preceding Proposition 3.1.

To get (b), observe that with $i > n$, one may write

$$M_i(y) = \begin{bmatrix} M_n(y) & | & B \\ - & & - \\ B' & | & C \end{bmatrix}$$

for appropriate matrices B, C, and we next prove that each column of B is identical to some column of $M_n(y)$.

Indeed, remember from (2.3) how an element $M_i(y)(k,p)$ can be obtained. Let $y_\gamma = M_i(y)(k,1)$ and $y_\alpha = M_i(y)(1,p)$. Then

$$M_i(y)(k,p) = y_\eta \quad \text{with } \eta_i = \gamma_i + \alpha_i, \quad i = 1, \ldots, n. \tag{3.11}$$

Now, consider a column B_j of B, that is, some column $M_i(y)(.,p)$ of $M_i(y)$, with first element $y_\alpha = B_j(1) = M_i(y)(1,p)$. Therefore, the element $B(k)$ (or, equivalently, the element $M_i(y)(k,p)$) is the variable y_η in (3.11). Note that α corresponds to a monomial in the basis (2.1) of degree larger than n, say $\alpha_1 \cdots \alpha_n$. Associate to this column B_j the column $v := M_n(y)(.,q)$ of $M_n(y)$, whose element $v(1) = M_n(y)(1,q) = y_\beta$ (for some q) with $\beta_i = 0$ if $\alpha_i = 0$ and 1 otherwise, for all $i = 1, \ldots, n$. Then the element $v(k) = M_n(y)(k,q)$ is obtained as

$$v(k) = y_\delta \quad \text{with } \delta_i = \gamma_i + \beta_i, \quad i = 1, \ldots, n.$$

But then, as for each entry y_α of $M_j(y)$, we can make the substitution $\alpha_i \leftrightarrow 1$ whenever $\alpha_i \geq 1$; it follows that the element $v(k)$ is identical to the element $B(k)$. In other words, each column of B is identical to some column of $M_n(y)$.

If we now write

$$M_j(y) = [A|D] \quad \text{with } A := \begin{bmatrix} M_n(y) \\ - \\ B' \end{bmatrix} \text{ and } D := \begin{bmatrix} B \\ - \\ C \end{bmatrix},$$

then, with exactly same arguments, every column of D is also identical to some column of A, and consequently, (3.10) follows. ∎

For instance, when $n = 2$, the reader can check that $M_3(y)$ (or $\widehat{M}_3(y)$) has the same rank as $M_2(y)$ (or $\widehat{M}_2(y)$) in (3.6) (or in (3.7)). We now can state our main result in the following theorem.

THEOREM 3.2. *Let $\mathbb{P}$ be the problem defined in* (1.1) *and let $v := \max_{k=1,\ldots,m} v_k$. Then for every $i \geq n + v$*

(a) *$\mathbb{Q}_i$ is solvable with $p^* = \min \mathbb{Q}_i$, and to every optimal solution x^* of $\mathbb{P}$ corresponds the optimal solution*

$$y^* := (x_1^*, \ldots, x_n^*, \ldots, (x_1^*)^{2i}, \ldots, (x_n^*)^{2i}) \tag{3.12}$$

of $\mathbb{Q}_i$;

(b) *every optimal solution y^* of $\mathbb{Q}_i$ is the (finite) vector of moments of a probability measure finitely supported on s optimal solutions of $\mathbb{P}$, with $s = \operatorname{rank} M_i(y) = \operatorname{rank} M_n(y)$.*

Proof. Let y be an admissible solution of $\mathbb{Q}_{n+v}$. From Proposition 3.1, we have that $\operatorname{rank} M_i(y) = \operatorname{rank} M_n(y)$ for all $i > n$ (in particular, for $i = n+v$). From a deep result of Curto and Fialkow [2, Theorem 1.6, p. 6], it follows that y is the vector of moments of some $\operatorname{rank} M_n(y)$ atomic measure μ_y. ($M_{n+1}(y)$ is called a *flat positive extension* of $M_n(y)$, and it follows that $M_{n+1}(y)$ admits unique flat extension moment matrices $M_{n+2}(y)$, $M_{n+3}(y)$, etc. (see Curto and Fialkow [2, p. 3]).) Moreover, from the constraints $M_{n+v-v_k}(g_k y) \succeq 0$ for all $k = 1, \ldots, m + 2n$, it also follows that μ_y is supported on $\mathbb{K}$ (see Theorem 1.6 in Curto and Fialkow [2, p. 6], stated in dimension 2 for the complex plane, but also valid for n real or complex variables (see the comments on page 2 in [2])). In the notation of Theorem 1.6 in Curto and Fialkow [2], $M(n)$ is our moment matrix $M_n(y)$, and the localizing matrix $M_{g_k}(n + v_k)$ is our localizing matrix $M_{n+v-v_k}(g_k(y))$ associated with the constraint $g_k(x) \geq 0$.

But then, as μ_y is supported on $\mathbb{K}$, it also follows that

$$\sum_\alpha (g_0)_\alpha y_\alpha \;=\; \int_{\mathbb{K}} g_0(x)\, \mu_y(dx) \;\geq\; p^*.$$

From this and $\inf \mathbb{Q}_i \leq p^*$, statement (a) in Theorem 3.2 follows for $i = n + v$. For $i > n + v$, the result follows from $p^* \geq \inf \mathbb{Q}_{i+1} \geq \inf \mathbb{Q}_i$ for all i. Finally, y^* in (3.12) is obviously admissible for $\mathbb{Q}_i$ with value p^* and therefore is an optimal solution of $\mathbb{Q}_i$.

(b) follows from same arguments as in (a). First observe that from (a), $\mathbb{Q}_i$ is solvable for all $i \geq n + v$. Now, let y^* be an optimal solution of $\mathbb{Q}_i$. From (a), we have that y^* is the vector of moments of an atomic measure μ_{y^*} supported on $s\,(= \operatorname{rank} M_n(y))$ points $x^1, \ldots, x^s \in \mathbb{K}$; that is, with $\delta_{\{.\}}$ the Dirac measure at ".",

$$\mu_{y^*} \;=\; \sum_{j=1}^s \gamma_j \delta_{x_j} \;\text{ with } \gamma_j \geq 0, \quad \sum_{j=1}^s \gamma_j \;=\; 1.$$

Therefore, from $g_0(x^j) \geq p^*$ for all $j = 1, \ldots, s$ and from

$$\begin{aligned} p^* = \min \mathbb{Q}_i \;&=\; \sum_\alpha (g_0)_\alpha y_\alpha^* \\ &= \int_{\mathbb{K}} g_0(x)\, \mu_{y^*}(dx) \;=\; \sum_{j=1}^s \gamma_j g_0(x^j), \end{aligned}$$

it follows that each point x^j must be an optimal solution of $\mathbb{P}$. ∎

Hence, Theorem 3.2 shows that every 0-1 program is equivalent to the psd program $\mathbb{Q}_{n+v}$ with $2^n - 1$ variables. An alternative proof of Theorem 3.2 which does not invoke results of algebraic geometry is proposed in the recent work of Laurent [6], where the relaxations $\{\mathbb{Q}_i\}$ are shown to be stronger than the Sherali and Adams linear relaxations [12]. As the latter converge in finitely many steps, so do the former.

Remark 3.3. As a consequence of Proposition 3.1, in all relaxations $\mathbb{Q}_i$ in (3.9) with $i > n$, one may replace the constraint $\widehat{M}_i(y) \succeq 0$ by $\widehat{M}_n(y) \succeq 0$. Indeed, if $\widehat{M}_n(y) \succeq 0$, then, from the definition of $\widehat{M}_i(y)$ and the proof of Proposition 3.1, it follows that $\widehat{M}_i(y) \succeq 0$ whenever $i > n$.

Moreover, and in the same spirit, at any relaxation $\mathbb{Q}_i$, the matrix $\widehat{M}_i(y)$ can be reduced in size. When looking at the kth column $\widehat{M}_i(y)(.,k)$, if $\widehat{M}_i(y)(1,k) = \widehat{M}_i(y)(p)$ for some $p < k$, then the whole column $\widehat{M}_i(y)(.,k)$ as well as the corresponding line $\widehat{M}_i(y)(k,.)$ can be deleted. For instance, with $\widehat{M}_2(y)$ as in (3.7), the constraint

$$\widehat{M}_2(y) = \left[\begin{array}{c|cc|ccc} 1 & y_{10} & y_{01} & y_{10} & y_{11} & y_{01} \\ - & - & - & - & - & - \\ y_{10} & y_{10} & y_{11} & y_{10} & y_{11} & y_{11} \\ y_{01} & y_{11} & y_{01} & y_{11} & y_{11} & y_{01} \\ - & - & - & - & - & - \\ y_{10} & y_{10} & y_{11} & y_{10} & y_{11} & y_{11} \\ y_{11} & y_{11} & y_{11} & y_{11} & y_{11} & y_{11} \\ y_{01} & y_{11} & y_{01} & y_{11} & y_{11} & y_{01} \end{array}\right] \succeq 0$$

is equivalent to the constraint

$$\left[\begin{array}{c|cc|c} 1 & y_{10} & y_{01} & y_{11} \\ - & - & - & - \\ y_{10} & y_{10} & y_{11} & y_{11} \\ y_{01} & y_{11} & y_{01} & y_{11} \\ - & - & - & - \\ y_{11} & y_{11} & y_{11} & y_{11} \end{array}\right] \succeq 0,$$

for the 4th and 6th columns of $\widehat{M}_2(y)$ are the same as the 2nd and 3rd columns. Thus, in the matrix $\widehat{M}_i(y)$, one retains only the columns (and the rows) corresponding to the monomials in the basis (2.1) that are *distinct* after the simplification $x_i^2 = x_i$. The same simplification occurs for the matrices of the LMI constraints $\widehat{M}_{i-v_k}(g_k y) \succeq 0$. Therefore, in the $\mathbb{Q}_i$ relaxation (3.9), the above simplification of the matrix $\widehat{M}_i(y)$ is an $r \times r$ matrix with $r := \sum_{k=0}^{i} \binom{i}{k} = 2^i$.

Finally, in view of Remark 3.3, the relaxation $\mathbb{Q}_{n+v}$ has the equivalent simpler form

$$\left\{\begin{array}{rcll} \min\limits_y \sum\limits_\alpha (g_0)_\alpha y_\alpha & & & \\ \widehat{M}_n(y) & \succeq & 0, & \\ \widehat{M}_{n+v-v_k}(g_k y) & \succeq & 0, & k = 1, \ldots, m, \end{array}\right. \tag{3.13}$$

since $\widehat{M}_n(y) \succeq 0$ implies $\widehat{M}_{n+v}(y) \succeq 0$.

Interpretation. The interpretation of the relaxation $\mathbb{Q}_{n+v}$ is as follows. The unknown variables $\{y_\alpha\}$ should be interpreted as the moments of some probability measure μ, up to order $n + v$. The LMI constraints $M_{n+v-v_k}(g_k y) \succeq 0$ state that

$$\int g_k(x) q(x)^2 \, d\mu \geq 0 \quad \text{for all polynomials of degree } \leq n + v - v_k \tag{3.14}$$

(as opposed to only $g_k(x) \geq 0$ in the original description). The integrality constraints $x_i^2 = x_i$ are hidden in the special form of the matrix $\widehat{M}_n(y)$. The result of Curto and Fialkow [2] used in the proof of Theorem 3.2 ensures that the support of μ is concentrated on $\{0,1\}^n \cap [\cap_{k=1}^m \{g_k(x) \geq 0\}]$.

Therefore, the projection of the feasible set of $\mathbb{Q}_{n+v}$ onto the subspace spanned by $\{y_{10\ldots0}, \ldots, y_{0\ldots01}\}$ is the convex hull of the original constraint set $\mathbb{K} := \{0,1\}^n \cap [\cap_{k=1}^m \{g_k(x) \geq 0\}]$.

3.1. Examples. For illustration purposes, we provide below the explicit description of the equivalent psd programs for quadratic 0-1 programs and MAX-CUT programs in $\mathbb{R}^3$, respectively.

Quadratic 0-1 programs. Consider the quadratic program $\min\{x'Ax \,|\, x \in \{0,1\}^3\}$ for some real-valued symmetric matrix $A \in \mathbb{R}^{3\times3}$. As the only constraints are the integral constraints $x \in \{0,1\}^n$, this is equivalent to the psd program

$$\min A_{11}y_{100} + A_{22}y_{010} + A_{33}y_{001} + A_{12}y_{110} + A_{13}y_{101} + A_{23}y_{011}$$

$$\widehat{M}_3(y) = \begin{bmatrix} 1 & y_{100} & y_{010} & y_{001} & y_{110} & y_{101} & y_{011} & y_{111} \\ y_{100} & y_{100} & y_{110} & y_{101} & y_{110} & y_{101} & y_{111} & y_{111} \\ y_{010} & y_{110} & y_{010} & y_{011} & y_{110} & y_{111} & y_{011} & y_{111} \\ y_{001} & y_{101} & y_{011} & y_{001} & y_{111} & y_{101} & y_{011} & y_{111} \\ y_{110} & y_{110} & y_{110} & y_{111} & y_{110} & y_{111} & y_{111} & y_{111} \\ y_{101} & y_{101} & y_{111} & y_{101} & y_{111} & y_{101} & y_{111} & y_{111} \\ y_{011} & y_{111} & y_{011} & y_{011} & y_{111} & y_{111} & y_{011} & y_{111} \\ y_{111} & y_{111} & y_{111} & y_{111} & y_{111} & y_{111} & y_{111} & y_{111} \end{bmatrix} \succeq 0,$$

where we have used Remark 3.3 and the simplified form of $\widehat{M}_3(y)$.

MAX-CUT. In this case, the criterion is $g_0(x) := x'Ax$ for some real-valued symmetric matrix $A = \{A_{ij}\} \in \mathbb{R}^{3\times3}$ with zeros on the diagonal, and the constraint set is $\{-1,1\}^3$. The fact that we look for solutions in $\{-1,1\}^n$ instead of in $\{0,1\}^n$ leads to obvious modifications in the relaxations $\{\mathbb{Q}_i\}$. The integral constraints are now $x_i^2 = 1$ for all $i = 1, \ldots, n$. Therefore, in a relaxation $\mathbb{Q}_i$, the analogue of what we did for 0-1 programs in order to obtain the matrices $\widehat{M}_i(y)$ is as follows. In the matrix $M_i(y)$, replace any entry y_α by y_β with $\beta_i = 1$ whenever α_i is odd and $\beta_i = 0$ otherwise. Remark 3.3 on the simplified form of $\widehat{M}_i(y)$ and the fact that $\widehat{M}_i(y) \succeq 0$ can be replaced with $\widehat{M}_n(y) \succeq 0$ whenever $i > n$ are also valid.

Hence, similarly to quadratic 0-1 programs, MAX-CUT in $\mathbb{R}^3$ is equivalent to the psd program

$$\min A_{12}y_{110} + A_{13}y_{101} + A_{23}y_{011}$$

$$\widehat{M}_3(y) = \begin{bmatrix} 1 & y_{100} & y_{010} & y_{001} & y_{110} & y_{101} & y_{011} & y_{111} \\ y_{100} & 1 & y_{110} & y_{101} & y_{010} & y_{001} & y_{111} & y_{011} \\ y_{010} & y_{110} & 1 & y_{011} & y_{100} & y_{111} & y_{001} & y_{101} \\ y_{001} & y_{101} & y_{011} & 1 & y_{111} & y_{100} & y_{010} & y_{110} \\ y_{110} & y_{010} & y_{100} & y_{111} & 1 & y_{011} & y_{101} & y_{001} \\ y_{101} & y_{001} & y_{111} & y_{100} & y_{011} & 1 & y_{110} & y_{010} \\ y_{011} & y_{111} & y_{001} & y_{010} & y_{101} & y_{110} & 1 & y_{100} \\ y_{111} & y_{011} & y_{101} & y_{110} & y_{001} & y_{010} & y_{100} & 1 \end{bmatrix} \succeq 0,$$

where we have also used the simplified form of $\widehat{M}_3(y)$.

When solving randomly generated MAX-CUT problems in $\mathbb{R}^3$ with the Matlab LMI toolbox, we obtained optimal solutions y of the form

$$y = \begin{bmatrix} y_{100} \\ y_{010} \\ y_{001} \\ y_{110} \\ y_{101} \\ y_{011} \\ y_{111} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \pm 1 \\ \pm 1 \\ \pm 1 \\ 0 \end{bmatrix},$$

because, as there are always two optimal solutions x^* and $-x^*$, the optimal solution y we obtained is the moment of a probability measure that gives weights of $1/2$ and $1/2$ to x^* and $-x^*$, respectively. That is why the first moments y_{100}, y_{010}, and y_{001} vanish, as well as the third moment y_{111}. Adding a term like, say, ϵy_{100} into the criterion with ϵ very small permits one to recover the optimal solution x^* or $-x^*$ that minimizes ϵx_1. Thus, a MAX-CUT problem in $\mathbb{R}^n$ is equivalent to a psd program with $2^n - 1$ variables and a single LMI constraint of size $2^n \times 2^n$.

3.2. A duality result. We also have the following *representation* result.

PROPOSITION 3.4. *Assume that there is no duality gap between the primal psd program $\mathbb{Q}_{n+v}$ and its dual $\mathbb{Q}^*_{n+v}$, and that $\mathbb{Q}^*_{n+v}$ is solvable. Then*

$$g_0(x) - p^* = \sum_{j=1}^{r_0} q_j(x)^2 + \sum_{k=1}^{m} g_k(x) \left[\sum_{j=1}^{r_k} q_{kj}(x)^2 \right] + \sum_{l=1}^{n} (x_l^2 - x_l) \left[\sum_{j=1}^{s_n} (v_{lj}(x)^2 - w_{lj}(x)^2) \right] \tag{3.15}$$

for some polynomials $\{q_j(x)\}$ of degree at most $n+v$, some polynomials $\{q_{kj}(x)\}$ of degree at most $n+v-v_k$, and some polynomials $\{v_{lj}(x), w_{lj}(x)\}$ of degree at most $n+v-1$.

Proof. The proof follows from the interpretation of the dual psd programs $\{\mathbb{Q}^*_i\}$ in terms of the representation of polynomials that are strictly positive on a compact semialgebraic set, here the set $\mathbb{K} := \{0,1\}^n \cap [\cap_{k=1}^m \{g_k(x) \geq 0\}]$ (see Lasserre [7, 8]). □

Proposition 3.4 states that $g_0(x) - p^*$, which is nonnegative on $\mathbb{K}$, can be written as a sum of squares of polynomials of degree at most $n+v$, weighted by the polynomials $g_k(x)$ defining the constraint set $\mathbb{K}$.

Even though we did not prove the absence of a duality gap, we believe that this is the reason behind the finite termination of the relaxation procedure at step n.

For unconstrained 0-1 programs we even obtain the following.

PROPOSITION 3.5. *Let $g_0(x) : \mathbb{R}^n \to \mathbb{R}$ be an arbitrary polynomial of degree at most $2(n+1)$. Then, with $p^* := \min_{x \in \{0,1\}^n} g_0(x)$,*

$$g_0(x) - p^* = \sum_j q_j(x)^2 + \sum_{k=1}^{n} (x_k^2 - x_k) \left[\sum_j u_{kj}(x)^2 - v_{kj}(x)^2 \right] \tag{3.16}$$

for some polynomials $\{q_j(x), u_{kj}(x), v_{kj}(x)\}$ of degree at most $n+1$.

Similarly, with $p^* := \min_{x\in\{-1,1\}^n} g_0(x)$,

$$g_0(x) - p^* \;=\; \sum_j q_j(x)^2 + \sum_{k=1}^n (x_k^2 - 1)\left[\sum_j u_{kj}(x)^2 - v_{kj}(x)^2\right] \tag{3.17}$$

for some polynomials $\{q_j(x), u_{kj}(x), v_{kj}(x)\}$ *of degree at most* $n+1$.

Proof. Consider the unconstrained problem $\mathbb{P}$ in $\{0,1\}^n$. In this case, from Theorem 3.2, $\mathbb{Q}_{n+1}$ provides the optimal value p^* (as $v=1$). We have seen that in $\mathbb{Q}_{n+1}$, one may replace $M_{n+1}(y) \succeq 0$ by the constraint $\widehat{M}_n(y) \succeq 0$ (see Remark 3.3 and (3.13)); that is, $\mathbb{Q}_{n+1}$ is equivalent to solving the psd program

$$\left\{\begin{array}{ll}\min\limits_y \sum\limits_\alpha (g_0)_\alpha y_\alpha & \\ \qquad\qquad \widehat{M}_n(y) & \succeq\ 0.\end{array}\right. \tag{3.18}$$

Next, to each (of the 2^n) admissible solution $x\in\{0,1\}^n$ of $\mathbb{P}$ corresponds a vector

$$(x_1,\ldots,x_n,x_1x_2,\ldots,x_1^n,\ldots,x_n^n) \;=:\; y \in \mathbb{R}^{2^n-1}$$

that is admissible for $\mathbb{Q}_{n+1}$ (the psd program (3.18)). Label as $y^{(k)}$, $k=1,\ldots,2^n$, every such solution. Let $z := \sum_{k=1}^{2^n} \alpha_k y^{(k)}$ with $\alpha_k > 0$ and $\sum_k \alpha_k = 1$. Clearly,

$$\widehat{M}_n(z) \;=\; \sum_{k=1}^{2^n} \alpha_k \widehat{M}_n(y^{(k)}) \;\succeq\; 0,$$

so that z is admissible for $\mathbb{Q}_{n+1}$. In addition, from the definition of the moment matrix $M_n(y)$ it follows that $M_n(y^{(k)})$ is the rank-one matrix $(1,y^{(k)})(1,y^{(k)})'$, and thus $\widehat{M}(y^{(k)})$ is also a rank-one matrix. Moreover, the vectors $(1,y^{(k)})$, $k=1,\ldots,2^n$, are obviously linearly independent. Therefore, as $\widehat{M}_n(z)$ is the $2^n\times 2^n$ matrix $\sum_{k=1}^{2^n}\alpha_k\widehat{M}_n(y^{(k)})$, it follows that $\widehat{M}_n(z) \succ 0$. Hence, the psd program (3.18) satisfies Slater's interior point condition. As $p^* > -\infty$, from a standard strong duality result in convex optimization (see, e.g., Sturm [15, Theorem 2.24]), the dual is solvable and there is no duality gap. But if we remember that (3.18) has the equivalent form

$$\left\{\begin{array}{lll}\min\limits_y \sum\limits_\alpha (g_0)_\alpha y_\alpha & & \\ \qquad M_{n+1}(y) & \succeq & 0, \\ \qquad M_n(g_k y) & = & 0,\ k=1,\ldots,n\end{array}\right. \tag{3.19}$$

(with $g_k(x) = x_k^2 - x_k$), the result follows from the interpretation of an optimal solution of the dual psd program (see Lasserre [7]). ∎

Hence, the ability to solve exactly an unconstrained 0-1 polynomial program or a MAX-CUT problem at some relaxation $\mathbb{Q}_i$ with $i<n$ depends on whether or not the representation (3.16) (or (3.17)) can hold with polynomials of degree less than i.

Despite its theoretical interest, Theorem 3.2 is of little value for computational purposes, because the number of variables is exponential in the size of the problem. However, in many cases, low order relaxations (that is, with $i \ll n$) will provide the optimal value p^*. Therefore, one would like to have a test to detect whether some

relaxation $\mathbb{Q}_i$ achieves the optimal value p^*. One way is to determine by inspection whether an optimal solution y of $\mathbb{Q}_i$ is a moment vector. This will be the case if, for instance, $\operatorname{rank} M_r(y) = 1$. However, in the case in which $\mathbb{P}$ has multiple optimal solutions (as in MAX-CUT problems), it can happen that y is a convex combination of moments of Dirac measures supported on the optimal solutions, which in general is not easy to detect.

We next provide a criterion to test whether the SDP relaxation $\mathbb{Q}_i$ indeed achieves the optimal value p^*.

THEOREM 3.6. *Let $\mathbb{P}$ be the problem defined in* (1.1) *and let $v := \max_{k=1,\ldots,m} v_k$. Let y^* be an optimal solution of $\mathbb{Q}_i$ with $i < n + v$. If*

$$\operatorname{rank} M_{i-v+1}(y^*) = \operatorname{rank} M_{i-v}(y^*), \tag{3.20}$$

then $\min \mathbb{Q}_i = p^*$ and y^* is the vector of moments of a probability measure supported on $s = \operatorname{rank} M_i(y^*) = \operatorname{rank} M_{i-v}(y^*)$ optimal solutions of $\mathbb{P}$.

Proof. The proof mimics that of Theorem 3.2. By the flat extension theorem (see Curto and Fialkow [2]), it follows that $\operatorname{rank} M_{i-v+k}(y^*) = \operatorname{rank} M_{i-v}(y^*)$ for all $k \geq 1$. It then suffices to apply Theorem 1.1 of Curto and Fialkow [2], which states that there exists a $\operatorname{rank} M_{i-v}(y^*)$ atomic measure μ_{y^*} with moment vector y^*. Moreover, from the constraints $M_{i-v_k}(g_k y^*) \succeq 0$ for all $k = 1, \ldots, m + 2n$, it follows that μ_{y^*} is supported on $\mathbb{K}$, and the result follows as in the proof of Theorem 3.2. Again, in the notation of Theorem 1.6 in Curto and Fialkow [2], we have $M(i-v) \succeq 0$, and $M(i-v)$ has a flat positive extension $M(i-v+1)$ (hence unique flat positive extensions $M(i-v+k)$ for all $k \geq 1$), with $M_{g_k}(i-v+v_k) \succeq 0$ for all $k = 1, \ldots, m + 2n$. □

To illustrate the power of the SDP relaxations $\mathbb{Q}_i$, we have run a sample of 50 MAX-CUT problems in $\mathbb{R}^{10}$ ($\min_{x \in \{-1,1\}^n} x'Qx$), where Q is a symmetric matrix with a null diagonal and with nondiagonal entries randomly generated, uniformly between 0 and 1. (In some examples, zeros and negative entries were allowed.) In all cases, the $\mathbb{Q}_2$ SDP relaxation provided the optimal value. The corresponding psd program has one LMI constraint ($\widehat{M_2}(y) \succeq 0$) of dimension 56×56 and has 385 variables y_β.

4. Conclusion. We have provided an equivalent continuous psd program for arbitrary constrained nonlinear 0-1 programs. For practical computation, it appears that in some (many?) cases, an SDP relaxation of low order suffices. However, for large or even moderate size problems, the resulting SDP relaxations $\mathbb{Q}_i$ might still be too large for the present status of SDP programming packages, as soon as i is larger than, say, 3. A topic of further research is to test the efficiency of the above SDP relaxations when compared to the different LP-based relaxations in the literature, notably the Sherali and Adams RLT technique [12, 13].

REFERENCES

[1] R.E. CURTO AND L.A. FIALKOW, *Recursiveness, positivity, and truncated moment problems*, Houston J. Math., 17 (1991), pp. 603–635.

[2] R.E. CURTO AND L.A. FIALKOW, *The truncated complex K-moment problem*, Trans. Amer. Math. Soc., 352 (2000), pp. 2825–2855.

[3] T. JACOBI, *A representation theorem for certain partially ordered commutative rings*, Math. Z., 237 (2001), pp. 259–273.

[4] T. JACOBI AND A. PRESTEL, *Distinguished representations of strictly positive polynomials*, J. Reine Angew. Math., 532 (2001), pp. 223–235.

[5] M. KOJIMA AND L. TUNÇEL, *Cones of matrices and successive convex relaxations of nonconvex sets*, SIAM J. Optim., 10 (2000), pp. 750–778.

[6] M. Laurent, *A Comparison of the Sherali-Adams, Lovász-Schrijver and Lasserre Relaxations for* 0-1 *Programming*, Technical report PNA-R0108, CWI, Amsterdam, The Netherlands, 2001.

[7] J.B. Lasserre, *Global optimization with polynomials and the problem of moments*, SIAM J. Optim., 11 (2001), pp. 796–817.

[8] J.B. Lasserre, *Optimality Conditions and LMI Relaxations for* 0-1 *Programs*, Technical report #00099, LAAS-CNRS, Toulouse, France, 2000, submitted.

[9] L. Lovász and A. Schrijver, *Cones of matrices and set-functions and* 0-1 *optimization*, SIAM J. Optim., 1 (1991), pp. 166–190.

[10] M. Putinar, *Positive polynomials on compact semi-algebraic sets*, Indiana Univ. Math. J., 42 (1993), pp. 969–984.

[11] K. Schmüdgen, *The K-moment problem for compact semi-algebraic sets*, Math. Ann., 289 (1991), pp. 203–206.

[12] H.D. Sherali and W.P. Adams, *A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems*, SIAM J. Discrete Math., 3 (1990), pp. 411–430.

[13] H.D. Sherali and W.P. Adams, *A hierarchy of relaxations and convex hull characterizations for mixed-integer zero-one programming problems*, Discrete Appl. Math., 52 (1994), pp. 83–106.

[14] B. Simon, *The classical moment problem as a self-adjoint finite difference operator*, Adv. Math., 137 (1998), pp. 82–203

[15] J.F. Sturm, *Theory and algorithms of semidefinite programming*, in High Performance Optimization Methods, H. Frenk, K. Roos, and S. Zhang, eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000, pp. 3–20.

SIAM J. OPTIM.
Vol. 12, No. 3, pp. 770–781

SELF-CONCORDANT BARRIERS FOR CONES GENERATED BY CHEBYSHEV SYSTEMS*

LEONID FAYBUSOVICH†

Abstract. We explicitly calculate characteristic functions of cones of generalized polynomials corresponding to Chebyshev systems on intervals of the real line and the circle. Thus, in principle, we calculate homogeneous self-concordant barriers for this class of cones. This class includes almost all "cones of squares" considered in [Y. Nesterov, *High Performance Optimization*, Kluwer, Dordrecht, 2000, pp. 441–466]. Our construction, however, is applicable to a much broader class of cones.

Key words. interior-point algorithms, characteristic functions of convex cones, T-systems

AMS subject classifications. 90C51, 90C34

PII. S1052623401386782

1. Introduction.

To apply a modern interior-point technique as it is developed in [6], it is necessary to know a self-concordant barrier for a feasible domain of a given convex optimization problem. Given a convex domain in a finite-dimensional vector space, there exists an explicit formula for at least one such barrier, the so-called universal barrier function [6]. For example, let K be a closed convex pointed cone in $\mathbf{R^n}$ with a nonempty interior. Consider

$$\Phi(p) = \ln \int_{K^*} e^{-\langle c,p\rangle} d\mu(c), \tag{1.1}$$

where $p \in \mathrm{int}(K)$, K^* is the cone dual to K, and μ is the standard Lebesgue measure on $\mathbf{R^n}$. Then Φ after an appropriate normalization is the so-called homogeneous self-concordant barrier function. The knowledge of such a function in a "computable" form enables one, in principle, to develop interior-point algorithms (along with complexity estimates) for optimization problems whose feasibility domain is the intersection of K with an affine subspace in $\mathbf{R^n}$ and for many other related problems (through the barrier calculus).

Unfortunately, expression (1.1) requires the evaluation of multidimensional integrals over geometrically complicated domains for the computation of the value of Φ, its gradient, and the Hessian at a given point $p \in \mathrm{int(K)}$. This is, in general, computationally too expensive, taking into account the original task in question, i.e., solving a convex optimization problem.

There are a number of situations in which (1.1) can be more or less explicitly calculated. Several of the corresponding cones belong to the class of symmetric cones, and (1.1) is then easily expressed in terms of the attached Jordan algebra (see, e.g., [3]). A part of the theory of interior-point algorithms admits an infinite-dimensional generalization [7], but the concept of the universal barrier function seems to be essentially finite-dimensional.

In the present paper we significantly expand the class of cones for which (1.1) can be explicitly calculated. Correspondingly, we expand the class of optimization

*Received by the editors March 26, 2001; accepted for publication (in revised form) September 13, 2001; published electronically February 27, 2002. This work was supported in part by NSF grant DMS98-03191.

http://www.siam.org/journals/siopt/12-3/38678.html

†Department of Mathematics, University of Notre Dame, Notre Dame, IN 46556 (leonid.faybusovich.1@nd.edu).

problems to which the modern interior-point technique can be applied. Namely, we consider cones of generalized nonnegative polynomials generated by Chebyshev systems on the intervals of the real line or the unit circle. For such cones we find more or less explicit expressions for (1.1) only slightly more complicated (in a computational sense) than those for symmetric cones. In particular, practically all cones considered by Nesterov in [5] can be treated from our viewpoint. Note, however, that the representation of a given cone as a "cone of squares" (and hence the reducibility of a given problem to semidefinite programming), which is crucial for Nesterov's construction, does not play any role in our approach. Thus, our results are applicable to a broader class of cones. The calculation of (1.1) is new even for most of the cones considered in [5].

2. Chebyshev systems. We start with several examples of Chebyshev systems. We then formulate several important properties of such systems.

DEFINITION 1. *A system of real functions $u_0, \dots, u_n$ defined on an abstract set E is called a Chebyshev system (T-system) of order n on E if the determinant*

$$\det(u_i(t_j)),$$

$i, j = 0, 1, \dots, n$, *does not vanish for any pairwise distinct $t_0, \dots, t_n \in E$.*

If the set E is endowed with a topology, one usually assumes that the functions $u_0, \dots, u_n$ are continuous on E.

In this paper we are mostly interested in the cases in which $E = [a, b] \subset \mathbf{R}$ or $E = \mathbf{S^1}$ (unit circle). In the latter case, $\mathbf{S^1}$ may be viewed as an interval $[a, b]$ with identified endpoints. A T-system on a circle is a T-system of functions on $[a, b)$ with the additional property that $u_k(a) = u_k(b), k = 0, 1, \dots, n$.

Consider several examples of T-systems.

Example 1. Let $u_i(t) = t^i, i = 0, 1, \dots, n, t \in [a, b]$. This is a T-system, as it easily follows from the properties of the Vandermonde determinant.

Example 2. The functions

$$\frac{1}{t+\alpha_0}, \frac{1}{t+\alpha_1}, \dots, \frac{1}{t+\alpha_n},$$

$0 < \alpha_0 < \alpha_1 < \cdots < \alpha_n$, form a T-system on any interval $[a, b]$ such that $a + \alpha_0 > 0$.

Example 3. The functions

$$\exp(\alpha_0 t), \exp(\alpha_1 t), \dots, \exp(\alpha_n t)$$

form a T-system on any interval $[a, b]$.

Example 4. If $f : [a, b] \to \mathbf{R}$ is an n times continuously differentiable function on $[a, b]$ such that $f^{(n)}(t) > 0, t \in [a, b]$, then the functions

$$1, t, \dots, t^{n-1}, f(t)$$

form a T-system on $[a, b]$.

Example 5. The functions

$$1, \sin t, \dots, \sin(nt), \cos t, \dots, \cos(nt)$$

form a periodic T-system on $[0, 2\pi]$ of the order $2n$.

One can show that every periodic T-system has an even order. For a detailed discussion of the examples given above and many more examples, see, e.g., [4].

Given a T-system $u_0, \dots, u_n$ on the interval $[a, b]$, consider the cone K of nonnegative generalized polynomials associated with this system:

$$K = \left\{ p = \sum_{i=0}^{n} a_i u_i : p(t) \geq 0 \quad \forall t \in [a, b] \right\}.$$

We can associate with K the dual cone

$$K^* = \left\{ (c_0, \dots, c_n)^T \in \mathbf{R^{n+1}} : \sum_{i=0}^{n} c_i a_i \geq 0 \quad \forall p = \sum_{i=0}^{n} a_i u_i \in K \right\}.$$

THEOREM 1. *We have*

$$\operatorname{int}(K) = \{p \in K : p(t) > 0 \quad \forall t \in [a, b]\} \neq \emptyset.$$

The vector $(c_0, \dots, c_n)^T \in K^*$ *if and only if there exists a Borel measure* σ *on* $[a, b]$ *such that*

$$c_i = \int_a^b u_i(t) d\sigma(t), \qquad i = 0, 1, \dots, n. \tag{2.1}$$

For a proof of Theorem 1, see, e.g., [4].

If in the representation (2.1) the corresponding measure σ is concentrated in a finite number of points

$$a \leq \xi_1 < \xi_2 < \cdots < \xi_m \leq b,$$

then (2.1) takes the form:

$$c_i = \sum_{j=0}^{m} \rho_j u_i(\xi_j), \qquad i = 0, 1, \dots, m, \tag{2.2}$$

$\rho_j > 0$. Following [4], the points ξ_j involved in the representation (2.2) will be called the roots, and the coefficients ρ_j will be called the weights. We further introduce the notation $\epsilon(t), a \leq t \leq b$, where $\epsilon(t) = 2, a < t < b, \epsilon(a) = \epsilon(b) = 1$. The sum

$$\sum_{j=1}^{m} \epsilon(\xi_j)$$

will be called the index of the representation (2.2). A representation (2.2) is called *principal* if its index is equal to $n + 1$, where n is the order of the Chebyshev system $u_0, \dots, u_n$. Consider the possible types of principal representations. If $n = 2\nu - 1, \nu = 1, 2, \dots$, then either all $\xi_j \in (a, b), m = \nu$, or $\xi_j \in (a, b), j = 2, 3, \dots, \nu$, $\xi_1 = a, \xi_{\nu+1} = b, m = \nu + 1$. In the former case the corresponding representation (2.2) is called the lower principal representation, and in the latter case the representation (2.2) is called the upper principal representation. If $n = 2\nu$, then either ν roots $\xi_j, j = 2, 3, \dots, \nu+1$, belong to (a, b) and $\xi_1 = a, m = \nu + 1$, or ν roots $\xi_j, j = 1, 2, \dots, \nu$, belong to (a, b) and $\xi_{\nu+1} = b, m = \nu + 1$. In the former case the representation (2.2) is called the lower principal representation, and in the latter case the representation (2.2) is called the upper principal representation. Thus, a principal representation is upper or lower

according to whether it has or has not a root at the right endpoint b of the interval $[a, b]$.

THEOREM 2. *Given a T-system $u_0, \dots, u_n$ on the interval $[a, b]$, each point $c \in \operatorname{int}(K^*)$ (see (2.1)) has exactly one lower principal representation and exactly one upper principal representation.*

This result admits the following modification for the case of a periodic T-system on the interval $[a, b), n = 2\nu$.

THEOREM 3. *Each point $c \in \operatorname{int}(K^*)$ admits a unique representation (2.2) with $m = \nu + 1$, one of whose roots $\xi_1, \dots, \xi_{\nu+1}$ is a prescribed point $\xi \in [a, b)$.*

For a proof of Theorems 2 and 3, see, e.g., [4].

3. Calculation of characteristic functions. We now use principal representations of elements of K^* to calculate the characteristic function of the cone K generated by a Chebyshev system $u_0, \dots, u_n$. We assume that $u_0, \dots, u_n$ are continuously differentiable functions on the interval $[a, b]$.

Let us start with the case $n = 2\nu - 1$. Given $p \in K$, we wish to calculate

$$F(p) = \int_{K^*} e^{-\langle c,p\rangle} d\mu(c), \tag{3.1}$$

where μ is the standard Lebesgue measure on $\mathbf{R^{n+1}}$. We use the lower principal representation (2.2) to parametrize $\operatorname{int}(K^*)$:

$$c_i = \sum_{j=1}^{\nu} \rho_j u_j(\xi_i), \tag{3.2}$$

$i = 0, 1, \dots, 2\nu - 1$. According to Theorem 2, the map (3.2) gives a one-to-one correspondence between

$$\mathbf{R}_+^\nu \times \{\xi \in \mathbf{R}^\nu : a < \xi_1 < \xi_2 < \cdots < \xi_\nu < b\}$$

and $\operatorname{int}(K^*)$. Here $\mathbf{R}_+ = \{x \in \mathbf{R} : x > 0\}$. We will denote this map by $\Phi = \Phi(\rho_1, \dots, \rho_\nu, \xi_1, \dots, \xi_\nu)$. We obviously have

$$\frac{\partial \Phi}{\partial \rho_j} = u(\xi_j),$$

$j = 1, 2, \dots, \nu$, where

$$u(\xi_j) = (u_0(\xi_j), \dots, u_{2\nu-1}(\xi_j))^T \in \mathbf{R^{2\nu}},$$

$$\frac{\partial \Phi}{\partial \xi_j} = \rho_j u'(\xi_j),$$

$j = 1, 2, \dots, \nu$. Thus, the Jacobian of this map is equal to

$$| \det(u(\xi_1), \dots, u(\xi_\nu), \rho_1 u'(\xi_1), \dots, \rho_\nu u'(\xi_\nu)) |$$

$$= \left(\prod_{k=1}^{\nu} \rho_k \right) | \det(u(\xi_1), u'(\xi_1), \dots, u(\xi_\nu), u'(\xi_\nu)) | .$$

Making the change of variables in (3.1) and using the Fubini theorem, we obtain

$$F(p) = \int_{a\le\xi_1<\xi_2,\dots,\xi_\nu\le b} |\det(u(\xi_1), u'(\xi_1), \dots, u(\xi_\nu), u'(\xi_\nu))|$$

$$\times \left(\int_{\mathbf{R}^+} \rho_1 e^{-p(\xi_1)\rho_1} d\rho_1 \cdots \int_{\mathbf{R}_+} \rho_\nu e^{-p(\xi_\nu)\rho_\nu} d\rho_\nu\right) d\xi_1 \dots d\xi_\nu.$$

Here we used the observation that if $p = a_0u_0 + \cdots + a_nu_n$, then

$$\begin{aligned}\langle c, p\rangle &= \sum_{i=0}^{n} c_i a_i = \sum_{i=0}^{n}\sum_{j=1}^{\nu} \rho_j u_i(\xi_j) a_i \\ &= \sum_{j=1}^{\nu} \rho_j \sum_{i=0}^{n} a_i u_i(\xi_j) = \sum_{j=1}^{\nu} \rho_j p(\xi_j).\end{aligned}$$

Let

$$V(\xi_1, \xi_2, \dots, \xi_\nu) = \det(u(\xi_1), u'(\xi_1), \dots, u(\xi_\nu), u'(\xi_\nu)).$$

Since

$$\int_0^{+\infty} xe^{-\alpha x} dx = \frac{1}{\alpha^2}, \qquad \alpha > 0,$$

we obtain

$$F(p) = \int_{a<\xi_1<\cdots<\xi_\nu<b} \left(\prod_{j=1}^{\nu} \frac{1}{p(\xi_j)^2}\right) |V(\xi_1, \dots, \xi_\nu)| \, d\xi_1 \dots d\xi_\nu. \tag{3.3}$$

Observe that the function under the integral sign in (3.3) is symmetric with respect to variables $\xi_1, \dots, \xi_\nu$. Hence,

$$F(p) = \frac{1}{\nu!} \int_a^b \cdots \int_a^b \left(\prod_{j=1}^{\nu} \frac{1}{p(\xi_j)^2}\right) |V(\xi_1, \dots, \xi_\nu)| \, d\xi_1 \dots d\xi_\nu.$$

LEMMA 1. *The function $V(\xi_1, \dots, \xi_\nu)$ does not change sign on $[a, b]^\nu$.*

Proof. Consider, first, the case in which $a < \xi_1 < \xi_2 < \cdots < \xi_\nu < b$. Let $\eta_1, \dots, \eta_\nu$ be such that

$$\xi_1 < \eta_1 < \xi_2 < \eta_2 \dots \xi_\nu < \eta_\nu \le b. \tag{3.4}$$

Since $u_0, \dots, u_{2\nu-1}$ is a T-system, we can assume without loss of generality that

$$\gamma(\xi_1, \dots, \xi_\nu, \eta_1, \dots, \eta_\nu) := \det[u(\xi_1), u(\eta_1), u(\xi_2), u(\eta_2), \dots, u(\xi_\nu), u(\eta_\nu)] > 0$$

for any ξ, η_i satisfying (3.4). By the mean value theorem we have

$$\begin{aligned}&\gamma(\xi_1, \dots, \xi_\nu, \eta_1, \dots, \eta_\nu) \\ &= \prod_{i=1}^{\nu} (\eta_i - \xi_i) \det[u(\xi_1), u'(\theta_1), u(\xi_2), u'(\theta_2), \dots, u(\xi_\nu), u'(\theta_\nu)] > 0\end{aligned}$$

for some $\xi_i < \theta_i < \eta_i, i = 1, 2, \ldots, \nu$. Hence,

$$\det[u(\xi_1), u'(\theta_1), u(\xi_2), u'(\theta_2), \ldots, u(\xi_\nu), u'(\theta_\nu)] > 0.$$

Taking the limit when $\eta_i \to \xi_i, i = 1, 2, \ldots, \nu$, we obtain

$$V(\xi_1, \ldots, \xi_\nu) \geq 0$$

for all $a < \xi_1 < \xi_2 < \cdots < \xi_\nu < b$. Using the continuity of V, we obtain

$$V(\xi_1, \ldots, \xi_\nu) \geq 0$$

for $a \leq \xi_1 \leq \xi_2 \leq \xi_\nu \leq b$. Our final observation is that

$$V(\xi_{\sigma(1)}, \xi_{\sigma(2)}, \ldots, \xi_{\sigma(\nu)}) = V(\xi_1, \ldots, \xi_\nu)$$

for any permutation σ of the set $\{1, 2, \ldots, \nu\}$. Hence, $V(\xi_1, \xi_2, \ldots, \xi_\nu) \geq 0$ for all ξ_i satisfying $a \leq \xi_i \leq b, i = 1, 2, \ldots, \nu$.

Using Lemma 1, we obtain

$$F(p) = \frac{\epsilon}{\nu!} \int_a^b \cdots \int_a^b \det[\tilde{u}(\xi_1), \tilde{u}'(\xi_1), \ldots, \tilde{u}(\xi_\nu), \tilde{u}'(\xi_\nu)] d\xi_1 \ldots d\xi_\nu, \tag{3.5}$$

where

$$\tilde{u}(\xi) = \frac{u(\xi)}{p(\xi)}, \qquad \epsilon = \pm 1.$$

The next proposition, which comes from de Bruijn (see [1]), is crucial for the evaluation of the characteristic function. Recall the definition of the Pfaffian of an even-dimensional skew-symmetric matrix (see, e.g., [1]):

$$Pf(B) = \frac{1}{n! 2^n} \sum_{\sigma \in \Sigma(2n)} (-1)^{sign\sigma} b_{\sigma(1)\sigma(2)} b_{\sigma(3)\sigma(4)} \cdots b_{\sigma(2n-1)\sigma(2n)}.$$

Here the summation is taken over the set of all permutations of the set $[1, 2n]$.

PROPOSITION 1. *Let (X, μ) be a measurable space with a finite positive measure μ on X. Suppose that $\psi_1, \ldots, \psi_{2n}, \phi_1, \ldots, \phi_{2n}$ are integrable functions on X. Let*

$$D = D(t_1, \ldots, t_n)$$

be the determinant of the matrix with kth row

$$\phi_k(t_1), \psi_k(t_1), \phi_k(t_2), \psi_k(t_2), \ldots, \phi_k(t_n), \psi_k(t_n),$$

$k = 1, 2, \ldots, 2n$, $t_1, t_2, \ldots, t_n \in X$. Then

$$\Lambda = \int_X \cdots \int_X D d\mu(t_1) \ldots d\mu(t_n) = n! Pf(B).$$

Here $B = (b_{ij})$ is a $2n \times 2n$ skew-symmetric matrix with

$$b_{ij} = \int_X [\phi_i(x)\psi_j(x) - \phi_j(x)\psi_i(x)] d\mu(x).$$

Proof. Using the definition of the determinant, we have

$$D(t_1,\dots,t_n)$$
$$= \sum_{\sigma\in\Sigma(2n)} (-1)^{sign\sigma}\phi_{\sigma(1)}(t_1)\psi_{\sigma(2)}(t_1)\phi_{\sigma(3)}(t_2)\psi_{\sigma(4)}(t_2)\cdots\phi_{\sigma(2n-1)}(t_n)\psi_{\sigma(2n)}(t_n).$$

Hence,

$$\Lambda = \sum_{\sigma\in\Sigma(2n)} (-1)^{sign\sigma}\tilde{k}_{\sigma(1)\sigma(2)}\tilde{k}_{\sigma(3)\sigma(4)}\cdots\tilde{k}_{\sigma(2n-1)\sigma(2n)},$$

where

$$\tilde{k}_{ij} = \int_X \phi_i(t)\psi_j(t)d\mu(t).$$

Observe now that, in the expression above, $\tilde{k}_{ij}$ can be substituted by its skew-symmetric part $l_{ij} = (k_{ij} - k_{ji})/2$. Indeed, consider a two-form

$$\beta = \sum_{1\le i,j\le 2n} \alpha_{ij}e_i\wedge e_j \in \bigwedge^2(\mathbf{R^{2n}}).$$

Here $e_1,\dots,e_{2n}$ is a canonical basis in $\mathbf{R^{2n}}$, and α_{ij} are some real numbers. Taking n times the wedge product of β with itself, we obtain

$$\beta\wedge\beta\wedge\cdots\wedge\beta = \left(\sum_{\sigma\in\Sigma(2n)} (-1)^{sign\sigma}\alpha_{\sigma(1)\sigma(2)}\cdots\alpha_{\sigma(2n-1)\sigma(2n)}\right)\omega,$$

$$\omega = e_1\wedge e_2\wedge\cdots\wedge e_{2n-1}\wedge e_{2n}.$$

On the other hand,

$$\beta = \sum_{1\le i<j\le 2n} (\alpha_{ij}-\alpha_{ji})e_i\wedge e_j = \sum_{1\le i,j\le 2n}\gamma_{ij}e_i\wedge e_j,$$

where $\gamma_{ij} = (\alpha_{ij}-\alpha_{ji})/2$. Hence,

$$\beta\wedge\beta\wedge\cdots\wedge\beta = \left(\sum_{\sigma\in\Sigma(2n)} (-1)^{sign\sigma}\gamma_{\sigma(1)\sigma(2)}\cdots\gamma_{\sigma(2n-1)\sigma(2n)}\right)\omega.$$

Applying this observation to our situation, we obtain

$$\Lambda = \frac{1}{2^n}\sum_{\sigma\in\Sigma(2n)} (-1)^{sign\sigma}b_{\sigma(1)\sigma(2)}b_{\sigma(3)\sigma(4)}\cdots b_{\sigma(2n-1)\sigma(2n)}.$$

Hence,

$$\Lambda = n!Pf(B).$$

We are now in position to calculate the characteristic function of a cone generated by a Chebyshev system of odd order. Applying Proposition 1 to (3.5), we obtain the following theorem.

THEOREM 4. *Let $u_0, \dots, u_{2\nu-1}$ be a Chebyshev system of continuously differentiable functions on the interval $[a, b]$. Let p be a generalized polynomial strictly positive on $[a, b]$. Then*

$$F(p) = \epsilon Pf(B(p)),$$

where $B(p) = (b_{ij}(p))$,

$$b_{ij}(p) = \int_a^b \frac{u_i(t)u_j'(t) - u_j(t)u_i'(t)}{p(t)^2} dt,$$

$i, j = 0, 1, \dots, 2\nu - 1, \epsilon = \pm 1$.

The case of an even-order Chebyshev system is slightly more complicated. Let $u_0, \dots, u_{2\nu}$ be a Chebyshev system of continuously differentiable functions on an interval $[a, b]$. Assume that

$$u_0(a) = 1, \quad u_i(a) = 0, \quad i = 1, \dots, 2\nu. \tag{3.6}$$

By Theorem 2, each point $(c_0, c_1, \dots, c_{2\nu})^T \in \operatorname{int}(K^*)$ admits a unique representation of the form:

$$c_i = \sum_{j=1}^{\nu+1} \rho_j u_i(\xi_j), \tag{3.7}$$

$i = 0, 1, \dots, 2\nu, \xi_1 = a < \xi_2 < \cdots < \xi_{\nu+1} < b$. Consider the map

$$\Phi : \mathbf{R}_+^{\nu+1} \times \{\xi \in \mathbf{R}^\nu : a < \xi_2 < \cdots < \xi_{\nu+1} < b\} \to \operatorname{int}(K^*)$$

defined by (3.7). As is easily seen, the Jacobian of this map J has the form:

$$J = \mid \det[u(a), u(\xi_2), \dots, u(\xi_{\nu+1}), \rho_2 u'(\xi_2), \dots, \rho_{\nu+1} u'(\xi_{\nu+1})] \mid .$$

Here $u(\xi) = (u_0(\xi), u_1(\xi), \dots, u_{2\nu}(\xi))^T$. Observe now that, according to our assumption, $u(a) = e_1$. Thus, expanding J over the first column, we obtain

$$J = \prod_{i=2}^{\nu+1} \rho_i \mid \det[\tilde{u}(\xi_2), \tilde{u}'(\xi_2), \dots, \tilde{u}(\xi_{\nu+1}), \tilde{u}'(\xi_{\nu+1})] \mid,$$

where

$$\tilde{u}(\xi) = [u_1(\xi), \dots, u_{2\nu}(\xi)]^T.$$

Obviously, $u_1, \dots, u_{2\nu}$ form a Chebyshev system on $(a, b]$. Hence, we can apply Lemma 1 to conclude that

$$\begin{aligned} F(p) = {} & \frac{\epsilon}{\nu!} \int_0^{+\infty} e^{-p(a)\rho_1} d\rho_1 \\ & \times \int_{a<\xi_2<\cdots<\xi_{\nu+1}<b} \det[\tilde{u}(\xi_2), \tilde{u}'(\xi_2), \dots, \tilde{u}(\xi_{\nu+1}), \tilde{u}'(\xi_{\nu+1})] \\ & \times \left(\int_0^{+\infty} e^{-p(\xi_2)\rho_2} \rho_2 d\rho_2 \cdots \int_0^{+\infty} e^{-p(\xi_{\nu+1})\rho_{\nu+1}} \rho_{\nu+1} d\rho_{\nu+1} \right) d\xi_2 \dots d\xi_{\nu+1}, \end{aligned}$$

where $\epsilon = \pm 1$. By applying Proposition 1, we obtain the following theorem.

THEOREM 5. *Let $u_0, \dots, u_{2\nu}$ be a Chebyshev system of an even order of continuously differentiable functions on the interval $[a, b]$ such that $u(a) = e_1$. Let, further, p be a generalized polynomial strictly positive on $[a, b]$. Then*

$$F(p) = \epsilon \frac{Pf(B(p))}{p(a)},$$

where $B(p) = (b_{ij}(p))$,

$$b_{ij}(p) = \int_a^b \frac{u_i(\xi)u_j^{'}(\xi) - u_j(\xi)u_i^{'}(\xi)}{p(\xi)^2} d\xi,$$

$i, j = 1, 2, \dots, 2\nu$. Here $\epsilon = \pm 1$.

Similarly, using Theorem 3, we obtain the following.

THEOREM 6. *Let $u_0, \dots, u_{2\nu}$ be a periodic Chebyshev system of continuously differentiable functions on the interval $[a, b]$ such that $u(a) = e_1$. Let p be a generalized polynomial strictly positive on $[a, b]$. Then*

$$F(p) = \epsilon \frac{Pf(B(p))}{p(a)},$$

where $B(p) = (b_{ij}(p))$,

$$b_{ij}(p) = \int_a^b \frac{u_i(\xi)u_j^{'}(\xi) - u_j(\xi)u_i^{'}(\xi)}{p(\xi)^2} d\xi,$$

$i, j = 1, \dots, 2\nu$. Here $\epsilon = \pm 1$.

Observe now that the assumption made in Theorems 5 and 6 does not restrict the generality of our approach.

LEMMA 2. *Let $u_0, u_1, \dots, u_n$ be a Chebyshev system on a set E. Let $a \in E$. One can always choose a basis $v_0, \dots, v_n$ in span$(u_0, \dots, u_n)$ such that $v_0(a) = 1, v_i(a) = 0,\ i = 1, 2, \dots, n$.*

Proof. Indeed, for any pairwise distinct points $t_i, i = 0, \dots, n$, there exists $v_i \in$ span$(u_0, \dots, u_n)$ such that $v_i(t_j) = \delta_{ij}, i, j = 0, \dots, n$ (see, e.g., [4]).

REMARK 1. *Since $F(p) > 0, p \in \text{int}(K)$, in Theorems 4–6, we conclude that $Pf(B(p))$ does not change the sign on $\text{int}(K)$. Furthermore, since $\det(B(p)) = Pf(B(p))^2$ (see, e.g., [1]), we can easily rewrite $\ln F(p)$ in terms of $\ln \det B(p)$.*

4. Examples. We present here several simple examples to show how our barrier functions appear in low-dimensional situations and to illustrate possible applications of our results. First, observe that from general considerations (see, e.g., [6]) it follows that, given a cone K of generalized nonnegative polynomials corresponding to a Chebyshev system,

$$\Phi(p) = \alpha \ln F(p), \qquad p \in \text{int}(K),$$

is a homogeneous self-concordant barrier. Here $F(p)$ is the characteristic function of the cone K calculated in the previous section, and α is some positive constant depending in general on K. Here we use the standard normalization in the inequality between the second and the third derivatives of F. Then the barrier parameter of F has a universal bound (see [6]).

Example 6. Let $[a_j, b_j] \subset \mathbf{R}, j = 1, 2, \ldots, l$, and $u_i(t) = t^i, i = 0, 1, \ldots, n$. Denote by $K_{[a,b]}$ the cone of nonnegative polynomials of degree less than or equal to n on the interval $[a, b]$. Let, further,

$$K = \bigcap_{j=1}^{l} K_{[a_j, b_j]}.$$

Then

$$\Phi(p) = \sum_{j=1}^{l} \alpha_j \ln F_j(p)$$

is a homogeneous self-concordant barrier for the cone K. Here α_j are some positive constants, and F_j are the characteristic functions of the cones $K_{[a_j,b_j]}$ calculated in the previous section. This result easily follows from the general barrier calculus developed in [6].

Example 7. Let $u_0 = 1, u_1$ be a Chebyshev system of order one on an interval $[a, b]$. We assume that u_1 is a continuously differentiable function. In this situation the cone of generalized positive polynomials

$$K = \{(c_0, c_1)^T \in \mathbf{R^2} : c_0 + c_1 u_1(t) \geq 0, \quad t \in [a, b]\}$$

has a very simple structure. Indeed, one can easily see that the map $p \to (p(a), p(b))^T$ defines a linear isomorphism of K onto the nonnegative orthant in $\mathbf{R^2}$. According to Theorem 4, the characteristic function of the cone K has the form:

$$F(p) = \epsilon b_{01}(p), \qquad p \in \operatorname{int}(K),$$

where $\epsilon = \pm 1$ and

$$b_{01}(p) = \int_a^b \frac{u_1'(t)dt}{[c_0 + c_1 u_1(t)]^2}.$$

An easy calculation shows that

$$b_{01}(p) = \frac{u_1(b) - u_1(a)}{p(a)p(b)}.$$

Observe that $u(a) \neq u(b)$, since $1, u_1$ form a Chebyshev system on $[a, b]$. We thus obtain that

$$\Phi(p) = -\ln p(a) - \ln p(b) + \ln \mid u_1(b) - u_1(a) \mid$$

is a homogeneous self-concordant barrier. Here $\alpha = 1$. Of course, this is not a surprising result due to the above-mentioned isomorphism.

Example 8. Consider a periodic Chebyshev system $u_0(t) = 1, u_1(t) = \sin t, u_2(t) = \cos t$ on the interval $[0, 2\pi]$. Let $v_0(t) = u_0(t), v_1(t) = u_1(t), v_2(t) = u_2(t) - u_0(t)$. The basis v_0, v_1, v_2 of the vector space $span(u_0, u_1, u_2)$ satisfies the condition $v(0) = e_1 \in \mathbf{R^3}$. Let

$$p(t) = a + b\cos t + c\sin t > 0, \quad t \in [0, 2\pi].$$

According to Theorem 6,

$$F(p) = \epsilon \frac{I(p)}{p(0)},$$

where

$$I(p) = \int_0^{2\pi} \frac{v_1'(t)v_2(t) - v_1(t)v_2'(t)}{p(t)^2} dt$$

$$= \int_0^{2\pi} \frac{1 - \cos t}{p(t)^2} dt.$$

Let

$$J(a, b, c) = \int_0^{2\pi} \frac{(1 - \cos t)dt}{a + b\cos t + c\sin t}.$$

Using an explicit formula for the primitive of the function

$$\frac{1 - \cos t}{a + b\cos t + c\sin t}$$

(see, e.g., [2]), we can easily calculate $J(a, b, c)$:

$$J(a, b, c) = 2\pi \left[\left(1 + \frac{ab}{b^2 + c^2} \right) \frac{1}{\sqrt{a^2 - b^2 - c^2}} - \frac{b}{b^2 + c^2} \right],$$

provided $a > \sqrt{b^2 + c^2}$. It is then obvious that

$$I(p) = -\frac{\partial J}{\partial a} = \frac{2\pi(a + b)}{(a^2 - b^2 - c^2)^{3/2}}.$$

Since $p(0) = a + b$, we conclude that

$$F(p) = \frac{2\pi}{(a^2 - b^2 - c^2)^{3/2}}.$$

For the homogeneous self-concordant barrier, we thus obtain

$$\Phi(p) = \frac{2}{3} \ln F(p) = -\ln(a^2 - b^2 - c^2) + \frac{2}{3} \ln(2\pi).$$

Observe that $p(t) > 0$ for all $t \in [0, 2\pi]$ if and only if $a > \sqrt{b^2 + c^2}$. Thus K is, in this case, the second order cone in $\mathbf{R^3}$, and the expression for the self-concordant barrier is again not surprising.

5. Concluding remarks. In the present paper we explicitly calculated characteristic functions for a broad class of convex cones generated by Chebyshev systems. For the "cones of squares" considered in [5] our results complement in a quite fortunate way the results of Nesterov, who found explicit self-concordant barriers for duals of cones of squares. It paves the way, at least in principle, for the construction of primal-dual algorithms for corresponding optimization problems. Observe that an important class of approximation problems involving cones generated by polynomial splines is within the reach of the technique developed in the present paper.

Acknowledgments. This paper was completed while the author visited Chinese University of Hong Kong. The hospitality and support of John Moore and Wing Wong are greatly appreciated.

REFERENCES

[1] N.G. de Bruijn, *On some multiple integrals involving determinants*, J. Indian Math. Soc., 19 (1955), pp. 133–151.

[2] I.S. Gradstein and I.M. Ryzhik, *Table of Integrals, Series and Products*, Academic Press, New York, 1994, p. 182.

[3] L. Faybusovich, *Euclidean Jordan algebras and interior-point algorithms*, J. Positivity, 1 (1997), pp. 331–357.

[4] S. Karlin and W.J. Studden, *Tchebycheff Systems: With Applications in Analysis and Statistics*, Inrescience, New York, 1966.

[5] Y. Nesterov, *Squared functional systems and optimization problems*, in High Performance Optimization, Kluwer, Dordrecht, 2000, pp. 441–466.

[6] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM Stud. Appl. Math. 13, SIAM, Philadelphia, 1994.

[7] J. Renegar, *Linear programming, complexity theory and elementary functional analysis*, Math. Programming, 70 (1995), pp. 279–351.

SIAM J. OPTIM.
Vol. 12, No. 3, pp. 782–810

WARM-START STRATEGIES IN INTERIOR-POINT METHODS FOR LINEAR PROGRAMMING*

E. ALPER YILDIRIM† AND STEPHEN J. WRIGHT‡

Abstract. We study the situation in which, having solved a linear program with an interior-point method, we are presented with a new problem instance whose data is slightly perturbed from the original. We describe strategies for recovering a "warm-start" point for the perturbed problem instance from the iterates of the original problem instance. We obtain worst-case estimates of the number of iterations required to converge to a solution of the perturbed instance from the warm-start points, showing that these estimates depend on the size of the perturbation and on the conditioning and other properties of the problem instances.

Key words. warm-start, reoptimization, interior-point methods, linear programming

AMS subject classifications. 90C51, 90C05

PII. S1052623400369235

1. Introduction.

This paper describes and analyzes warm-start strategies for interior-point methods applied to linear programming (LP) problems. We consider the situation in which one linear program, the "original instance," has been solved by an interior-point method, and we are then presented with a new problem of the same dimensions, the "perturbed instance," in which the data is slightly different. Interior-point iterates for the original instance are used to obtain warm-start points for the perturbed instance, so that when an interior-point method is started from this point, it finds the solution in fewer iterations than if no prior information were available. Although our results are theoretical, the strategies proposed here can be applied to practical situations, an aspect that is the subject of ongoing study.

The situation we have outlined arises, for instance, when linearization methods are used to solve nonlinear problems, as in the sequential LP algorithm. (One extension of this work that we plan to investigate is the extension to convex quadratic programs, which would be relevant to the solution of subproblems in sequential quadratic programming algorithms.) Our situation is different from the one considered by Gondzio [5], who deals with the case in which the number of variables or constraints in the problem is increased and the dimensions of the problem data objects are correspondingly expanded. The latter situation arises in solving subproblems arising from cutting-plane or column-generation algorithms, for example. The reader is also referred to Mitchell and Borchers [8] and Gondzio and Vial [6] for consideration of warm-start strategies in a cutting-plane scheme.

Freund [4] develops and analyzes a potential reduction algorithm from an infeasible warm-start, in which the iterate satisfies the equality constraints but is allowed to violate nonnegativity.

*Received by the editors March 7, 2000; accepted for publication (in revised form) August 18, 2001; published electronically February 27, 2002. This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, U.S. Department of Energy, under contract W-31-109-Eng-38, and by the National Science Foundation under grants 0082065 and 0086559.

http://www.siam.org/journals/siopt/12-3/36923.html

†Department of Applied Mathematics and Statistics, State University of New York at Stony Brook, 1-107 Math Tower, Stony Brook, NY 11794–3600 (yildirim@ams.sunysb.edu).

‡Computer Sciences Department, University of Wisconsin-Madison, 1210 West Dayton Street, Madison, WI 53706 (swright@cs.wisc.edu).

For our analysis, we use the tools developed by Nunez and Freund [10], which in turn are based on the work of Renegar [11, 12, 13, 14] on the conditioning of linear programs and the complexity of algorithms for solving them. We also use standard complexity analysis techniques from the interior-point literature for estimating the number of iterations required to solve a linear program to given accuracy.

We start in section 2 with an outline of notation and a restatement and slight generalization of the main result from Nunez and Freund [10]. Section 3 outlines the warm-start strategies that we analyze in the paper and describes how they can be used to obtain reduced complexity estimates for interior-point methods. In section 4 we consider a warm-start technique in which a least-squares change is applied to a feasible interior-point iterate for the original instance to make it satisfy the constraints for the perturbed instance. We analyze this technique for central path neighborhoods based on both the Euclidean norm and the ∞ norm, deriving in each case a worst-case estimate for the number of iterations required by an interior-point method to converge to an approximate solution of the perturbed instance. In section 5 we study the technique of applying one iteration of Newton's method to a system of equations that is used to recover a strictly feasible point for the perturbed instance from a feasible iterate for the original instance. Section 6 discusses the relationship between the two warm-start strategies and the weighted versions of least-squares corrections. A small example is used to illustrate the behavior of different correction strategies. Finally, we conclude the paper with some discussions in section 7.

2. Preliminaries: Conditioning of LPs, central path neighborhoods, bounds on feasible points. We consider the LP in the following standard form:

$$\text{(P)} \qquad \min_x \ c^T x \quad \text{subject to (s.t.)} \ \ Ax = b, \ x \geq 0,$$

where $A \in R^{m\times n}$, $b \in R^m$, and $c \in R^n$ are given and $x \in R^n$. The associated dual LP is given by the following:

$$\text{(D)} \qquad \max_{y,s} \ b^T y \quad \text{s.t.} \ \ A^T y + s = c, \ s \geq 0,$$

where $y \in R^m$ and $s \in R^n$. We borrow the notation of Nunez and Freund [10], denoting by d the data triplet (A, b, c) that defines the problems (P) and (D). We define the norm of d (differently from Nunez and Freund) as the maximum of the Euclidean norms of the three data components:

$$\|d\| \stackrel{\text{def}}{=} \max(\|A\|_2, \|b\|_2, \|c\|_2). \tag{2.1}$$

We use the norm notation $\|\cdot\|$ on a vector or matrix to denote the Euclidean norm and the operator norm it induces, respectively, unless explicitly indicated otherwise.

We use $\mathcal{F}$ to denote the space of strictly feasible data instances, that is,

$$\mathcal{F} = \{(A, b, c) : \exists\ x, y, s \text{ with } (x, s) > 0 \text{ such that } Ax = b,\ A^T y + s = c\}.$$

The complement of $\mathcal{F}$, denoted by $\mathcal{F}^C$, consists of data instances d for which either (P) or (D) does not have any strictly feasible solutions. The (shared) boundary of $\mathcal{F}$ and $\mathcal{F}^C$ is given by

$$\mathcal{B} = \text{cl}(\mathcal{F}) \cap \text{cl}(\mathcal{F}^C),$$

where $\text{cl}(\cdot)$ denotes the closure of a set. Since $(0, 0, 0) \in \mathcal{B}$, we have that $\mathcal{B} \neq \emptyset$. The data instances $d \in \mathcal{B}$ will be called *ill-posed data instances*, since arbitrarily small

perturbations in the data d can result in data instances in either $\mathcal{F}$ or $\mathcal{F}^C$. The *distance to ill-posedness* is defined as

$$\rho(d) = \inf\{\|\Delta d\| \,:\, d + \Delta d \in \mathcal{B}\}, \tag{2.2}$$

where we use the norm (2.1) to define $\|\Delta d\|$. The condition number of a feasible problem instance d is defined as

$$\mathcal{C}(d) \stackrel{\text{def}}{=} \frac{\|d\|}{\rho(d)} \quad (\text{with } \mathcal{C}(d) \stackrel{\text{def}}{=} \infty \text{ when } \rho(d) = 0). \tag{2.3}$$

Since the perturbation $\Delta d = -d$ certainly has $d + \Delta d = 0 \in \mathcal{B}$, we have that $\rho(d) \leq \|d\|$ and therefore $\mathcal{C}(d) \geq 1$. Note, too, that $\mathcal{C}(d)$ is invariant under a nonnegative multiplicative scaling of the data d; that is, $\mathcal{C}(\beta d) = \mathcal{C}(d)$ for all $\beta > 0$.

Robinson [15] and Ashmanov [1] showed that a data instance $d \in \mathcal{F}$ satisfies $\rho(d) > 0$ (that is, d lies in the interior of $\mathcal{F}$) if and only if A has full row rank. For such d, another useful bound on $\rho(d)$ is provided by the minimum singular value of A. If we write the singular value decomposition of A as

$$A = USV^T = \sum_{i=1}^{m} \sigma_i(A) u_i v_i^T,$$

where U and V are orthogonal and $S = \text{diag}(\sigma_1(A), \sigma_2(A), \ldots, \sigma_m(A))$, with $\sigma_1(A) \geq \sigma_2 \geq \cdots \geq \sigma_m(A) > 0$ denoting the singular values of A, then the perturbation

$$\Delta A = -\sigma_m(A) u_m v_m^T$$

is such that $A + \Delta A$ is singular, and moreover $\|\Delta A\| = \sigma_m(A)$ due to the fact that the Euclidean norm of a rank-one matrix satisfies the property

$$\|\beta u v^T\|_2 = |\beta| \, \|u\|_2 \, \|v\|_2. \tag{2.4}$$

We conclude that

$$\rho(d) \leq \sigma_m(A). \tag{2.5}$$

It is well known that for such $d \in \text{int}(\mathcal{F})$ the system given by

$$Ax = b, \tag{2.6a}$$
$$A^T y + s = c, \tag{2.6b}$$
$$XSe = \mu e, \tag{2.6c}$$
$$(x, s) > 0 \tag{2.6d}$$

has a unique solution for every $\mu > 0$, where e denotes the vector of ones in the appropriate dimension, and X and S are the diagonal matrices formed from the components of x and s, respectively. We denote the solutions of (2.6) by $(x(\mu), y(\mu), s(\mu))$ and use $\mathcal{P}$ to denote the *central path* traced out by these solutions for $\mu > 0$, that is,

$$\mathcal{P} \stackrel{\text{def}}{=} \{(x(\mu), y(\mu), s(\mu)) \,:\, \mu > 0\}. \tag{2.7}$$

Throughout this paper, we assume that the original data instance d lies in $\mathcal{F}$ and that $\rho(d) > 0$. In sections 4 and 5, we assume further that the original data

instance d has been solved by a feasible path-following interior-point method. Such a method generates a sequence of iterates (x^k, y^k, s^k) that satisfy the relations (2.6a), (2.6b), and (2.6d) and for which the pairwise products $x_i^k s_i^k$, $i = 1, 2, \ldots, n$, are not too different from one another, in the sense of remaining within some well-defined "neighborhood" of the central path. The duality measure $(x^k)^T s^k$ is driven toward zero as $k \to \infty$, and search directions are obtained by applying a modified Newton's method to the nonlinear system formed by (2.6a), (2.6b), and (2.6c).

We now give some notation for feasible sets and central path neighborhoods associated with the particular problem instance $d = (A, b, c)$. Let $\mathcal{S}$ and $\mathcal{S}^0$ denote the set of feasible and strictly feasible primal-dual points, respectively; that is,

$$\begin{aligned} \mathcal{S} &= \{(x, y, s) \ : \ Ax = b, \ A^T y + s = c, \ (x, s) \geq 0\}, \\ \mathcal{S}^0 &= \{(x, y, s) \in \mathcal{S} \ : \ (x, s) > 0\}. \end{aligned}$$

(Note that $d \in \mathcal{F}$ if and only if $\mathcal{S}^0 \neq \emptyset$.) We refer to the central path neighborhoods most commonly used in interior-point methods as the *narrow* and *wide* neighborhoods. The narrow neighborhood denoted by $\mathcal{N}_2(\theta)$ is defined as

$$\mathcal{N}_2(\theta) = \{(x, y, s) \in \mathcal{S}^0 \ : \ \|XSe - (x^T s/n)e\|_2 \leq \theta(x^T s/n)\} \tag{2.8}$$

for $\theta \in [0, 1)$. The wide neighborhood, which is denoted by $\mathcal{N}_{-\infty}(\gamma)$, is given by

$$\mathcal{N}_{-\infty}(\gamma) = \{(x, y, s) \in \mathcal{S}^0 \ : \ x_i s_i \geq \gamma(x^T s/n) \ \forall \ i = 1, 2, \ldots, n\}, \tag{2.9}$$

where u_i denotes the ith component of the vector u, and the parameter γ lies in $(0, 1]$.

We typically use a bar to denote the corresponding quantities for the perturbed problem instance $d + \Delta d$. That is, we have

$$\begin{aligned} \bar{\mathcal{S}} &= \{(x, y, s) \ : \ (A + \Delta A)x = (b + \Delta b), \ (A + \Delta A)^T y + s = (c + \Delta c), \ (x, s) \geq 0\}, \\ \bar{\mathcal{S}}^o &= \{(x, y, s) \in \bar{\mathcal{S}} \ : \ (x, s) > 0\}, \end{aligned}$$

whereas

$$\bar{\mathcal{N}}_2(\theta) = \{(x, y, s) \in \bar{\mathcal{S}}^o \ : \ \|XSe - (x^T s/n)e\|_2 \leq \theta(x^T s/n)\}, \tag{2.10a}$$

$$\bar{\mathcal{N}}_{-\infty}(\gamma) = \{(x, y, s) \in \bar{\mathcal{S}}^o \ : \ x_i s_i \geq \gamma(x^T s/n) \ \forall \ i = 1, 2, \ldots, n\}. \tag{2.10b}$$

We associate a value of μ with each iterate $(x, y, s) \in \mathcal{S}$ (or $\bar{\mathcal{S}}$) by setting

$$\mu = x^T s/n. \tag{2.11}$$

We call this μ the *duality measure* of the point (x, y, s). When (x, y, s) is feasible, it is easy to show that the duality gap $c^T x - b^T y$ is equal to $n\mu$.

Finally, we state a modified version of Theorem 3.1 from Nunez and Freund [10], which uses our definition (2.1) of the norm of the data instance and takes note of the fact that the proof in [10] continues to hold when we consider strictly feasible points that do not lie exactly on the central path $\mathcal{P}$.

THEOREM 2.1. *If $d = (A, b, c) \in \mathcal{F}$ and $\rho(d) > 0$, then for any point (x, y, s) satisfying the conditions*

$$Ax = b, \quad A^T y + s = c, \quad (x, s) > 0, \tag{2.12}$$

the following bounds are satisfied:

$$\|x\| \le \mathcal{C}(d)(\mathcal{C}(d) + \mu n/\|d\|), \tag{2.13a}$$

$$\|y\| \le \mathcal{C}(d)(\mathcal{C}(d) + \mu n/\|d\|), \tag{2.13b}$$

$$\|s\| \le 2\|d\|\mathcal{C}(d)(\mathcal{C}(d) + \mu n/\|d\|), \tag{2.13c}$$

where we have defined μ *as in* (2.11).

The proof exactly follows the logic of the proof in [10, Theorem 3.1], but differs in many details because of our use of Euclidean norms on the matrices and vectors. For instance, where the original proof defines a perturbation $\Delta A = -be^T/\|x\|_1$, to obtain an infeasible data instance, we instead use $\Delta A = -bx^T/\|x\|_2^2$. We also use the observation (2.4) repeatedly.

3. Warm starts and reduced complexity. Before describing specific strategies for warm starts, we preview the nature of our later results and show how they can be used to obtain improved estimates of the complexity of interior-point methods that use these warm starts.

We start by recalling some elements of the complexity analysis of interior-point methods. These methods typically produce iterates (x^k, y^k, s^k) that lie within a neighborhood such as (2.8) or (2.9) and for which the duality measure μ_k (defined as in (2.11) by $\mu_k = (x^k)^T s^k/n$) decreases monotonically with k, according to a bound of the following form:

$$\mu_{k+1} \le \left(1 - \frac{\delta}{n^\tau}\right)\mu_k, \tag{3.1}$$

where δ and τ are positive constants that depend on the algorithm. Typically, τ is 0.5, 1, or 2, while δ depends on the parameters θ or γ that define the neighborhood (see (2.8) and (2.9)) and on various other algorithmic parameters. Given a starting point (x^0, y^0, s^0) with duality measure μ_0, the number of iterations required to satisfy the stopping criterion

$$\mu \le \epsilon\|d\| \tag{3.2}$$

(for some small positive ϵ) is bounded by

$$\frac{\log(\epsilon\|d\|) - \log\mu_0}{\log\left(1 - \delta/n^\tau\right)} = \mathcal{O}\left(n^\tau \log\frac{\mu_0}{\|d\|\epsilon}\right). \tag{3.3}$$

It follows from this bound that, provided we have

$$\frac{\mu_0}{\|d\|} = \mathcal{O}(1/\epsilon^\eta)$$

for some fixed $\eta > 0$—which can be guaranteed for small ϵ when we apply a cold-start procedure—the number of iterations required to achieve (3.2) is

$$\mathcal{O}(n^\tau|\log\epsilon|). \tag{3.4}$$

Our warm-start strategies aim to find a starting point for the *perturbed* instance that lies inside one of the neighborhoods (2.10) and for which the initial duality measure $\bar{\mu}_0$ is not too large. By applying (3.3) to the perturbed instance, we see that

if $\bar{\mu}_0/\|d+\Delta d\|$ is less than 1, then the formal complexity of the method will be better than the general estimate (3.4).

Both warm-start strategies that we describe in subsequent sections proceed by taking a point (x, y, s) from a neighborhood such as (2.8), (2.9) for the original instance and calculating an adjustment $(\Delta x, \Delta y, \Delta s)$ based on the perturbation Δd to obtain a starting point for the perturbed instance. The strategies are simple; their computational cost is no greater than the cost of one interior-point iteration. They do not succeed in producing a valid starting point unless the point (x, y, s) from the original problem has a large enough value of $\mu = x^T s/n$. That is, we must retreat to a prior iterate for the original instance until the adjustment $(\Delta x, \Delta y, \Delta s)$, when added to this iterate, does not cause some components of x or s to become negative. (Indeed, we require a stronger condition to hold: that the adjusted point $(x + \Delta x, y + \Delta y, s + \Delta s)$ belong to a neighborhood such as those of (2.10).) Since larger perturbations Δd generally lead to larger adjustments $(\Delta x, \Delta y, \Delta s)$, the prior iterate to which we must retreat is further back in the iteration sequence when Δd is larger. Most of the results in the following sections quantify this observation. They give a lower bound on $\mu/\|d\|$—expressed in terms of the size of the components of Δd, the conditioning $\mathcal{C}(d)$ of the original problem, and other quantities—such that the warm-start strategy, applied from a point (x, y, s) satisfying $\mu = x^T s/n$ and a neighborhood condition, yields a valid starting point for the perturbed problem.

Our strategy contrasts with that of Gondzio, who uses the solution of the original problem as a starting point in the computation of a central path point for the new problem, which has additional columns in the matrix A. Our strategies instead rely on a single correction to an interior-point iterate for the original problem to obtain a loosely centered starting point for the modified problem. We focus just on correcting the infeasibility of the linear equality conditions in (P) and (D), relying on the loose centrality of the original iterate to provide us with sufficient centrality of the adjusted starting point.

These results can be applied in a practical way when an interior-point approach is used to solve the original instance. Let $\{(x^k, y^k, s^k),\ k = 0, \ldots, K\}$ denote the iterates generated while solving the original problem. One can then store a subset of the iterates $\{(x^{k_i}, y^{k_i}, s^{k_i}), i = 0, 1, \ldots, L\}$ with $k_0 = 0$, which is the shortest sequence satisfying the property

$$\mu_{k_{i+1}} \geq \nu \mu_{k_i} \quad \forall i = 0, 1, 2, \ldots \tag{3.5}$$

for some ν with $0 < \nu \ll 1$. Suppose that we denote the lower bound discussed in the preceding paragraph by $\mu^*/\|d\|$. Then the best available starting point from the saved subsequence is the one with index k_ℓ, where ℓ is the largest index for which

$$\mu_{k_\ell} \geq \mu^*.$$

Because of (3.5) and the choice of ℓ, we have in fact that

$$\mu^* \leq \mu_{k_\ell} \leq (1/\nu)\mu^*. \tag{3.6}$$

The warm-start point is then

$$(\bar{x}^0, \bar{y}^0, \bar{s}^0) = (x^{k_\ell}, y^{k_\ell}, s^{k_\ell}) + (\Delta x, \Delta y, \Delta s), \tag{3.7}$$

where $(\Delta x, \Delta y, \Delta s)$ is the adjustment computed from one of our warm-start strategies. The duality measure corresponding to this point is

$$\bar{\mu}_0 = (\bar{x}^0)^T \bar{s}^0/n = \mu_{k_\ell} + (x^{k_\ell})^T \Delta s + (s^{k_\ell})^T \Delta x + \Delta x^T \Delta s.$$

By using the bounds on the components of $(\Delta x, \Delta y, \Delta s)$ that are obtained during the proofs of each major result, in conjunction with the bounds (2.13), we find that $\bar{\mu}_0$ can be bounded above by some multiple of $\mu^* + \mu_{k_\ell}$. Because of (3.6), we can deduce in each case that

$$\bar{\mu}_0 \leq \kappa \mu^* \tag{3.8}$$

for some κ independent of the problem instance d and the perturbation Δd. We conclude, by applying (3.3) to the perturbed instance, that the number of iterations required to satisfy the stopping criterion

$$\mu \leq \epsilon \|d + \Delta d\|, \tag{3.9}$$

starting from $(\bar{x}^0, \bar{y}^0, \bar{s}^0)$, is bounded by

$$\mathcal{O}\left(n^\tau \log \frac{\mu^*}{\|d + \Delta d\|\epsilon}\right). \tag{3.10}$$

Since our assumptions on $\|\Delta d\|$ usually ensure that

$$\|\Delta d\| \leq 0.5\|d\|, \tag{3.11}$$

we have that

$$\frac{1}{\|d + \Delta d\|} \leq \frac{1}{\|d\| - \|\Delta d\|} \leq \frac{2}{\|d\|},$$

so that (3.10) can be expressed more conveniently as

$$\mathcal{O}\left(n^\tau \log \frac{\mu^*}{\|d\|\epsilon}\right). \tag{3.12}$$

After some of the results in subsequent sections, we will substitute for τ and μ^* in (3.12), to express the bound on the number of iterations in terms of the conditioning $\mathcal{C}(d)$ of the original instance and the size of the perturbation Δd.

Our first warm-start strategy, a least-squares correction, is described in section 4. The second strategy, a "Newton step correction," is based on a recent paper by Yıldırım and Todd [18] and is described in section 5.

4. Least-squares correction. For much of this section, we restrict our analysis to the changes in b and c only; that is, we assume

$$\Delta d = (0, \Delta b, \Delta c). \tag{4.1}$$

Perturbations to A will be considered in section 4.3.

Given any primal-dual feasible point (x, y, s) for the instance d, the least-squares correction for the perturbation (4.1) is the vector $(\Delta x, \Delta y, \Delta s)$ obtained from the solutions of the following subproblems:

$$\begin{aligned} \min \|\Delta x\| \quad \text{s.t.} \;\; & A(x + \Delta x) = b + \Delta b, \\ \min \|\Delta s\| \quad \text{s.t.} \;\; & A^T(y + \Delta y) + (s + \Delta s) = c + \Delta c. \end{aligned}$$

Because $Ax = b$ and $A^T y + s = c$, we can restate these problems as

$$\min \|\Delta x\| \quad \text{s.t.} \;\; A\Delta x = \Delta b, \tag{4.2a}$$

$$\min \|\Delta s\| \quad \text{s.t.} \;\; A^T \Delta y + \Delta s = \Delta c, \tag{4.2b}$$

which are independent of (x, y, s). Given the following QR factorization of A^T,

$$A^T = \begin{bmatrix} Y & Z \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix} = YR, \tag{4.3}$$

where $[Y \quad Z]$ is orthogonal and R is upper triangular, we find by simple manipulation of the optimality conditions that the solutions can be written explicitly as

$$\Delta x = YR^{-T}\Delta b, \tag{4.4a}$$

$$\Delta y = R^{-1}Y^T\Delta c, \tag{4.4b}$$

$$\Delta s = (I - YY^T)\Delta c. \tag{4.4c}$$

Contrary to a usual feasible interior-point step, Δx is in the range space of A^T, and Δs is in the null space of A. Consequently,

$$\Delta x^T \Delta s = 0. \tag{4.5}$$

Our strategy is as follows: We calculate the correction (4.4) just once, then choose an iterate (x^k, y^k, s^k) for the original problem such that $(x^k + \Delta x, s^k + \Delta s) > 0$, $(x^k + \Delta x, y^k + \Delta y, s^k + \Delta s)$ lies within either $\bar{\mathcal{N}}_2(\theta)$ or $\bar{\mathcal{N}}_{-\infty}(\gamma)$, and k is the largest index for which these properties hold. We hope to be able to satisfy these requirements for some index k for which the parameter μ_k is not too large. In this manner, we hope to obtain a starting point for the perturbed problem for which the initial value of μ is not large, so that we can solve the problem using a smaller number of interior-point iterations than if we had started without the benefit of the iterates from the original problem.

Some bounds that we use throughout our analysis follow immediately from (4.4):

$$\|\Delta s\| \le \|\Delta c\|, \qquad \|\Delta x\| \le \frac{\|\Delta b\|}{\sigma_m(A)} \le \frac{\|\Delta b\|}{\rho(d)}, \tag{4.6}$$

where, as in (2.5), $\sigma_m(A)$ is the minimum singular value of A. These bounds follow from the fact that $I - YY^T$ is an orthogonal projection matrix onto the null space of A and from the observation that R has the same singular values as A. By defining

$$\delta_b = \frac{\|\Delta b\|}{\|d\|}, \qquad \delta_c = \frac{\|\Delta c\|}{\|d\|}, \tag{4.7}$$

we can rewrite (4.6) as

$$\|\Delta s\| \le \|d\|\delta_c, \qquad \|\Delta x\| \le \mathcal{C}(d)\delta_b. \tag{4.8}$$

We also define the following quantity, which occurs frequently in the analysis:

$$\delta_{bc} = \delta_c + 2\mathcal{C}(d)\delta_b. \tag{4.9}$$

In the remainder of the paper, we make the mild assumption that

$$\delta_b < 1, \qquad \delta_c < 1. \tag{4.10}$$

4.1. Small neighborhood. Suppose that we have iterates for the original problem that satisfy the following property, for some $\theta_0 \in (0,1)$:

$$\|XSe - \mu e\|_2 \le \theta_0 \mu, \quad \text{where } \mu = x^T s/n. \tag{4.11}$$

That is, $(x,y,s) \in \mathcal{N}_2(\theta_0)$. Iterates of a short-step path-following algorithm typically satisfy a condition of this kind. Since (x,y,s) is a strictly feasible point, its components satisfy the bounds (2.13). Note, too, that we have

$$\|XSe - \mu e\| \le \theta_0 \mu \;\Rightarrow\; (1-\theta_0)\mu \le x_i s_i \le (1+\theta_0)\mu. \tag{4.12}$$

Our first proposition gives conditions on δ_{bc} and μ that ensure that the least-squares correction yields a point in the neighborhood $\bar{\mathcal{N}}_{-\infty}(\gamma)$.

PROPOSITION 4.1. *Let $\gamma \in (0, 1-\theta_0)$ be given, and let $\xi \in (0, 1-\gamma-\theta_0)$. Assume that Δd satisfies*

$$\delta_{bc} \le \frac{1-\theta_0-\gamma-\xi}{(n+1)\mathcal{C}(d)}. \tag{4.13}$$

Let $(x,y,s) \in \mathcal{N}_2(\theta_0)$, and suppose that $(\Delta x, \Delta y, \Delta s)$ is the least-squares correction (4.4). Then $(x+\Delta x, y+\Delta y, s+\Delta s)$ lies in $\bar{\mathcal{N}}_{-\infty}(\gamma)$ if

$$\mu \ge \frac{\|d\|}{\xi} 3\mathcal{C}(d)^2 \delta_{bc} \stackrel{\text{def}}{=} \mu_1^*. \tag{4.14}$$

Proof. By using (4.12), (2.13), (4.8), and (4.9), we obtain a lower bound on $(x_i+\Delta x_i)(s_i+\Delta s_i)$ as follows:

$$\begin{aligned}
&(x_i+\Delta x_i)(s_i+\Delta s_i)\\
&= x_i s_i + x_i \Delta s_i + \Delta x_i s_i + \Delta x_i \Delta s_i\\
&\ge (1-\theta_0)\mu - \|x\|\|\Delta s\| - \|\Delta x\|\|s\| - \|\Delta x\|\|\Delta s\|\\
&\ge (1-\theta_0)\mu - \mathcal{C}(d)\left(\mathcal{C}(d) + \mu n/\|d\|\right)\|d\|\delta_c\\
&\qquad - 2\|d\|\mathcal{C}(d)^2(\mathcal{C}(d)+\mu n/\|d\|)\delta_b - \|d\|\mathcal{C}(d)\delta_b\delta_c\\
&\ge \mu\left(1-\theta_0 - n\mathcal{C}(d)\delta_{bc}\right) - \mathcal{C}(d)^2\|d\|\delta_{bc} - \mathcal{C}(d)\|d\|\delta_b\delta_c\\
&\ge \mu\left(1-\theta_0 - n\mathcal{C}(d)\delta_{bc}\right) - 2\mathcal{C}(d)^2\|d\|\delta_{bc}. \qquad (4.15)
\end{aligned}$$

Because of our assumption (4.13), the coefficient of μ in (4.15) is positive, and thus (4.15) represents a positive lower bound on $(x_i+\Delta x_i)(s_i+\Delta s_i)$ for all μ sufficiently large.

For an upper bound on $(x+\Delta x)^T(s+\Delta s)/n$, we have from (2.13), (4.8), and the relation (4.5) that

$$\begin{aligned}
&(x+\Delta x)^T(s+\Delta s)/n\\
&\le \mu + \|\Delta x\|\|s\|/n + \|x\|\|\Delta s\|/n\\
&\le \mu + 2\mathcal{C}(d)^2\|d\|\delta_b(\mathcal{C}(d)+\mu n/\|d\|)/n + \mathcal{C}(d)\|d\|\delta_c(\mathcal{C}(d)+\mu n/\|d\|)/n\\
&\le \mu(1+\mathcal{C}(d)\delta_{bc}) + \mathcal{C}(d)^2\|d\|\delta_{bc}/n. \qquad (4.16)
\end{aligned}$$

It follows from this bound and (4.15) that a sufficient condition for the conclusion of the proposition to hold is that

$$\mu(1-\theta_0 - n\mathcal{C}(d)\delta_{bc}) - 2\mathcal{C}(d)^2\|d\|\delta_{bc} \ge \gamma\mu(1+\mathcal{C}(d)\delta_{bc}) + \gamma\mathcal{C}(d)^2\|d\|\delta_{bc}/n,$$

which is equivalent to

$$\mu \geq \frac{\|d\|\mathcal{C}(d)^2\delta_{bc}(2+\gamma/n)}{1-\theta_0-\gamma-\mathcal{C}(d)\delta_{bc}(n+\gamma)}, \tag{4.17}$$

provided that the denominator is positive. Because of condition (4.13), and using $\gamma \in (0,1)$ and $n \geq 1$, the denominator is in fact bounded below by the positive quantity ξ, and thus the condition (4.17) is implied by (4.14).

Finally, we show that our bounds ensure the positivity of $x+\Delta x$ and $s+\Delta s$. It is easy to show that the right-hand side of (4.15) is also a lower bound on $(x_i + \alpha\Delta x_i)(s_i+\alpha\Delta s_i)$ for all $\alpha \in [0,1]$ and all $i = 1,2,\ldots,n$. Because μ satisfies (4.17), we have $(x_i+\alpha\Delta x_i)(s_i+\alpha\Delta s_i) > 0$ for all $\alpha \in [0,1]$. Since we know that $(x,s) > 0$, we conclude that $x_i + \Delta x_i > 0$ and $s_i + \Delta s_i > 0$ for all i as well, completing the proof. ∎

Next, we seek conditions on δ_{bc} and μ that ensure that the corrected iterate lies in a narrow central path neighborhood for the perturbed problem.

PROPOSITION 4.2. *Let $\theta > \theta_0$ be given, and let $\xi \in (0, \theta-\theta_0)$. Assume that the perturbation Δd satisfies*

$$\delta_{bc} \leq \frac{\theta-\theta_0-\xi}{(2n+1)\mathcal{C}(d)}. \tag{4.18}$$

Suppose that $(x,y,s) \in \mathcal{N}_2(\theta_0)$ for the original problem and that $(\Delta x, \Delta y, \Delta s)$ is the least-squares correction (4.4). *Then, $(x+\Delta x, y+\Delta y, s+\Delta s)$ will lie in $\bar{\mathcal{N}}_2(\theta)$ if*

$$\mu \geq \frac{\|d\|}{\xi} 4\mathcal{C}(d)^2\delta_{bc} \stackrel{\text{def}}{=} \mu_2^*. \tag{4.19}$$

Proof. We start by finding a bound on the norm of the vector

$$[(x_i+\Delta x_i)(s_i+\Delta s_i)]_{i=1,2,\ldots,n} - \left[(x+\Delta x)^T(s+\Delta s)/n\right] e. \tag{4.20}$$

Given two vectors y and z in R^n, we have that

$$\left\|[y_i z_i]_{i=1,2,\ldots,n}\right\| \leq \|y\|\,\|z\|, \qquad \left|y^T z\right| \leq \|y\|\,\|z\|. \tag{4.21}$$

By using these elementary inequalities together with (4.5), (4.8), (4.9), and (2.13), we have that the norm of (4.20) is bounded by

$$\begin{aligned}
&\left\|[x_i s_i]_{i=1,2,\ldots,n} - \mu e\right\| + 2\left[\,\|\Delta x\|\,\|s\| + \|x\|\,\|\Delta s\|\,\right] + \|\Delta x\|\,\|\Delta s\| \\
&\leq \theta_0\mu + 2\mathcal{C}(d)\|d\|\delta_{bc}\left(\mathcal{C}(d) + n\mu/\|d\|\right) + \mathcal{C}(d)\|d\|\delta_b\delta_c \\
&\leq \left[\theta_0 + 2n\mathcal{C}(d)\delta_{bc}\right]\mu + 3\|d\|\mathcal{C}(d)^2\delta_{bc}.
\end{aligned}$$

Meanwhile, we obtain a lower bound on the duality measure after the correction by using the same set of relations:

$$\begin{aligned}
(x+\Delta x)^T(s+\Delta s)/n &\geq \mu - \left[\,\|\Delta x\|\,\|s\| + \|x\|\,\|\Delta s\|\,\right]/n \\
&\geq \mu - \mathcal{C}(d)\|d\|\delta_{bc}(\mathcal{C}(d) + n\mu/\|d\|)/n \\
&\geq \mu\left[1 - \mathcal{C}(d)\delta_{bc}\right] - \mathcal{C}(d)^2\|d\|\delta_{bc}/n.
\end{aligned} \tag{4.22}$$

Therefore, a sufficient condition for

$$(x+\Delta x, y+\Delta y, s+\Delta s) \in \bar{\mathcal{N}}_2(\theta)$$

is that

$$[\theta_0 + 2n\mathcal{C}(d)\delta_{bc}]\,\mu + 3\|d\|\mathcal{C}(d)^2\delta_{bc} \le \theta\mu\,[1 - \mathcal{C}(d)\delta_{bc}] - \theta\mathcal{C}(d)^2\|d\|\delta_{bc}/n,$$

which after rearrangement becomes

$$\mu\,[\theta - \theta_0 - 2n\mathcal{C}(d)\delta_{bc} - \theta\mathcal{C}(d)\delta_{bc}] \ge 3\|d\|\mathcal{C}(d)^2\delta_{bc} + \theta\|d\|\mathcal{C}(d)^2\delta_{bc}/n. \tag{4.23}$$

We have from (4.18) that the coefficient of μ on the left-hand side of this expression is bounded below by ξ. By dividing both sides of (4.23) by this expression and using $\theta \in (0,1)$ and $n \ge 1$, we find that (4.19) is a sufficient condition for (4.23). A similar argument as in the proof of Proposition 4.1, together with the fact that $\mu_2^* > \mu_1^*$, ensures positivity of $(x + \Delta x, s + \Delta s)$. □

We now specialize the discussion of section 3 to show that Propositions 4.1 and 4.2 can be used to obtain lower complexity estimates for the interior-point warm-start strategy.

Considering first the case of Proposition 4.1, we have from the standard analysis of a long-step path-following algorithm that constrains its iterates to lie in $\bar{\mathcal{N}}_{-\infty}(\gamma)$ (see, for example, Wright [16, Chapter 5]) that the reduction in duality measure at each iteration satisfies (3.1) with

$$\tau = 1, \quad \delta = 2^{\frac{3}{2}}\gamma\frac{1-\gamma}{1+\gamma}\min\{\sigma_{\min}(1-\sigma_{\min}), \sigma_{\max}(1-\sigma_{\max})\},$$

where $0 < \sigma_{\min} < \sigma_{\max} < 1$ are the lower and upper bounds on the centering parameter σ at each iteration. Choosing one of the iterates of this algorithm (x^ℓ, y^ℓ, s^ℓ) in the manner of section 3 and defining the starting point as in (3.7), we have from (4.16), (4.13), (4.14), and the conditions $0 < \xi < 1$ and $n \ge 1$ that

$$\begin{aligned}\bar{\mu}_0 &= (\bar{x}^0)^T\bar{s}^0/n \\ &\le \mu_\ell(1 + \mathcal{C}(d)\delta_{bc}) + \mathcal{C}(d)^2\|d\|\delta_{bc}/n \le \mu_\ell(1 + 1/n) + \mu_1^*(\xi/n) \le 2\mu_\ell + \mu_1^*.\end{aligned}$$

Now from the property (3.6), it follows that

$$\bar{\mu}_0 \le (1 + 2/\nu)\mu_1^*.$$

It is easy to verify that (4.13) implies that $\|\Delta d\| \le \|d\|/2$, so that we can use the expression (3.12) to estimate the number of iterations. By substituting $\tau = 1$ and $\mu^* = \mu_1^*$ into (3.12), we obtain

$$\mathcal{O}\left(n\log\left(\frac{1}{\epsilon}\mathcal{C}(d)^2\delta_{bc}\right)\right) \quad \text{iterations.} \tag{4.24}$$

We conclude that if δ_{bc} is small in the sense that $\delta_{bc} \ll \mathcal{C}(d)^{-2}$, then the estimate (4.24) is an improvement on the cold-start complexity estimate (3.4), and thus it is advantageous to use the warm-start strategy.

Taking now the case of a starting point in the smaller neighborhood of Proposition 4.2, we set $\theta = 0.4$, and the centering parameter σ to the constant value $1 - 0.4/n^{1/2}$. The standard analysis of the short-step path-following algorithm (see, for example, [16, Chapter 4]) then shows that (3.1) holds with

$$\tau = 0.5, \qquad \delta = 0.4.$$

By using the procedure outlined in section 3 to derive the warm-start point, the argument of the preceding paragraph can be applied to obtain the following on the number of iterations:

$$\mathcal{O}\left(n^{1/2}\log\left(\frac{1}{\epsilon}\mathcal{C}(d)^2\delta_{bc}\right)\right). \tag{4.25}$$

We conclude as before that improved complexity over a cold start is available, provided that $\delta_{bc} \ll \mathcal{C}(d)^{-2}$.

4.2. Wide neighborhood. We now consider the case in which the iterates for the original problem lie in a wide neighborhood of the central path. To be specific, we suppose that they satisfy $x_i s_i \geq \gamma_0 \mu$ for some $\gamma_0 \in (0,1)$, that is, $(x,y,s) \in \mathcal{N}_{-\infty}(\gamma_0)$. Note that, in this case, we have the following bounds on the pairwise products:

$$\gamma_0\mu \leq x_i s_i \leq (n-(n-1)\gamma_0)\mu. \tag{4.26}$$

Similarly to the upper bounds (2.13) on $\|x\|$ and $\|s\|$, we can derive lower bounds on x_i and s_i by combining (2.13) with (4.26) and using $x_i \leq \|x\|$ and $s_i \leq \|s\|$:

$$x_i \geq \frac{\gamma_0\mu}{2\|d\|\mathcal{C}(d)(\mathcal{C}(d)+n\mu/\|d\|)}, \tag{4.27a}$$

$$s_i \geq \frac{\gamma_0\mu}{\mathcal{C}(d)(\mathcal{C}(d)+n\mu/\|d\|)}. \tag{4.27b}$$

These lower bounds will be useful in the later analysis. The following proposition gives a sufficient condition for the least-squares corrected point to be a member of the wide neighborhood for the perturbed problem. The proof uses an argument identical to the proof of Proposition 4.1, with γ_0 replacing $(1-\theta_0)$.

PROPOSITION 4.3. *Given γ and γ_0 such that $0 < \gamma < \gamma_0 < 1$, suppose that ξ is a parameter satisfying $\xi \in (0, \gamma_0 - \gamma)$. Assume that Δd satisfies*

$$\delta_{bc} \leq \frac{\gamma_0 - \gamma - \xi}{(n+1)\mathcal{C}(d)}. \tag{4.28}$$

Suppose also that $(x,y,s) \in \mathcal{N}_{-\infty}(\gamma_0)$, and denote by $(\Delta x, \Delta y, \Delta s)$ the least-squares correction (4.4)*. Then a sufficient condition for*

$$(x+\Delta x, y+\Delta y, s+\Delta s) \in \bar{\mathcal{N}}_{-\infty}(\gamma) \tag{4.29}$$

is that

$$\mu \geq \frac{\|d\|}{\xi} 3\mathcal{C}(d)^2\delta_{bc} \stackrel{\text{def}}{=} \mu_3^*. \tag{4.30}$$

An argument like the one leading to (4.24) can now be used to show that a long-step path-following method requires

$$\mathcal{O}\left(n\log\left(\frac{1}{\epsilon}\mathcal{C}(d)^2\delta_{bc}\right)\right) \quad \text{iterations} \tag{4.31}$$

to converge from the warm-start point to a point that satisfies (3.9).

4.3. Perturbations in A. We now allow for perturbations in A as well as in b and c. By doing so, we introduce some complications in the analysis that can be circumvented by imposing an a priori upper bound on the values of μ that we are willing to consider. This upper bound is large enough to encompass all values of μ of interest from the viewpoint of complexity, in the sense that when μ exceeds this bound, the warm-start strategy does not lead to an appreciably improved complexity estimate over the cold-start approach.

For some constant $\zeta > 1$, we assume that μ satisfies the bound

$$\mu \leq \frac{\zeta - 1}{n} \|d\| \mathcal{C}(d) \stackrel{\text{def}}{=} \mu_{\text{up}}, \tag{4.32}$$

so that, for a subexpression that recurs often in the preceding sections, we have

$$\mathcal{C}(d) + n\mu/\|d\| \leq \zeta \mathcal{C}(d).$$

For $\mu \in [0, \mu_{\text{up}}]$, we can simplify a number of estimates in the preceding sections, to remove their explicit dependence on μ. In particular, the bounds (2.13) on the strictly feasible point (x, y, s) with $\mu = x^T s/n$ become

$$\|x\| \leq \zeta \mathcal{C}(d)^2, \quad \|y\| \leq \zeta \mathcal{C}(d)^2, \quad \|s\| \leq 2\zeta \|d\| \mathcal{C}(d)^2. \tag{4.33}$$

Given a perturbation $\Delta d = (\Delta A, \Delta b, \Delta c)$ with $\|\Delta d\| < \rho(d)$, we know that $A + \Delta A$ has full rank. In particular, for the smallest singular value, we have

$$\sigma_m(A + \Delta A) \geq \sigma_m(A) - \|\Delta A\|. \tag{4.34}$$

To complement the definitions (4.7), we introduce

$$\delta_A = \frac{\|\Delta A\|}{\|d\|}. \tag{4.35}$$

As before, we consider a warm-start strategy obtained by applying least-squares corrections to a given point (x, y, s) that is strictly feasible for the unperturbed problem. The correction Δx is the solution of the following subproblem:

$$\min \|\Delta x\| \ \text{ s.t. } \ (A + \Delta A)(x + \Delta x) = b + \Delta b, \tag{4.36}$$

which is given explicitly by

$$\Delta x = (A + \Delta A)^T \left[(A + \Delta A)(A + \Delta A)^T\right]^{-1} (\Delta b - \Delta A x), \tag{4.37}$$

where we have used $Ax = b$. By using the QR factorization of $(A + \Delta A)^T$ as in (4.3) and (4.4), we find the following bound on $\|\Delta x\|$:

$$\|\Delta x\| \leq \frac{\|\Delta b\| + \|\Delta A\| \|x\|}{\sigma_m(A + \Delta A)}. \tag{4.38}$$

By using (4.34), (2.5), and the definitions (4.7), (4.35), and (2.3), we have

$$\|\Delta x\| \leq \frac{\|\Delta b\| + \|\Delta A\| \|x\|}{\sigma_m(A) - \|\Delta A\|} \leq \frac{\|\Delta b\| + \|\Delta A\| \|x\|}{\rho(d) - \|\Delta A\|} = \frac{\delta_b + \delta_A \|x\|}{1/\mathcal{C}(d) - \delta_A}.$$

In particular, when x is strictly feasible for the original problem, we have from (4.33) that

$$\|\Delta x\| \leq \mathcal{C}(d)\frac{\delta_b + \zeta\mathcal{C}(d)^2\delta_A}{1-\delta_A\mathcal{C}(d)},$$

while if we make the additional simple assumption that

$$\delta_A \leq \frac{1}{2\mathcal{C}(d)}, \tag{4.39}$$

then we have immediately that

$$\|\Delta x\| \leq 2\mathcal{C}(d)\delta_b + 2\zeta\mathcal{C}(d)^3\delta_A. \tag{4.40}$$

By using (4.39) again, together with (4.10) and the known bounds $\mathcal{C}(d) \geq 1$ and $\zeta > 1$, we obtain

$$\|\Delta x\| \leq 2\mathcal{C}(d)\delta_b + 2\zeta\mathcal{C}(d)^3\delta_A \leq 2\mathcal{C}(d) + \zeta\mathcal{C}(d)^2 \leq 3\zeta\mathcal{C}(d)^2. \tag{4.41}$$

The dual perturbation is the solution of the problem

$$\min \|\Delta s\| \ \text{ s.t. } \ (A+\Delta A)^T(y+\Delta y) + (s+\Delta s) = c + \Delta c. \tag{4.42}$$

Once again, the minimum norm solution is unique and given by

$$\Delta s = \left[I - (A+\Delta A)^T\left((A+\Delta A)(A+\Delta A)^T\right)^{-1}(A+\Delta A)\right](\Delta c - \Delta A^T y). \tag{4.43}$$

Therefore, we have the following upper bound:

$$\|\Delta s\| \leq \|\Delta c\| + \|\Delta A\|\|y\|. \tag{4.44}$$

Using (4.33), we have for (x, y, s) strictly feasible for the original problem that

$$\|\Delta s\| \leq \|\Delta c\| + \|\Delta A\|\zeta\mathcal{C}(d)^2 \leq \|d\|\delta_c + \zeta\|d\|\mathcal{C}(d)^2\delta_A. \tag{4.45}$$

By using these inequalities, we can prove a result similar to Proposition 4.3.

PROPOSITION 4.4. *Suppose we are given γ and γ_0 such that $0 < \gamma < \gamma_0 < 1$, and a feasible primal-dual point $(x, y, s) \in \mathcal{N}_{-\infty}(\gamma_0)$. Assume further that $\mu = x^T s/n$ satisfies* (4.32) *and that the perturbation component ΔA satisfies* (4.39). *For the perturbation Δd, suppose that $(\Delta x, \Delta y, \Delta s)$ is the least-squares correction obtained from* (4.36) *and* (4.42). *We then have*

$$(x+\Delta x, y+\Delta y, s+\Delta s) \in \bar{\mathcal{N}}_{-\infty}(\gamma), \tag{4.46}$$

provided that μ satisfies the following lower bound:

$$\mu \geq 19\zeta\mathcal{C}(d)^2\frac{\|d\|}{\gamma_0-\gamma}\max\left(\delta_{bc}, \zeta\mathcal{C}(d)^3\delta_A\right) \stackrel{\text{def}}{=} \mu_4^*. \tag{4.47}$$

Proof. By using the upper bounds (4.40) and (4.41) on $\|\Delta x\|$, (4.45) on $\|\Delta s\|$, and (4.33) on $\|x\|$ and $\|s\|$, we have

$$\begin{aligned}
&(x_i+\Delta x_i)(s_i+\Delta s_i)\\
&\geq \gamma_0\mu - (\|x\|+\|\Delta x\|)\|\Delta s\| - \|s\|\|\Delta x\|\\
&\geq \gamma_0\mu - [4\zeta\mathcal{C}(d)^2][\|d\|\delta_c + \zeta\|d\|\mathcal{C}(d)^2\delta_A]\\
&\qquad - [2\|d\|\zeta\mathcal{C}(d)^2][2\mathcal{C}(d)\delta_b + 2\zeta\mathcal{C}(d)^3\delta_A]\\
&\geq \gamma_0\mu - 4\|d\|\zeta\mathcal{C}(d)^3\delta_b - 4\|d\|\zeta\mathcal{C}(d)^2\delta_c - 8\|d\|\zeta^2\mathcal{C}(d)^5\delta_A\\
&\geq \gamma_0\mu - 4\|d\|\zeta\mathcal{C}(d)^2\delta_{bc} - 8\|d\|\zeta^2\mathcal{C}(d)^5\delta_A,
\end{aligned}$$

where for the last inequality we have used the definition (4.9). By similar logic, and using (4.5), we have for the updated duality measure that

$$\begin{aligned}
&(x+\Delta x)^T(s+\Delta s)/n \\
&\le \mu + \|\Delta x\|\,\|s\|/n + \|x\|\,\|\Delta s\|/n \\
&\le \mu + [2\mathcal{C}(d)\delta_b + 2\zeta\mathcal{C}(d)^3\delta_A]2\zeta\|d\|\mathcal{C}(d)^2/n + \zeta\mathcal{C}(d)^2[\|d\|\delta_c + \zeta\|d\|\mathcal{C}(d)^2\delta_A]/n \\
&= \mu + 4\zeta\mathcal{C}(d)^3\|d\|\delta_b/n + \zeta\mathcal{C}(d)^2\|d\|\delta_c/n + 5\zeta^2\mathcal{C}(d)^5\|d\|\delta_A/n \\
&\le \mu + 2\zeta\mathcal{C}(d)^2\|d\|\delta_{bc}/n + 5\zeta^2\mathcal{C}(d)^5\|d\|\delta_A/n.
\end{aligned}$$

By comparing these two inequalities in the usual way and using $\gamma \in (0,1)$ and $n \ge 1$, we have that a sufficient condition for the conclusion (4.46) to hold is that

$$(\gamma_0 - \gamma)\mu \ge 6\|d\|\zeta\mathcal{C}(d)^2\delta_{bc} + 13\|d\|\zeta^2\mathcal{C}(d)^5\delta_A. \tag{4.48}$$

Since from (4.47), we have

$$\begin{aligned}
\frac{6}{19}(\gamma_0 - \gamma)\mu &\ge 6\|d\|\zeta\mathcal{C}(d)^2\delta_{bc}, \\
\frac{13}{19}(\gamma_0 - \gamma)\mu &\ge 13\|d\|\zeta^2\mathcal{C}(d)^5\delta_A,
\end{aligned}$$

then (4.48) holds. Finally, the positivity of $x + \Delta x$ and that of $s + \Delta s$ can be shown in a way similar to the proof of Proposition 4.1. Once again, the lower bound for $(x_i + \Delta x_i)(s_i + \Delta s_i)$ also holds for $(x_i + \alpha\Delta x_i)(s_i + \alpha\Delta s_i)$ for any $\alpha \in [0,1]$. Using the simple inequality $a + b \le 2\max(a,b)$, we obtain

$$(x_i + \alpha\Delta x_i)(s_i + \alpha\Delta s_i) \ge \gamma_0\mu - 8\zeta\mathcal{C}(d)^2\|d\|\max\left(\delta_{bc}, 2\zeta\mathcal{C}(d)^3\delta_A\right),$$

which yields a positive lower bound by (4.47), and the proof is complete. ∎

By using an argument like the ones leading to (4.24) and (4.31), we deduce that a long-step path-following algorithm that uses the warm start prescribed in Proposition 4.4 requires

$$\mathcal{O}\left(n\left[\log\left(\frac{1}{\epsilon}\mathcal{C}(d)^2\delta_{bc}\right) + \log\left(\frac{1}{\epsilon}\mathcal{C}(d)^5\delta_A\right)\right]\right) \quad \text{iterations} \tag{4.49}$$

to converge to a point that satisfies (3.9).

5. Newton step correction. In a recent study, Yıldırım and Todd [18] analyzed the perturbations in b and c in linear and semidefinite programming using interior-point methods. For such perturbations they stated a sufficient condition on the norm of the perturbation, which depends on the current iterate, so that an adjustment to the current point based on applying an iteration of Newton's method to the system (2.6a), (2.6b), (2.6c) yields a feasible iterate for the perturbed problem with a lower duality gap than that of the original iterate. In this section, we augment some of the analysis of [18] with other results, like those of section 4, to find conditions on the duality gap $\mu = x^T s/n$ and the perturbation size under which the Newton step yields a warm-start point that gives significantly better complexity than a cold start.

Each iteration of a primal-dual interior-point method involves solving a Newton-like system of linear equations whose coefficient matrix is the Jacobian of the system (2.6a), (2.6b), (2.6c). The general form of these equations is

$$\begin{array}{rcccccl}
 & A\Delta x & & & = & r_p, \\
A^T\Delta y & & + & \Delta s & = & r_d, \\
 & S\Delta x & + & X\Delta s & = & r_{xs},
\end{array} \tag{5.1}$$

where typically $r_p = b - Ax$ and $r_d = c - A^T y - s$. The choice of r_{xs} typically depends on the particular method being applied, but usually represents a Newton or higher-order step toward some "target point" (x', y', s'), which often lies on the central path $\mathcal{P}$ defined in (2.7).

In the approach used in Yıldırım and Todd [18] and in this section, this Newton-like system is used to correct for perturbations in the data (A, b, c) rather than to advance to a new primal-dual iterate. The right-hand side quantities are chosen so that the adjustment $(\Delta x, \Delta y, \Delta s)$ yields a point that is strictly feasible for the perturbed problem and whose duality gap is no larger than that of the current point (x, y, s).

In section 5.1, we consider the case of perturbations in b and c but not in A. In section 5.2 we allow perturbations in A as well.

5.1. Perturbations in *b* and *c*. In our strategy, we assume that

- the current point (x, y, s) is strictly primal-dual feasible for the original problem;
- the target point (x', y', s') used to define r_{xs} is a point that is strictly feasible for the perturbed problem for which $x'_i s'_i = x_i s_i$ for all $i = 1, 2, \ldots, n$;
- the step is a pure Newton step toward (x', y', s'); that is, $r_p = \Delta b$, $r_d = \Delta c$, and $r_{xs} = X'S'e - XSe = 0$.

Note that, in general, the second assumption is not satisfied for an arbitrary current point (x, y, s) because such a feasible point for the perturbed problem need not exist. However, Newton's method is still well defined with the above choices of r_p, r_d, and r_{xs}, and that assumption is merely stated for the sake of a complete description of our strategy.

Since A has full row rank by our assumption of $\rho(d) > 0$, we have, by substituting our right-hand side in (5.1) and performing block elimination, that the solution is given explicitly by

$$\Delta y = (AD^2A^T)^{-1}(\Delta b + AD^2\Delta c), \tag{5.2a}$$

$$\Delta s = \Delta c - A^T \Delta y, \tag{5.2b}$$

$$\Delta x = -S^{-1}X\Delta s, \tag{5.2c}$$

where

$$D^2 \stackrel{\text{def}}{=} S^{-1}X. \tag{5.3}$$

Since A has full row rank and D is positive diagonal, AD^2A^T is invertible.

The following is an extension of the results in Yıldırım and Todd [18] to the case of simultaneous perturbations in b and c. Note in particular that the Newton step yields a decrease in the duality gap $x^T s$.

PROPOSITION 5.1. *Assume that (x, y, s) is a strictly feasible point for d. Let $\Delta d = (0, \Delta b, \Delta c)$. Consider a Newton step $(\Delta x, \Delta y, \Delta s)$ taken from (x, y, s) targeting the point (x', y', s') that is strictly feasible for the perturbed problem and satisfies $X'S'e = XSe$, and let*

$$(\tilde{x}, \tilde{y}, \tilde{s}) \stackrel{\text{def}}{=} (x, y, s) + (\Delta x, \Delta y, \Delta s). \tag{5.4}$$

Then if

$$\left\| \begin{bmatrix} \Delta c \\ \Delta b \end{bmatrix} \right\|_\infty \leq \left\| \begin{bmatrix} S^{-1}\left(I - A^T(AD^2A^T)^{-1}AD^2\right) & -S^{-1}A^T(AD^2A^T)^{-1} \end{bmatrix} \right\|_\infty^{-1}, \tag{5.5}$$

$(\tilde{x}, \tilde{y}, \tilde{s})$ is feasible for the perturbed problem and satisfies

$$\tilde{x}^T \tilde{s} \leq x^T s. \tag{5.6}$$

Proof. By rearranging (5.2c) and writing it componentwise, we have

$$s_i \Delta x_i + x_i \Delta s_i = 0 \Longleftrightarrow \frac{\Delta x_i}{x_i} + \frac{\Delta s_i}{s_i} = 0, \qquad i = 1, 2, \ldots, n. \tag{5.7}$$

The next iterate will be feasible if and only if

$$\frac{\Delta x_i}{x_i} \geq -1, \quad \frac{\Delta s_i}{s_i} \geq -1, \quad i = 1, 2, \ldots, n.$$

By combining these inequalities with (5.7), we find that feasibility requires

$$\left| \frac{\Delta x_i}{x_i} \right| \leq 1, \quad \left| \frac{\Delta s_i}{s_i} \right| \leq 1, \quad i = 1, 2, \ldots, n,$$

or, equivalently,

$$\left\| S^{-1} \Delta s \right\|_\infty = \left\| X^{-1} \Delta x \right\|_\infty \leq 1. \tag{5.8}$$

By using (5.2a) and (5.2c), we have

$$\begin{aligned}
&\|S^{-1}\Delta s\|_\infty \\
&\quad = \left\|S^{-1}[\Delta c - A^T \Delta y]\right\|_\infty \\
&\quad = \left\|S^{-1}\left[\Delta c - A^T (AD^2A^T)^{-1} AD^2 \Delta c - A^T (AD^2A^T)^{-1} \Delta b\right]\right\|_\infty \\
&\quad \leq \left\|\left[S^{-1}\left(I - A^T (AD^2A^T)^{-1} AD^2\right) \quad - S^{-1}A^T (AD^2A^T)^{-1}\right]\right\|_\infty \left\| \begin{bmatrix} \Delta c \\ \Delta b \end{bmatrix} \right\|_\infty .
\end{aligned} \tag{5.9}$$

Hence, (5.5) is sufficient to ensure that $\|S^{-1}\Delta s\|_\infty \leq 1$.

By summing (5.7) over $i = 1, 2, \ldots, n$, we obtain $x^T \Delta s + s^T \Delta x = 0$. It is also clear from (5.7) that Δx_i and Δs_i have opposite signs for each $i = 1, 2, \ldots, n$, and thus $\Delta x^T \Delta s \leq 0$. Therefore, we have

$$(x + \Delta x)^T (s + \Delta s) = x^T s + x^T \Delta s + s^T \Delta x + \Delta x^T \Delta s = x^T s + \Delta x^T \Delta s \leq x^T s,$$

proving (5.6). ∎

Proposition 5.1 does not provide any insight about the behavior of the expression on the right-hand side of (5.5) as a function of μ. To justify our strategy of retreating to successively earlier iterates of the original problem, we need to show that the expression in question increases as μ corresponding to (x, y, s) increases, so that we can handle larger perturbations by considering iterates with larger values of μ. In the next theorem, we will show that there exists an increasing function $f(\mu)$ with $f(0) = 0$ that is a lower bound to the corresponding expression in (5.5) for all values of μ. The key to our result is the following bound:

$$\chi(H) \stackrel{\text{def}}{=} \sup_{\Sigma \in \mathcal{D}_+} \left\| \Sigma H^T (H \Sigma H^T)^{-1} \right\|_\infty < \infty, \tag{5.10}$$

where $\mathcal{D}_+$ denotes the set of diagonal matrices in $R^{n \times n}$ with strictly positive diagonal elements (i.e., positive definite diagonal matrices) and $\|\cdot\|_\infty$ is the ℓ_∞ matrix norm

defined as the maximum of the sums of the absolute values of the entries in each row. This result, by now well known, was apparently first proved by Dikin [2]. For a survey of the background and applications of this and related results, see Forsgren [3].

THEOREM 5.2. *Consider points (x, y, s) in the neighborhood $\mathcal{N}_{-\infty}(\gamma_0)$ for the original problem, with $\gamma_0 \in (0, 1)$ and $\mu = x^T s/n$ as defined in (2.11). Then there exists an increasing function $f(\mu)$ with $f(0) = 0$ such that the expression on the right-hand side of (5.5) is bounded below by $f(\mu)$ for all (x, y, s) in this neighborhood.*

Proof. Let (x, y, s) be a strictly feasible pair of points for the original problem, which lies in $\mathcal{N}_{-\infty}(\gamma_0)$ for some $\gamma_0 \in (0, 1)$. From (4.27) and (5.10), we have

$$\begin{aligned} \left\|S^{-1}A^T(AD^2A^T)^{-1}\right\|_\infty &= \left\|S^{-1}D^{-2}D^2A^T(AD^2A^T)^{-1}\right\|_\infty \\ &\le \left\|X^{-1}\right\|_\infty \left\|D^2A^T(AD^2A^T)^{-1}\right\|_\infty \\ &\le \left(\frac{1}{\mu}\right)\frac{2\|d\|\mathcal{C}(d)}{\gamma_0}\left(\mathcal{C}(d) + n\mu/\|d\|\right)\chi(A). \end{aligned} \tag{5.11}$$

The first inequality is simply the matrix norm inequality. Since $D^2 = XS^{-1}$, and x and s are strictly feasible, D^2 is a positive definite diagonal matrix, and thus the bound in (5.10) applies.

Similarly, consider the following:

$$\begin{aligned} &\left\|S^{-1}\left(I - A^T(AD^2A^T)^{-1}AD^2\right)\right\|_\infty \\ &= \left\|S^{-1}D^{-1}\left(I - DA^T(AD^2A^T)^{-1}AD\right)D\right\|_\infty. \end{aligned} \tag{5.12}$$

Note that $(I - DA^T(AD^2A^T)^{-1}AD)$ is a projection matrix onto the null space of AD; therefore, its ℓ_2-norm is bounded by 1. Using the elementary matrix norm inequality $\|P\|_\infty \le n^{1/2}\|P\|_2$ for any $P \in R^{n\times n}$, we obtain the following sequence of inequalities:

$$\begin{aligned} &\left\|S^{-1}\left(I - A^T(AD^2A^T)^{-1}AD^2\right)\right\|_\infty \\ &= \left\|S^{-1}D^{-1}(I - DA^T(AD^2A^T)^{-1}AD)D\right\|_\infty \\ &\le \left\|X^{-1/2}S^{-1/2}\right\|_\infty \left\|I - DA^T(AD^2A^T)^{-1}AD\right\|_\infty \left\|X^{1/2}S^{-1/2}\right\|_\infty \\ &\le \max_{i=1,2,\ldots,n}\frac{1}{\sqrt{x_i s_i}}\, n^{1/2} \max_{i=1,2,\ldots,n}\sqrt{\frac{x_i}{s_i}} \\ &\le n^{1/2}\frac{1}{\sqrt{\gamma_0\mu}}\max_{i=1,2,\ldots,n}\frac{x_i}{\sqrt{x_i s_i}} \\ &\le \left(\frac{1}{\mu}\right)\frac{n^{1/2}\mathcal{C}(d)}{\gamma_0}\left(\mathcal{C}(d) + n\mu/\|d\|\right), \end{aligned} \tag{5.13}$$

where we used $D^2 = XS^{-1}$, $x_i s_i \ge \gamma_0\mu$, and (2.13).

If we consider the reciprocal of the right-hand side of expression (5.5), we obtain

$$\begin{aligned} &\left\|\left[S^{-1}\left(I - A^T(AD^2A^T)^{-1}AD^2\right) \quad - S^{-1}A^T(AD^2A^T)^{-1}\right]\right\|_\infty \\ &\le \left\|S^{-1}\left(I - A^T(AD^2A^T)^{-1}AD^2\right)\right\|_\infty + \left\|S^{-1}A^T(AD^2A^T)^{-1}\right\|_\infty \\ &\le \left(\frac{1}{\mu}\right)\frac{n^{1/2}\mathcal{C}(d)}{\gamma_0}\left(\mathcal{C}(d) + \{n\mu\}\|d\|\right) + \left(\frac{1}{\mu}\right)\frac{2\|d\|\mathcal{C}(d)}{\gamma_0}\left(\mathcal{C}(d) + n\mu/\|d\|\right)\chi(A), \end{aligned} \tag{5.14}$$

which follows from (5.11) and (5.13). Therefore, (5.14) implies

$$\begin{aligned}&\frac{1}{\|[S^{-1}\,(I - A^T(AD^2A^T)^{-1}AD^2) \quad - S^{-1}A^T(AD^2A^T)^{-1}]\|_\infty}\\ &\geq\ f(\mu) \stackrel{\text{def}}{=} \frac{\gamma_0\mu}{\mathcal{C}(d)\left(n^{1/2} + 2\|d\|\chi(A)\right)\left[\mathcal{C}(d) + n\mu/\|d\|\right]}.\end{aligned} \tag{5.15}$$

It is easy to verify our claims both that f is monotone increasing in μ and that $f(0) = 0$. ∎

Note that Proposition 5.1 guarantees only that the point $(\tilde{x}, \tilde{y}, \tilde{s})$ is feasible for the perturbed problem. To initiate a feasible path-following interior-point method, we need to impose additional conditions to obtain a strictly feasible point for the perturbed problem that lies in some neighborhood of the central path. For example, in the proof, we imposed only the condition $(\tilde{x}, \tilde{s}) \geq 0$. Strict positivity of $\tilde{x}$ and $\tilde{s}$ could be ensured by imposing the following condition, for some $\epsilon \in (0, 1)$:

$$x_i + \Delta x_i \geq \epsilon x_i, \qquad s_i + \Delta s_i \geq \epsilon s_i \quad \forall\ i = 1, 2, \ldots, n. \tag{5.16}$$

Equivalently, we can replace the necessary and sufficient condition $\|S^{-1}\Delta s\|_\infty \leq 1$ in (5.8) by the condition $(\epsilon - 1)e \leq S^{-1}\Delta s \leq (1 - \epsilon)e$, that is,

$$\|S^{-1}\Delta s\|_\infty \leq 1 - \epsilon,$$

in the proof of Proposition 5.1. With this requirement, we obtain the following bounds:

$$\epsilon x_i \leq \tilde{x}_i \leq (2 - \epsilon)x_i, \qquad \epsilon s_i \leq \tilde{s}_i \leq (2 - \epsilon)s_i. \tag{5.17}$$

Note that if $(\Delta x, \Delta y, \Delta s)$ is the Newton step given by (5.2), then $\Delta x_i \Delta s_i \leq 0$ for all $i = 1, 2, \ldots, n$. First, consider the case $\Delta x_i \geq 0$, which implies $\tilde{x}_i \geq x_i$. We have from (5.17) that

$$\tilde{x}_i\tilde{s}_i \geq x_i\tilde{s}_i \geq \epsilon x_i s_i. \tag{5.18}$$

A similar set of inequalities holds for the case $\Delta s_i \geq 0$. Thus, if we define $\tilde{\mu} = \tilde{x}^T\tilde{s}/n$, we obtain

$$\tilde{\mu} \geq \epsilon\mu. \tag{5.19}$$

Note that by (5.6), we already have $\tilde{\mu} \leq \mu$. With this observation, we can relate the neighborhood in which the original iterate (x, y, s) lies to the one in which the adjusted point $(\tilde{x}, \tilde{y}, \tilde{s})$ lies.

PROPOSITION 5.3. *Let (x, y, s) be a strictly feasible point for d, and suppose that $\Delta d = (0, \Delta b, \Delta c)$ and $\epsilon \in (0, 1)$ are given. Consider the Newton step of Proposition* 5.1 *and the adjusted point $(\tilde{x}, \tilde{y}, \tilde{s})$ of* (5.4). *If*

$$\begin{aligned}&\left\|\begin{bmatrix}\Delta c\\ \Delta b\end{bmatrix}\right\|_\infty\\ &\leq \frac{1 - \epsilon}{\|[S^{-1}(I - A^T(AD^2A^T)^{-1}AD^2) \quad - S^{-1}A^T(AD^2A^T)^{-1}]\|_\infty},\end{aligned} \tag{5.20}$$

with D defined in (5.3), *then $(\tilde{x}, \tilde{y}, \tilde{s})$ is strictly feasible for $d + \Delta d$ with $\tilde{\mu} \leq \mu$. Moreover, if $(x, y, s) \in \mathcal{N}_{-\infty}(\gamma_0)$ for the original problem with $\gamma_0 \in (0, 1)$, then $(\tilde{x}, \tilde{y}, \tilde{s})$ satisfies $(\tilde{x}, \tilde{y}, \tilde{s}) \in \mathcal{N}_{-\infty}(\epsilon\gamma_0)$.*

Proof. It suffices to prove the final statement of the theorem. If we assume that $(x, y, s) \in \mathcal{N}_{-\infty}(\gamma_0)$, then, using (5.18) and (5.6), we have

$$\tilde{x}_i \tilde{s}_i \geq \epsilon x_i s_i \geq \epsilon \gamma_0 \mu \geq \epsilon \gamma_0 \tilde{\mu}, \tag{5.21}$$

which implies that $(\tilde{x}, \tilde{y}, \tilde{s}) \in \bar{\mathcal{N}}_{-\infty}(\epsilon\gamma_0)$, as required. ∎

We now have all the tools to be able to prove results like those of section 4. Suppose that the iterates of the original problem lie in a wide neighborhood with parameter γ_0. For convenience we define

$$\|\Delta d\|_\infty \stackrel{\text{def}}{=} \left\| \begin{bmatrix} \Delta b \\ \Delta c \end{bmatrix} \right\|_\infty = \max(\|\Delta b\|_\infty, \|\Delta c\|_\infty). \tag{5.22}$$

We also define the relative perturbation measure δ_d as follows:

$$\delta_d \stackrel{\text{def}}{=} \frac{\|\Delta d\|_\infty}{\|d\|}. \tag{5.23}$$

Note from (4.7) and (4.9) that

$$\delta_d = \max\left(\frac{\|\Delta b\|_\infty}{\|d\|}, \frac{\|\Delta c\|_\infty}{\|d\|} \right) \leq \max(\delta_b, \delta_c) \leq \delta_{bc}.$$

Hence, it is easy to compare results such as Proposition 5.4 below, which obtain a lower bound on μ in terms of δ_d, to similar results in preceding sections.

Note that Theorem 5.2 provides a lower bound $f(\mu)$ on the term on the right-hand side of (5.5). Therefore, combining this result with Proposition 5.3, we conclude that a sufficient condition for the perturbation Δd to satisfy (5.20) is that $\|\Delta d\|_\infty$ is bounded above by the lower bound (5.15) multiplied by $(1 - \epsilon)$, that is,

$$\|\Delta d\|_\infty \leq \frac{(1-\epsilon)\gamma_0 \mu}{\mathcal{C}(d)\left(n^{1/2} + 2\|d\|\chi(A)\right)\left(\mathcal{C}(d) + n\mu/\|d\|\right)},$$

which by rearrangement yields

$$\mu \geq \frac{\mathcal{C}(d)^2 \|\Delta d\|_\infty (n^{1/2} + 2\|d\|\chi(A))}{(1-\epsilon)\gamma_0 - n\mathcal{C}(d)\|\Delta d\|_\infty (n^{1/2} + 2\|d\|\chi(A))/\|d\|}, \tag{5.24}$$

provided that the denominator of this expression is positive. To ensure the latter condition, we impose the following bound on δ_d:

$$\delta_d = \frac{\|\Delta d\|_\infty}{\|d\|} < \frac{(1-\epsilon)\gamma_0}{n\mathcal{C}(d)(n^{1/2} + 2\|d\|\chi(A))}. \tag{5.25}$$

Indeed, when this bound is not satisfied, the perturbation may be so large that the adjusted point $(\tilde{x}, \tilde{y}, \tilde{s})$ may not be feasible for $d + \Delta d$ no matter how large we choose μ for the original iterate (x, y, s).

We now state and prove a result like Proposition 4.3 that gives a condition on $\|\Delta d\|_\infty$ and μ sufficient to ensure that the adjusted point $(\tilde{x}, \tilde{y}, \tilde{s})$ lies within a wide neighborhood of the central path for the perturbed problem.

PROPOSITION 5.4. *Let γ and γ_0 be given with $0 < \gamma < \gamma_0 < 1$, and suppose that ξ satisfies $\xi \in (0, \gamma_0 - \gamma)$. Assume that δ_d satisfies*

$$\delta_d \leq \frac{\gamma_0 - \gamma - \xi}{n\mathcal{C}(d)(n^{1/2} + 2\|d\|\chi(A))}. \tag{5.26}$$

Suppose that $(x, y, s) \in \mathcal{N}_{-\infty}(\gamma_0)$ *for the original problem, and let* $(\tilde{x}, \tilde{y}, \tilde{s})$ *be as defined in* (5.4). *Then if*

$$\mu \geq \frac{\|d\|}{\xi} \mathcal{C}(d)^2 \delta_d \left(n^{1/2} + 2\|d\| \chi(A) \right) \stackrel{\text{def}}{=} \mu_5^*, \tag{5.27}$$

we have $(\tilde{x}, \tilde{y}, \tilde{s}) \in \bar{\mathcal{N}}_{-\infty}(\gamma)$.

Proof. Setting $\epsilon = \gamma/\gamma_0$, we note that (5.26) satisfies condition (5.25), and so the Newton step adjustment yields a strictly feasible point for the perturbed problem. By the argument preceding the proposition, (5.24) gives a condition sufficient for the resulting iterate to lie in $\bar{\mathcal{N}}_{-\infty}(\gamma)$ by Proposition 5.3 since $\gamma = \epsilon\gamma_0$ by the hypothesis. However, (5.26) implies that the denominator of (5.24) is bounded below by ξ; hence, (5.24) is implied by (5.27), as required. ∎

The usual argument can now be used to show that a long-step path-following method requires

$$\mathcal{O}\left(n \log \left(\frac{1}{\epsilon} \mathcal{C}(d)^2 \delta_d \left(n^{1/2} + \|d\| \chi(A) \right) \right) \right) \quad \text{iterations} \tag{5.28}$$

to converge from the warm-start point to a point that satisfies (3.9).

5.2. Perturbations in A. In this section, we also allow perturbations in A (i.e., we let $\Delta d = (\Delta A, \Delta b, \Delta c)$) and propose a Newton step correction strategy to recover warm-start points for the perturbed problem from the iterates of the original problem.

The underlying idea is the same as that in section 5.1. Given a strictly feasible iterate $(x, y, s) \in \mathcal{N}_{-\infty}(\gamma_0)$ for the original problem, we apply Newton's method to recover a feasible point for the perturbed problem by keeping the pairwise products $x_i s_i$ fixed. As in section 4.3, we impose an upper bound on μ that excludes values of μ that are not likely to yield an adjusted starting point with significantly better complexity than a cold-start strategy. In particular, we assume that μ satisfies (4.32) for some $\zeta > 1$. Let

$$\bar{A} \stackrel{\text{def}}{=} A + \Delta A. \tag{5.29}$$

Given a feasible iterate (x, y, s) for the original problem, the Newton step correction is then given by the solution to

$$\begin{array}{rcrcrcl} & & \bar{A}\Delta x & & & = & \Delta b - \Delta A x, \\ \bar{A}^T \Delta y & & & + & \Delta s & = & \Delta c - \Delta A^T y, \\ & & S\Delta x & + & X\Delta s & = & 0. \end{array} \tag{5.30}$$

Under the assumption that $\bar{A}$ has full row rank, the solution to (5.30) is then given by

$$\Delta y = (\bar{A} D^2 \bar{A}^T)^{-1} (\bar{A} D^2 (\Delta c - \Delta A^T y) + \Delta b - \Delta A x), \tag{5.31a}$$

$$\Delta s = \Delta c - \Delta A^T y - \bar{A}^T \Delta y, \tag{5.31b}$$

$$\Delta x = -S^{-1} X \Delta s, \tag{5.31c}$$

where $D^2 = S^{-1}X$ as in (5.3).

By a similar argument, a necessary and sufficient condition to have strictly feasible iterates for the perturbed problem is

$$\|S^{-1}\Delta s\|_\infty \leq 1 - \epsilon \quad \text{for some } \epsilon \in (0, 1). \tag{5.32}$$

By Proposition 5.3, the duality gap of the resulting iterate will also be smaller than that of the original iterate. We will modify the analysis in section 5 to incorporate the perturbation in A and will refer to the previous analysis without repeating the propositions.

Using (5.31), we get

$$\begin{aligned} S^{-1}\Delta s = S^{-1}(I - \bar{A}^T(\bar{A}D^2\bar{A}^T)^{-1}\bar{A}D^2)(\Delta c - \Delta A^T y) \\ - S^{-1}\bar{A}^T(\bar{A}D^2\bar{A}^T)^{-1}(\Delta b - \Delta A x). \end{aligned}$$

Therefore, $\|S^{-1}\Delta s\|_\infty$ is bounded above by

$$\left\|[S^{-1}(I - \bar{A}^T(\bar{A}D^2\bar{A}^T)^{-1}\bar{A}D^2) \quad - S^{-1}\bar{A}^T(\bar{A}D^2\bar{A}^T)^{-1}]\right\|_\infty \left\| \begin{bmatrix} \Delta c - \Delta A^T y \\ \Delta b - \Delta A x \end{bmatrix} \right\|_\infty .$$

By Theorem 5.2, the first term in this expression is bounded above by $1/\bar{f}(\mu)$, where $\bar{f}(\mu)$ is obtained from $f(\mu)$ in (5.15) by replacing $\chi(A)$ by $\chi(\bar{A})$. For the second term, we extend the definition in (5.22) to account for the perturbations in A as follows,

$$\|\Delta d\|_\infty \stackrel{\text{def}}{=} \max(\|\Delta b\|_\infty, \|\Delta c\|_\infty, \|\Delta A\|_\infty, \|\Delta A^T\|_\infty), \tag{5.33}$$

and continue to define δ_d as in (5.23). We obtain that

$$\begin{aligned} &\left\| \begin{bmatrix} \Delta c - \Delta A^T y \\ \Delta b - \Delta A x \end{bmatrix} \right\|_\infty \\ &\le \max\{\|\Delta c\|_\infty + \|\Delta A^T\|_\infty \|y\|_\infty, \|\Delta b\|_\infty + \|\Delta A\|_\infty \|x\|_\infty\} \\ &\le \max\{\|\Delta d\|_\infty(1 + \|y\|_\infty), \|\Delta d\|_\infty(1 + \|x\|_\infty)\} \\ &\le \|\Delta d\|_\infty(1 + \zeta\mathcal{C}(d)^2) \\ &\le 2\|\Delta d\|_\infty \zeta\mathcal{C}(d)^2, \end{aligned} \tag{5.34}$$

where we used (5.33), (4.33), $\zeta > 1$, and $\mathcal{C}(d) \ge 1$ to derive the inequalities. By combining the two upper bounds we obtain

$$\|S^{-1}\Delta s\|_\infty \le \left(\frac{1}{\mu}\right) \frac{1}{\gamma_0} 2\zeta\mathcal{C}(d)^3 \left(n^{1/2} + 2\|d\|\chi(\bar{A})\right) (\mathcal{C}(d) + n\mu/\|d\|) \|\Delta d\|_\infty. \tag{5.35}$$

Therefore, a sufficient condition to ensure (5.32) is obtained by requiring the upper bound in (5.35) to be less than $1 - \epsilon$. Rearranging the resulting inequality yields a lower bound on μ,

$$\mu \ge \frac{2\zeta\mathcal{C}(d)^4 \left(n^{1/2} + 2\|d\|\chi(\bar{A})\right) \|\Delta d\|_\infty}{\gamma_0(1-\epsilon) - 2\zeta n\mathcal{C}(d)^3 \left(n^{1/2} + 2\|d\|\chi(\bar{A})\right) \|\Delta d\|_\infty/\|d\|}, \tag{5.36}$$

provided that the denominator is positive, which is ensured by the condition

$$\delta_d = \frac{\|\Delta d\|_\infty}{\|d\|} < \frac{\gamma_0(1-\epsilon)}{2\zeta n\mathcal{C}(d)^3 \left(n^{1/2} + 2\|d\|\chi(\bar{A})\right)}. \tag{5.37}$$

The proof of the following result is similar to that of Proposition 5.4.

PROPOSITION 5.5. *Let γ and γ_0 be given with $0 < \gamma < \gamma_0 < 1$, and suppose that ξ satisfies $\xi \in (0, \gamma_0 - \gamma)$. Assume that Δd satisfies*

$$\delta_d \le \frac{\gamma_0 - \gamma - \xi}{2\zeta n\mathcal{C}(d)^3 \left(n^{1/2} + 2\|d\|\chi(\bar{A})\right)}. \tag{5.38}$$

Suppose that $(x, y, s) \in \mathcal{N}_{-\infty}(\gamma_0)$ *and that* $(\tilde{x}, \tilde{y}, \tilde{s})$ *is the adjusted point defined in* (5.4). *Then we have* $(\tilde{x}, \tilde{y}, \tilde{s}) \in \mathcal{N}_{-\infty}(\gamma)$, *provided that*

$$\mu \geq \frac{\|d\|}{\xi} 2\zeta \mathcal{C}(d)^4 \delta_d \left(n^{1/2} + 2\|d\| \chi(\bar{A}) \right) \stackrel{\text{def}}{=} \mu_6^*. \tag{5.39}$$

The usual argument can be used again to show that a long-step path-following method requires

$$\mathcal{O}\left(n \log \left(\frac{1}{\epsilon} \mathcal{C}(d)^4 \delta_d \left(n^{1/2} + \|d\| \chi(\bar{A}) \right) \right) \right) \quad \text{iterations} \tag{5.40}$$

to converge from the warm-start point to a point that satisfies (3.9).

6. Comparison of the strategies. Here we comment on the relationship between the Newton step correction strategy of section 5, the least-squares correction strategy of section 4, and a weighted least-squares approach described below. In particular, we discuss the effects of weighting in different situations and show that the strategies of sections 4 and 5 jointly retain all the benefits of the weighted least-squares strategy. The weighted least-squares strategy is discussed in section 6.1, relationships between the strategies in various circumstances are discussed in section 6.2, and some numerical results are presented in section 6.3.

6.1. Weighted least-squares strategy. When the data perturbations are confined to the vectors b and c, we can define $n \times n$ positive diagonal matrices Σ and Λ and solve the following variants on the subproblems (4.2):

$$\min \|\Sigma \Delta x\| \quad \text{s.t. } A\Delta x = \Delta b, \tag{6.1a}$$

$$\min \|\Lambda \Delta s\| \quad \text{s.t. } A^T \Delta y + \Delta s = \Delta c. \tag{6.1b}$$

The solutions are as follows:

$$\Delta x_\Sigma = \Sigma^{-2} A^T (A\Sigma^{-2} A^T)^{-1} \Delta b, \tag{6.2a}$$

$$\Delta y_\Lambda = (A\Lambda^2 A^T)^{-1} A\Lambda^2 \Delta c, \tag{6.2b}$$

$$\Delta s_\Lambda = (I - A^T (A\Lambda^2 A^T)^{-1} A\Lambda^2) \Delta c. \tag{6.2c}$$

When $\Sigma = \Lambda = I$, we recover the least-squares solutions (4.4). Alternative scalings include the following:

$$\Sigma = X^{-1}, \quad \Lambda = S^{-1} \tag{6.3}$$

and

$$\Sigma = D^{-1} = X^{-1/2} S^{1/2}, \quad \Lambda = D = X^{1/2} S^{-1/2}. \tag{6.4}$$

The second scaling is of particular interest, as a comparison of (6.2) with the substitutions (6.4) yields corrections quite similar to (5.2). The difference arises from the fact that the Newton step contains an additional condition that couples Δx and Δs; namely, $X\Delta s + S\Delta x = 0$. If the perturbation is confined to b (that is, $\Delta c = 0$), then the correction in x given by the Newton step scheme (5.2) is the same as the one obtained from (6.2a) with Σ as defined in (6.4). In this case, the Newton step correction reduces to a weighted least-squares correction.

In fact, Mitchell and Todd [9] use a similar weighted least-squares strategy in a column-generation framework. Assuming that $\tilde{x}$ is feasible for problem (P), a new column a is introduced so that $(\tilde{x}, 0)$ is feasible for the new problem. Then, in order to obtain a strictly feasible point to restart the interior-point algorithm, a step is taken in the direction given by $(d, 1)$, where d solves (6.1a) with $\Sigma = X^{-1}$ and $\Delta b = -a$. The reader is also referred to [7] and the references therein for consideration of other directions.

The weighted least-squares approach suffers some disadvantages relative to the approaches of sections 4 and 5. When the weighting matrices Σ and Λ depend on x and s, the solutions (6.2) must be computed afresh for each candidate initial point, whereas for unweighted least-squares, a single solution suffices (4.4). Unlike the Newton step, the weighted least-squares approach does not guarantee a smaller value of μ than at the initial point.

6.2. Relating the strategies. We now focus on the primal correction Δx obtained from the least-squares, weighted least-squares, and Newton step strategies. In deciding how to choose Δx, we need to recover primal feasibility while ensuring that our choice of Δx does not unnecessarily compromise the positivity of x. The strategy of section 4 minimizes the norm $\|\Delta x\|$, while in section 5 our analysis of the Newton step strategy used the quantity $\|X^{-1}\Delta x\|_\infty$ (and its dual counterpart) to bound the size of the perturbations that could be corrected by this strategy. The weighted least-squares approach in which we aim to minimize $\|X^{-1}\Delta x\|$ explicitly is a natural alternative to both these strategies. We now discuss this strategy in the case in which the perturbation is confined to b, that is, $\Delta c = 0$ and $\Delta A = 0$.

Suppose we partition A as $[B \quad N]$, where B represents the "basic" columns for the original problem—the columns i such that $x_i^* > 0$ for some solution x^* of problem (P). Consider first the case in which Δb does not lie in the range of B. Since A is assumed to have full row rank, there exist v_B and $v_N \neq 0$ such that

$$Bv_B + Nv_N = \Delta b. \tag{6.5}$$

From (6.2a), we have

$$\Delta x = \Sigma^{-2} A^T w, \quad \text{where} \quad A\Sigma^{-2}A^T w = \Delta b = Bv_B + Nv_N.$$

It follows that

$$B\Sigma_B^{-2}B^T w + N\Sigma_N^{-2}N^T w = Bv_B + Nv_N,$$

where Σ_B and Σ_N are the appropriate partitions of Σ. Since Δb does not lie in the range space of B, we have $\|\Delta x_N\| = \|\Sigma_N^{-2}N^T w\| \geq \alpha$ for some $\alpha > 0$. If Σ is defined as in (6.3), we have from $\|x_N\| = O(\mu)$ that

$$\|X_N^{-1}\Delta x_N\| \geq \frac{\alpha}{\max(x_N)_i} \geq \frac{\bar{\alpha}}{\mu}$$

for some constant $\bar{\alpha} > 0$. A similar result applies when we choose the scaling as in (6.4), and thus the Newton step strategy will also yield a large value of $\|X_N^{-1}\Delta x_N\|$ under these circumstances. Clearly the unweighted least-squares strategy ($\Sigma = I$) also will yield a correction Δx_N with $\|\Delta x_N\| \geq \alpha$, so in this case too we may have to back off to a much earlier iterate of the interior-point method for the original problem before the correction strategy yields a feasible starting point. The point here

is that, even though we are minimizing $\|X^{-1}\Delta x\|$ explicitly in the weighted strategy, we cannot expect Δx_N to be appreciably smaller than in the other strategies when Δb does not lie in the range of B.

If Δb *does* lie in the range space of B, then there are two cases. First, consider the case in which B has full column rank. The analysis of Yıldırım and Todd [17] can be modified to show that the weighted least-squares correction using $\Sigma = X^{-1}$ converges to $(v^T, 0)^T$ as μ tends to 0, where v is the (unique) vector such that $Bv = \Delta b$. In [17], it is shown that the strategy of section 5 also converges to $(v^T, 0)^T$ as $\mu \downarrow 0$, so that the Newton step strategy gives asymptotically the same results as the weighted least-squares strategy in this case. Second, if B does not have full column rank, then it is shown in [17] that the Newton step strategy yields a correction Δx for which $\|X^{-1}\Delta x\|$ is well-behaved, in the sense that it remains bounded asymptotically as μ tends to 0. The analysis in [17] is based on a technical lemma (Lemma 5.1) which holds for the Newton step but does not necessarily hold for the weighted least-squares correction with $\Sigma = X^{-1}$. Hence, when Δb lies in the range space of B, it appears that the Newton step correction behaves at least as well as the weighted least-squares strategy that uses the scaling $\Sigma = X^{-1}$, at least asymptotically as $\mu \downarrow 0$.

Let us examine further the case in which B has full column rank and Δb is in the range space of B, but the perturbation is *not* necessarily small. In the notation of (6.5), we have that there exists a unique vector v_B such that

$$Bv_B = \Delta b. \tag{6.6}$$

If an interior-point method has been used to solve the original problem, and if we are close to the solution obtained with such a method, then the basic components (those contained in the subvector x_B) will be bounded away from zero, while the components of x_N will have size $O(\mu)$. By setting $v = (v_B^T, 0)^T$, we note that $Av = Bv_B = \Delta b$, so that v is feasible in both (4.2a) and (6.1a). In fact, we would expect v to be near-optimal in (6.1a) because it would yield an objective of size $O(\|\Delta b\|)$, and both weighting schemes (6.3) and (6.4) discourage solutions in which Δx_N is appreciably different from zero. As discussed above, we would also expect v to be near-optimal for the Newton step strategy. The plain least-squares strategy, on the other hand, does not discriminate between B and N components and may give a solution in which Δx_N is not especially small.

If it happens that $x+v \geq 0$, then $x+v$ will lie very close to a primal solution of the perturbed problem whenever x lies near a solution of the original problem. Hence, we would expect the weighted least-squares strategy to perform very well from an initial point that is an advanced iterate for the original problem, provided that the perturbation Δb can be accommodated without a change of basis. The plain least-squares method will usually perform less well, because it will be necessary to choose an initial point from the iterates for the original problem with an appreciably larger value of μ, to ensure that $x_N + \Delta x_N$ is nonnegative. In general, we would expect to need a μ that is bounded below by a multiple of $\|\Delta b\|$ in the plain least-squares strategy, whereas much smaller values of μ may be permissible in the other two strategies.

In the more interesting case in which the perturbation is large enough to force a change of basis, it is not at all clear that the weighted least-squares and Newton step strategies retain their advantage over plain least-squares. To be specific, if we do *not* have $x_B + \Delta x_B \geq 0$, it will be necessary to back up to an initial point in which the components of x_N are large enough to allow some of the perturbation to be absorbed by the Δx_N components

The need for backing up sufficiently far along the central path can also be motivated with reference to the dual problem (D) and to the geometry of the central path. When the perturbation is large enough to change the basis, the dual solution will usually change to a different vertex of its feasible polytope. Consequently, the central paths $\mathcal{P}$ (see (2.7)) for the original and perturbed problems (and therefore the neighborhoods $\mathcal{N}_2$ and $\bar{\mathcal{N}}_2$, and $\mathcal{N}_{-\infty}$ and $\bar{\mathcal{N}}_{-\infty}$) diverge significantly as $\mu \downarrow 0$. For large μ, however, the paths and neighborhood are quite similar for the original and perturbed problems. We need to choose μ sufficiently large that the neighborhoods are broad enough, and have a wide enough overlap, to ensure that the adjusted point $(x + \Delta x, y + \Delta y, s + \Delta s)$ lies inside the appropriate neighborhood for the perturbed problem.

We conclude that in the case of a perturbation to b, the strategies of sections 4 and 5 capture the potential advantages of using a weighted least-squares correction. Similar arguments can be made for the dual-only scaling, due to the symmetry between the primal and the dual problems.

6.3. Numerical results. We illustrate the remarks of the previous subsection—particularly the remarks about the relative performance of the strategies when the primal perturbation is and is not large enough to force a change of basis—with the following simple problem in R^2:

$$\min x_1 + x_2 \ \text{ s.t. } x_1 - x_2 = \epsilon, \ \ x \geq 0, \tag{6.7}$$

where $\epsilon > 0$ is a constant. We set $\epsilon = 10^{-2}$ throughout this section. This problem is well-conditioned (a large perturbation to the data is needed to make it infeasible) and has solution $x = (\epsilon, 0)^T$. Its dual is

$$\max \epsilon y \ \text{ s.t. } \begin{bmatrix} 1 \\ -1 \end{bmatrix} y + s = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \ \ s \geq 0.$$

Since $\epsilon > 0$, the dual solution is $s^* = (0, 2)^T$, $y^* = 1$. It is easy to show that the central path defined by (2.6) is as follows:

$$x(\mu) = \left(\frac{\mu + \epsilon}{2} + \frac{1}{2}\sqrt{\epsilon^2 + \mu^2}, \frac{\mu - \epsilon}{2} + \frac{1}{2}\sqrt{\epsilon^2 + \mu^2} \right)^T, \tag{6.8a}$$

$$s_1(\mu) = \frac{\mu}{x_1(\mu)}, \qquad s_2(\mu) = \frac{\mu}{x_2(\mu)}, \qquad y(\mu) = 1 - s_1(\mu). \tag{6.8b}$$

Note that for $\mu \gg \epsilon$ we have that

$$x(\mu) \approx (\mu, \mu)^T, \qquad s(\mu) = (1, 1)^T, \qquad y(\mu) \approx 0. \tag{6.9}$$

We plot $(x(\mu), y(\mu), s(\mu))$ for various values of μ in Table 6.1.

We consider perturbations to the right-hand side ϵ of the equality constraint in (6.7); namely, $\Delta b = \beta$. For $\beta > -\epsilon$, the solution of the perturbed primal can be attained without a change of basis. The solution of the perturbed primal becomes $x = (\epsilon + \beta, 0)^T$, while the solution of the dual remains unchanged. For $\beta < -\epsilon$, however, the solution of the perturbed primal becomes $x = (0, -(\beta + \epsilon))^T$, while the solution of the dual becomes $s^* = (2, 0)^T$, $y^* = 1$. Even for the latter case, we still have for $\mu \gg -(\beta + \epsilon)$ that the limits (6.9) hold, indicating that central paths for the original and perturbed problems are quite similar for larger values of μ.

TABLE 6.1
Central path points for (6.7)*, with* $\epsilon = 10^{-2}$.

μ	x	y	s
1.e-5	(1.0e-2,5.0e-6)	1.0e0	(1.0e-3, 2.0e0)
1.e-4	(1.0e-2, 5.0e-5)	9.9e-1	(1.0e-2, 2.0e0)
1.e-3	(1.1e-2, 5.2e-4)	9.0e-1	(9.5e-2, 1.9e0)
5.e-3	(1.3e-2, 3.1e-3)	6.2e-1	(3.8e-1, 1.6e0)
1.e-2	(1.7e-2, 7.1e-3)	4.1e-1	(5.9e-1, 1.4e0)
2.e-2	(2.6e-2, 1.6e-2)	2.4e-1	(7.6e-1, 1.2e0)
1.e-1	(1.1e-1, 9.5e-2)	5.0e-2	(9.5e-1, 1.0e0)
5.e-1	(5.1e-1, 5.0e-1)	1.0e-2	(9.9e-1, 1.0e0)

By solving (4.4), we find that the plain least-squares adjustment is

$$\Delta x_{LS} = (\beta/2, -\beta/2), \qquad \Delta y_{LS} = 0, \qquad \Delta s_{LS} = 0$$

(independently of x). By substituting (6.3) into (6.2), we obtain the following weighted least-squares adjustments:

$$\Delta x_{WLS} = \frac{\beta}{x_1^2 + x_2^2} \begin{bmatrix} x_1^2 \\ -x_2^2 \end{bmatrix}, \qquad \Delta y_{WLS} = 0, \qquad \Delta s_{WLS} = 0.$$

Finally, we obtain the Newton step adjustment by substituting (6.4) into (6.2):

$$\Delta x_{NS} = \frac{\beta}{(\frac{x_1}{s_1} + \frac{x_2}{s_2})} \begin{bmatrix} x_1/s_1 \\ -x_2/s_2 \end{bmatrix}, \quad \Delta s_{NS} = -X^{-1} S \Delta x_{NS}, \quad A^T \Delta y_{NS} + \Delta s_{NS} = 0.$$

The primal corrections Δx_{WLS} and Δx_{NS} coincide when (x, y, s) is on the central path, confirming our previous observation regarding the asymptotic coincidence. The dual correction is of course different for the two strategies.

In Tables 6.2, 6.3, and 6.4 we indicate the effects of the plain least-squares, weighted least-squares, and Newton step adjustments for perturbations $\beta = -0.1\epsilon$, $\beta = -\epsilon$, and $\beta = -10\epsilon$, respectively. We tabulate the following quantities against a selection of values of μ:

- The values of μ obtained after each of the adjustment strategies, that is, $\mu(x + \Delta x, s + \Delta s) = (x + \Delta x)^T (s + \Delta s)/2$, provided that all components of $(x + \Delta x, s + \Delta s)$ are positive. If not, we enter "-".
- The centrality indicators $(x + \Delta x)_i (s + \Delta s)_i / \mu(x + \Delta x, s + \Delta s)$, $i = 1, 2$. If any components of $(x + \Delta x, s + \Delta s)$ are nonpositive, we enter "-".

A good starting point for the perturbed problem is one for which the centrality indicators are not too far from 1 in all components (all greater than 10^{-1}, say), while the value of $\mu(x + \Delta x, s + \Delta s)$ is as small as possible.

In Table 6.2, the perturbation is small enough that a basis change is not needed and, as expected, the weighted least-squares and Newton adjustments perform well. Even when the central path point with $\mu = 10^{-5}$ is used as the basis for adjustment, well-centered starting points with small duality gaps are obtained from both strategies. The plain least-squares approach does not give particularly good adjusted points when applied at the central path points with $\mu = 10^{-5}$ or $\mu = 10^{-4}$, but becomes comparable for higher values of μ (that is, when the initial point is taken to be slightly further back along the central path).

In Table 6.3, where the perturbation is large enough to make the problem degenerate, the performances of the plain and weighted least-squares adjustment strategies

TABLE 6.2
Centrality of adjusted points for various μ, with $\epsilon = 10^{-2}$ and $\beta = -10^{-3}$.

μ	μ_{LS}	Centrality	μ_{WLS}	Centrality	μ_{NS}	Centrality
1.e-5	5.1e-4	(1.9e-2, 2.0e0)	9.5e-6	(9.5e-1, 1.1e0)	1.0e-5	(9.9e-1, 1.0e0)
1.e-4	6.0e-4	(1.6e-1, 1.8e0)	9.5e-5	(9.5e-1, 1.1e0)	1.0e-4	(1.0e0, 1.0e0)
1.e-3	1.5e-3	(6.6e-1, 1.3e0)	9.5e-4	(9.5e-1, 1.1e0)	1.0e-3	(1.0e0, 1.0e0)
5.e-3	5.3e-3	(9.1e-1, 1.1e0)	4.9e-3	(9.5e-1, 1.0e0)	5.0e-3	(1.0e0, 1.0e0)
1.e-2	1.0e-2	(9.5e-1, 1.0e0)	9.9e-3	(9.6e-1, 1.0e0)	1.0e-2	(1.0e0, 1.0e0)
2.e-2	2.0e-2	(9.8e-1, 1.0e0)	2.0e-2	(9.8e-1, 1.0e0)	2.0e-2	(1.0e0, 1.0e0)
1.e-1	1.0e-1	(1.0e0, 1.0e0)	1.0e-1	(1.0e0, 1.0e0)	1.0e-1	(1.0e0, 1.0e0)
5.e-1	5.0e-1	(1.0e0, 1.0e0)	5.0e-1	(1.0e0, 1.0e0)	5.0e-1	(1.0e0, 1.0e0)

TABLE 6.3
Centrality of adjusted points for various μ, with $\epsilon = 10^{-2}$ and $\beta = -10^{-2}$.

μ	μ_{LS}	Centrality	μ_{WLS}	Centrality	μ_{N}	Centrality
1.e-5	5.0e-3	(1.0e-3, 2.0e0)	5.0e-6	(1.0e-3, 2.0e0)	5.0e-6	(2.0e-3, 2.0e0)
1.e-4	5.1e-3	(1.0e-2, 2.0e0)	5.0e-5	(1.0e-2, 2.0e0)	5.0e-5	(2.0e-2, 2.0e0)
1.e-3	5.5e-3	(9.5e-2, 1.9e0)	5.5e-4	(9.5e-2, 1.9e0)	5.5e-4	(1.9e-1, 1.8e0)
5.e-3	8.1e-3	(3.8e-1, 1.6e0)	3.6e-3	(3.8e-1, 1.6e0)	3.6e-3	(6.6e-1, 1.3e0)
1.e-2	1.2e-2	(5.9e-1, 1.4e0)	8.5e-3	(5.9e-1, 1.4e0)	8.5e-3	(8.8e-1, 1.1e0)
2.e-2	2.1e-2	(7.6e-1, 1.2e0)	1.9e-2	(7.6e-1, 1.2e0)	1.9e-2	(9.8e-1, 1.0e0)
1.e-1	1.0e-1	(9.5e-1, 1.0e0)	1.0e-1	(9.5e-1, 1.0e0)	9.8e-2	(1.0e0, 1.0e0)
5.e-1	5.0e-1	(9.9e-1, 1.0e0)	5.0e-1	(9.9e-1, 1.0e0)	5.0e-1	(1.0e0, 1.0e0)

TABLE 6.4
Centrality of adjusted points for various μ, with $\epsilon = 10^{-2}$ and $\beta = -10^{-1}$.

μ	μ_{LS}	Centrality	μ_{WLS}	Centrality	μ_{N}	Centrality
1.e-5	-	-	-	-	-	-
1.e-4	-	-	-	-	-	-
1.e-3	-	-	-	-	-	-
5.e-3	-	-	-	-	-	-
1.e-2	-	-	-	-	-	-
2.e-2	-	-	-	-	-	-
1.e-1	1.0e-1	(5.1e-1, 1.5e0)	9.8e-2	(4.9e-1, 1.5e0)	7.5e-2	(9.7e-1, 1.0e0)
5.e-1	5.0e-1	(9.0e-1, 1.1e0)	5.0e-1	(9.0e-1, 1.1e0)	5.0e-1	(1.0e0, 1.0e0)

are similar. For both strategies, we need to adjust from a central path point with value around $\mu = 10^{-3}$ or $\mu = 5 \times 10^{-3}$ to obtain a well-centered starting point for the perturbed problem. The Newton step correction strategy yields adjusted points that are better centered, but again we need to use the central path point with $\mu = 10^{-3}$ to obtain a reasonably adjusted point.

In Table 6.4, the perturbation is large enough to force a change of basis, and we see that all approaches behave in a similar fashion. To obtain a starting point that is well centered, we need to choose a central path point from the original problem with a duality gap of $\mu = 10^{-1}$.

7. Conclusions. We have described two schemes by which the iterates of an interior-point method applied to an LP instance can be adjusted to obtain starting points for a perturbed instance. We have derived worst-case estimates for the number of iterations required to obtain convergence from these warm starting points. These

estimates depend chiefly on the size of the perturbation, on the conditioning of the original problem instance, and on a key property of the constraint matrix.

In future work, we plan to extend the techniques to infeasible interior-point methods and perform computational experiments to determine the practical usefulness of these techniques. We will also investigate extensions to wider classes of problems, such as convex quadratic programs and linear complementarity problems.

Acknowledgments. The first author would like to thank James Renegar for many helpful discussions and Mike Todd for pointing out the correct references to Dikin's result.

We are grateful to the referees for their careful reading of the first version of the paper and their perceptive comments, which prompted the addition of section 6.

REFERENCES

[1] S. A. Ashmanov, *Stability conditions for linear programming problems*, Comput. Math. Math. Phys., 21 (1981), pp. 40–49.
[2] I. I. Dikin, *On the speed of an iterative process*, Upravlyaemye Sistemi, 12 (1974), pp. 54–60.
[3] A. Forsgren, *On linear least-squares problems with diagonally dominant weight matrices*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 763–788.
[4] R. M. Freund, *A potential–function reduction algorithm for solving a linear program directly from an infeasible "warm start,"* Math. Programming, 52 (1991), pp. 441–466.
[5] J. Gondzio, *Warm start of the primal-dual method applied in the cutting-plane scheme*, Math. Programming, 83 (1998), pp. 125–143.
[6] J. Gondzio and J.-P. Vial, *Warm start and epsilon-subgradients in the cutting plane scheme for block-angular linear programs*, Comput. Optim. Appl., 14 (1999), pp. 17–36.
[7] J. E. Mitchell, *Karmarkar's Algorithm and Combinatorial Optimization Problems*, Ph.D. thesis, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, 1988.
[8] J. E. Mitchell and B. Borchers, *Solving real-world linear ordering problems using a primal–dual interior point cutting plane method*, Ann. Oper. Res., 62 (1996), pp. 253–276.
[9] J. E. Mitchell and M. J. Todd, *Solving combinatorial optimization problems using Karmarkar's algorithm*, Math. Programming, 56 (1992), pp. 245–284.
[10] M. A. Nunez and R. M. Freund, *Condition measures and properties of the central trajectory of a linear program*, Math. Programming, 83 (1998), pp. 1–28.
[11] J. Renegar, *Some perturbation theory for linear programming*, Math. Programming, 65 (1994), pp. 73–92.
[12] J. Renegar, *Incorporating condition measures into the complexity theory of linear programming*, SIAM J. Optim., 5 (1995), pp. 506–524.
[13] J. Renegar, *Linear programming, complexity theory, and elementary functional analysis*, Math. Programming, 70 (1995), pp. 279–351.
[14] J. Renegar, *Condition numbers, the barrier method, and the conjugate-gradient method*, SIAM J. Optim., 6 (1996), pp. 879–912.
[15] S. M. Robinson, *A characterization of stability in linear programming*, Oper. Res., 25 (1977), pp. 435–447.
[16] S. J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, 1997.
[17] E. A. Yildirim and M. J. Todd, *An interior-point approach to sensitivity analysis in degenerate linear programs*, SIAM J. Optim., to appear.
[18] E. A. Yildirim and M. J. Todd, *Sensitivity analysis in linear programming and semidefinite programming using interior-point methods*, Math. Program., 90 (2001), pp. 229–261.

SIAM J. OPTIM.
Vol. 12, No. 3, pp. 811–833

ON TRACTABLE APPROXIMATIONS OF UNCERTAIN LINEAR MATRIX INEQUALITIES AFFECTED BY INTERVAL UNCERTAINTY*

AHARON BEN-TAL† AND ARKADI NEMIROVSKI†

Abstract. We present efficiently verifiable sufficient conditions for the validity of specific NP-hard semi-infinite systems of linear matrix inequalities (LMIs) arising from LMIs with uncertain data and demonstrate that these conditions are "tight" up to an absolute constant factor. In particular, we prove that given an $n \times n$ interval matrix $\mathcal{U}_\rho = \{A \mid |A_{ij} - A^*_{ij}| \le \rho C_{ij}\}$, one can build a computable lower bound, accurate within the factor $\frac{\pi}{2}$, on the supremum of those ρ for which all instances of $\mathcal{U}_\rho$ share a common quadratic Lyapunov function. We then obtain a similar result for the problem of quadratic Lyapunov stability synthesis. Finally, we apply our techniques to the problem of maximizing a homogeneous polynomial of degree 3 over the unit cube.

Key words. robust semidefinite optimization, data uncertainty, Lyapunov stability synthesis, relaxations of combinatorial problems

AMS subject classifications. 90C05, 90C25, 90C30

PII. S1052623400374756

1. Introduction.

In this paper, we focus on the following "matrix cube" problem:

> MatrCube: Given an affine mapping $u \to \mathcal{B}(u) = B^0 + \sum_{\ell=1}^L u_\ell B^\ell$ from $\boldsymbol{R}^L$ to the space $\boldsymbol{S}^m$ of $m \times m$ real symmetric matrices and $\rho > 0$, check whether the image
>
> $$\mathcal{C}[\rho] = \{A \mid \exists (u, \|u\|_\infty \le \rho) : A = \mathcal{B}(u)\}$$
>
> of the box $\{\|u\|_\infty \le \rho\}$ under this mapping is contained in the cone $\boldsymbol{S}^m_+$ of positive semidefinite matrices.

Problem MatrCube is closely related to what is called *uncertain semidefinite programming with interval uncertainty*. Specifically, consider a linear matrix inequality (LMI)

$$A_0 + \sum_{j=1}^n x_j A_j \succeq 0; \tag{1}$$

here $x \in \boldsymbol{R}^n$ is the vector of variables, $A_0, \dots, A_n \in \boldsymbol{S}^m$, and $A \succeq B$ means that $A - B \in \boldsymbol{S}^m_+$. Assume that the *data* $[A_0, \dots, A_n]$ of the LMI "are uncertain"—we only know that the data belong to a given *uncertainty set* $\mathcal{U}$. Our aim is to find *robust* solutions of the resulting "uncertain LMI," i.e., solutions x of the semi-infinite system of LMIs

$$A_0 + \sum_{j=1}^n x_j A_j \succeq 0 \quad \forall [A_0, \dots, A_n] \in \mathcal{U}. \tag{2}$$

*Received by the editors July 31, 2000; accepted for publication (in revised form) August 22, 2001; published electronically February 27, 2002. This research was partially supported by Israeli Ministry of Science grant 0200-1-98 and Israel Science Foundation grant 683/99-10.0.

http://www.siam.org/journals/siopt/12-3/37475.html

†Faculty of Industrial Engineering and Management, Technion – Israel Institute of Technology, Technion City, Haifa 32000, Israel (abental@ie.technion.ac.il, nemirovs@ie.technion.ac.il). Part of this research was done while the first author was a Visiting Professor at Delft University of Technology, with support of the Dutch Research Organization NWO.

We say that the uncertainty is *interval* if $\mathcal{U}$ is the image of a box under an affine mapping:

$$(3)\quad \mathcal{U} = \mathcal{U}_\rho = \left\{ [A_0, \ldots, A_n] = [A_0^0, \ldots, A_n^0] + \sum_{\ell=1}^{L} u_\ell [A_0^\ell, \ldots, A_n^\ell] \mid \|u\|_\infty \le \rho \right\}.$$

As an example, consider the following *Lyapunov stability analysis* problem.

(LSA): Given an uncertain linear time-varying system

$$(4)\qquad \frac{d}{dt}x(t) = A(t)x(t), \quad A(t) \in \mathcal{A}\ \forall t,$$

where $\mathcal{A}$ is a given compact set of matrices, check whether the system admits a quadratic Lyapunov function, i.e., whether there exists a positive definite matrix X such that

$$\frac{d}{dt}(x^T(t)X^{-1}x(t)) < 0$$

for all nonzero trajectories $x(t)$ of (4).

Recall that the existence of a quadratic Lyapunov function is a standard *sufficient* condition for the stability of the system (i.e., for the fact that $x(t) \to 0, t \to \infty$, for every trajectory of the system, whatever is a (measurable) choice of $A(\cdot)$ taking values in $\mathcal{A}$). It is easily seen that the existence of a quadratic Lyapunov function for (4) is equivalent to the solvability of the semi-infinite system of LMIs

$$(5)\qquad \begin{array}{ll} \text{(a)} & X \succeq I, \\ \text{(b)} & AX + XA^T \preceq -I \quad \forall A \in \mathcal{A}, \end{array}$$

and every solution X of the latter system defines a quadratic Lyapunov function for (4). Note that (5) is of the form of (2), so that finding a quadratic Lyapunov stability certificate for uncertain linear dynamic system (4) is exactly the same as solving a semi-infinite system of LMIs (2) associated with an appropriately chosen uncertainty set $\mathcal{U}$. Note also that the latter set is an interval uncertainty, provided that $\mathcal{A}$ is also; e.g., provided that $\mathcal{A}$ is an interval matrix:

$$(6)\qquad \mathcal{A} = \mathcal{A}_\rho = \{A : |A_{ij} - A^*_{ij}| \le \rho D_{ij}\ \ \forall i, j\}.$$

(A^* is the "nominal" matrix, $D = [D_{ij} \ge 0]_{i,j}$ is a "perturbation scale," and $\rho > 0$ is a "perturbation level.")

The Lyapunov analysis example, as well as other examples which can be found in [1, 2, 4, 6], demonstrates the importance of robust solutions to semidefinite problems affected by data uncertainty, in particular by interval uncertainty. Theoretically speaking, the major difficulty with this concept is that (2) is a semi-infinite system of LMIs, and as such it can be computationally intractable. However, the set $\mathcal{X}$ of solutions to (2) is clearly closed and convex; it follows that, essentially (for details, see [8]), the "computational tractability" of (2) (i.e., the ability to find efficiently a point in $\mathcal{X}$ or to maximize efficiently a linear function over $\mathcal{X}$) is equivalent to the possibility of solving efficiently the following associated *analysis* problem.

Anal$[x]$: Given a candidate solution x, check whether it satisfies (2).

The role of the matrix cube problem in the context of uncertain semidefinite programming comes from the evident fact that *in the case of interval uncertainty* (3)*, problem*

Anal$[x]$ *is equivalent to problem* MatrCube *with the data*

$$B^\ell = A_0^\ell + \sum_{j=1}^n x_j A_j^\ell, \qquad \ell = 0, 1, \ldots, L.$$

Unfortunately, the matrix cube problem in general is NP-hard. This is so even in the case in which all "edges" $B^1, \ldots, B^L$ of the "matrix box" $\mathcal{C}[\rho]$ are of rank ≤ 2 (see [9] or section 4 below). Consequently, one is forced to look for *verifiable sufficient conditions* for the inclusion $\mathcal{C}[\rho] \subset \boldsymbol{S}_+^m$. The simplest condition of this type is evident:

(S) Assume that there exist matrices $X_1, \ldots, X_L$ satisfying the system of LMIs

$$\begin{array}{lrcl} \text{(a)} & X_\ell & \succeq & \pm\rho B^\ell, \qquad \ell = 1, \ldots, L, \\ \text{(b)} & \displaystyle\sum_{\ell=1}^L X_\ell & \preceq & B^0. \end{array} \tag{7}$$

Then $\mathcal{C}[\rho] \subset \boldsymbol{S}_+^m$.

In the context of semi-infinite system of LMIs (2) with interval uncertainty (3), condition (S) results in the following system of LMIs in variables $x, \{X_\ell\}$:

$$\begin{array}{rcl} X_\ell & \succeq & \pm\rho \left[A_0^\ell + \displaystyle\sum_{j=1}^n x_j A_j^\ell \right], \qquad \ell = 1, \ldots, L, \\ \displaystyle\sum_{\ell=1}^L X_\ell & \preceq & A_0^0 + \displaystyle\sum_{j=1}^n x_j A_j^0. \end{array} \tag{8}$$

This system is a "computationally tractable conservative approximation" of (2) in the sense that whenever x can be extended to a feasible solution of (8), x is feasible for (2) (by (S)).

The main result of this paper is as follows:

(N) The simple sufficient condition (S) for the inclusion $\mathcal{C}[\rho] \subset \boldsymbol{S}_+^m$ is not too conservative, provided that the edges $B^1, \ldots, B^L$ of the matrix box $\mathcal{C}[\rho]$ are of small ranks. Specifically, if (S) is not satisfied, then a $\vartheta(\mu)$-enlargement $\mathcal{C}[\vartheta(\mu)\rho]$ of the box $\mathcal{C}[\rho]$ is not contained in $\boldsymbol{S}_+^m$. Here

$$\mu = \max_{\ell=1,\ldots,L} \operatorname{rank}(B^\ell),$$

and $\vartheta(\mu)$ is a certain universal function such that

$$\vartheta(1) = 1; \quad \vartheta(2) = \tfrac{\pi}{2} = 1.57\ldots; \quad \vartheta(3) = 1.73\ldots; \quad \vartheta(4) = 2;$$
$$\vartheta(\mu) \le \tfrac{\pi\sqrt{\mu}}{2} \ \forall \mu.$$

Note that in typical semi-infinite systems of LMIs arising in control, perturbation of a single data entry results in small rank perturbations of the LMIs; whenever this is the case, (S) and (N) allow us to build a "tight" (up to a moderate absolute constant factor), computationally tractable conservative approximation of the semi-infinite system in question, provided that the uncertainty is interval. For example, in the Lyapunov stability analysis system (5), by perturbing a single entry in A, we

perturb the left-hand side of the semi-infinite LMI (5.b) by a matrix of rank ≤ 2; as we shall see, this observation combined with (N) allows us to build efficiently a lower bound, tight up to the factor $\frac{\pi}{2}$, for the "Lyapunov stability radius" of an interval matrix (i.e., for the supremum of those $\rho > 0$ for which all instances of the interval matrix (6) share a common quadratic Lyapunov function).

The rest of this paper is organized as follows. In section 2, we prove our main result (N). Section 3 is devoted to control applications of this result, specifically, those in Lyapunov stability analysis and synthesis. In section 4, we establish links between the matrix cube problem and the problem of maximizing a positive definite quadratic form over the unit cube; in particular, we demonstrate that (N) allows us to rederive the "$\frac{\pi}{2}$ theorem" of Nesterov [12], which states that the standard semidefinite bound on the maximum of a positive definite quadratic form over the unit cube is tight within the factor $\frac{\pi}{2}$. In the concluding section 5, we apply our techniques to the problem of maximizing a homogeneous polynomial of degree 3 over the unit cube.

In what follows, we frequently use the *semidefinite duality*; for the reader's convenience, we list here the relevant results (for proofs, see, e.g., [11]). Consider a semidefinite problem

$$\text{(Pr)} \qquad \min_x \left\{ c^T x : \sum_{j=1}^n x_j A_j - A_0 \succeq 0 \right\}; \tag{Pr}$$

here $x \in \boldsymbol{R}^n$, $A_0, \ldots, A_n \in \boldsymbol{S}^m$. It is assumed that no nontrivial linear combination of the matrices $A_1, \ldots, A_n$ is zero.

The *semidefinite dual* of (Pr) is the problem

$$\text{(Dl)} \qquad \max_X \{ \mathrm{Tr}(A_0 X) : \mathrm{Tr}(A_j X) = c_j,\ j = 1, \ldots, n, X \succeq 0 \}. \tag{Dl}$$

The duality is symmetric: (Dl) can be straightforwardly rewritten in the form of (Pr), and the semidefinite dual of this reformulation is (equivalent to) (Pr). The *semidefinite duality theorem* says that if (Pr) is bounded below and strictly feasible (i.e., $\sum_j \bar{x}_j A_j - A_0 \succ 0$, for certain $\bar{x}$, where $A \succ B$ means that $A - B$ is positive definite), then (Dl) is solvable and has the same optimal value as (Pr).

2. The matrix cube problem. The formal statement of our main result (N) is given by the following.

THEOREM 2.1. *Consider problem* MatrCube *along with system of LMIs* (7) *in matrix variables* $X_1, \ldots, X_L$, *and let*

$$\mu = \max_{1 \leq \ell \leq L} \mathrm{rank}(B^\ell)$$

(note $1 \leq \ell$ *in the* max*!). Then*

(i) *if system* (7) *is solvable, the matrix box* $\mathcal{C}[\rho]$ *is contained in the positive semidefinite cone* $\boldsymbol{S}^m_+$;

(ii.a) *if system* (7) *is unsolvable, the* $\vartheta(\mu)$*-enlargement* $\mathcal{C}[\vartheta(\mu)\rho]$ *of the matrix box* $\mathcal{C}[\rho]$ *is not contained in the positive semidefinite cone, where the function* $\vartheta(\cdot)$ *is given by*

$$\text{(9)} \qquad \frac{1}{\vartheta(k)} = \min_\alpha \left\{ \int_{\boldsymbol{R}^k} \left| \sum_{i=1}^k \alpha_i u_i^2 \right| (2\pi)^{-k/2} \exp\left\{ -\frac{u^T u}{2} \right\} du \;\middle|\; \sum_{i=1}^k |\alpha_i| = 1 \right\}; \tag{9}$$

(ii.b) *the function $\vartheta(\cdot)$ satisfies the relations*

$$\vartheta(k) \le \frac{\pi\sqrt{k}}{2} \quad \forall k; \qquad \vartheta(2) = \frac{\pi}{2}. \tag{10}$$

Proof. (i) is evident: If $\{X_\ell\}_{\ell=1}^L$ solves (7), then $u_\ell B^\ell \succeq -X_\ell$ for all ℓ and all $u_\ell \in [-\rho, \rho]$ by (7.a), so that

$$\|u\|_\infty \le \rho \Rightarrow B^0 + \sum_{\ell=1}^{\ell} u_\ell B^\ell \succeq B^0 - \sum_{\ell=1}^{L} X_\ell \succeq 0$$

(we have used (7.b)), and thus $\mathcal{C}[\rho] \subset \boldsymbol{S}_+^m$.

(ii.a): Assume that (7) is unsolvable, and let us prove that in this case $\mathcal{C}[\vartheta(\mu)\rho] \not\subset \boldsymbol{S}_+^m$.

1^0. Since (7) is unsolvable, the optimal value in the semidefinite program

$$\min_{t,\{X_\ell\}} \left\{ t \mid tI + B^0 \succeq \sum_{\ell=1}^{L} X_\ell,\ X_\ell \succeq \pm\rho B^\ell,\ \ell = 1, \dots, L \right\} \tag{P}$$

is positive. Since (P) is strictly feasible, it follows from the semidefinite duality theorem that the semidefinite dual of (P), i.e., the program

$$\max_{U,\{Y_\ell,Z_\ell\}} \left\{ \rho \sum_{\ell=1}^{L} \mathrm{Tr}([Y_\ell - Z_\ell]B^\ell) - \mathrm{Tr}(UB^0) \,\middle|\, \begin{array}{c} \mathrm{Tr}(U) = 1,\ U \succeq 0, \\ Y_\ell + Z_\ell = U,\ \ell = 1, \dots, L, \\ Y_\ell, Z_\ell \succeq 0,\ \ell = 1, \dots, L, \end{array} \right\} \tag{D}$$

is solvable with a positive optimal value.

2^0. To proceed, we need the following simple result.

LEMMA 2.2. *Let $U \succeq 0$ and B be a symmetric matrix of the same size as U. Then*

$$\max_{Y,Z\succeq 0: Y+Z=U} \mathrm{Tr}([Y-Z]B) = \max_{V=V^T, \|V\|\le 1} \mathrm{Tr}(VU^{1/2}BU^{1/2}) = \|\lambda(U^{1/2}BU^{1/2})\|_1, \tag{11}$$

where $\lambda(Z)$ is the vector of eigenvalues of a symmetric matrix Z (counted with their multiplicities) and $\|Z\| = \|\lambda(Z)\|_\infty$ is the operator norm of Z.

Proof. We clearly have

$$\begin{aligned} \max_{Y,Z\succeq 0: Y+Z=U} \mathrm{Tr}([Y-Z]B) &= \max_{P,Q\succeq 0: P+Q=I} \mathrm{Tr}([U^{1/2}PU^{1/2} - U^{1/2}QU^{1/2}]B) \\ &= \max_{P,Q\succeq 0: P+Q=I} \mathrm{Tr}([P-Q][U^{1/2}BU^{1/2}]) \\ &= \max_{V=V^T: \|V\|\le 1} \mathrm{Tr}(V[U^{1/2}BU^{1/2}]), \end{aligned}$$

as stated in the first equality in (11). To get the second equality, it suffices to consider the case in which the matrix $U^{1/2}BU^{1/2}$ is diagonal; in that case the equality becomes evident. □

In view of Lemma 2.2, the fact that (D) is solvable with positive optimal value means that there exists $U \succeq 0$ such that

$$\rho \sum_{\ell=1}^{L} \|\lambda(U^{1/2}B^\ell U^{1/2})\|_1 > \mathrm{Tr}(U^{1/2}B^0U^{1/2}). \tag{12}$$

We are about to provide a probabilistic interpretation of (12), and this interpretation will lead us to (ii.a).

3^0. Let us write $\xi \sim \mathcal{N}(0, I_k)$ to express that ξ is a random Gaussian k-dimensional vector with zero mean and unit covariance matrix, and let

$$p_k(u) = (2\pi)^{-k/2} \exp\{-u^T u/2\}$$

be the corresponding Gaussian density. We need the following fact.

LEMMA 2.3. *Whenever k is an integer and B is a symmetric $m \times m$ matrix with* $\mathrm{rank}(B) \le k$ *and* $\xi \sim \mathcal{N}(0, I_m)$, *one has*

$$\boldsymbol{E}\left\{|\xi^T B \xi|\right\} \ge \frac{\|\lambda(B)\|_1}{\vartheta(k)}.$$

Proof. It suffices to consider the case in which B is diagonal; in this case the relation in question immediately follows from the definition of $\vartheta(\cdot)$. □

4^0. Let $\xi \sim \mathcal{N}(0, I_m)$. We have

$$\begin{aligned}
\boldsymbol{E}\left\{\vartheta(\mu)\rho\sum_{\ell=1}^{L}|\xi^T U^{1/2} B^\ell U^{1/2}\xi|\right\} &= \rho\sum_{\ell=1}^{L}\vartheta(\mu)\boldsymbol{E}\left\{|\xi^T U^{1/2} B^\ell U^{1/2}\xi|\right\} \\
&\ge \rho\sum_{\ell=1}^{L}\|\lambda(U^{1/2} B^\ell U^{1/2})\|_1 \\
&\qquad \text{[by Lemma 2.3 and in view of } \mathrm{rank}(U^{1/2} B^\ell U^{1/2}) \le \mathrm{rank}(B^\ell) \le \mu\text{]} \\
&> \mathrm{Tr}(U^{1/2} B^0 U^{1/2}) \qquad \text{[by (12)]} \\
&= \boldsymbol{E}\left\{\xi^T U^{1/2} B^0 U^{1/2}\xi\right\} \qquad \text{[evident]},
\end{aligned}$$

so that there exists $r \in \boldsymbol{R}^m$ such that

$$\sum_{\ell=1}^{L}\vartheta(\mu)\rho|r^T U^{1/2} B^\ell U^{1/2} r| > r^T U^{1/2} B^0 U^{1/2} r.$$

Consequently, there exists a collection $\{\epsilon_\ell = \pm 1,\ \ell = 1, \dots, L\}$ such that

$$r^T\left[\sum_{\ell=1}^{L}\vartheta(\mu)\rho\epsilon_\ell U^{1/2} B^\ell U^{1/2}\right] r > r^T U^{1/2} B^0 U^{1/2} r,$$

i.e., the matrix $B^0 - \sum_{\ell=1}^{L}\vartheta(\mu)\rho\epsilon_\ell B^\ell$ is not positive semidefinite. Thus, $\mathcal{C}[\vartheta(\mu)\rho] \not\subset \boldsymbol{S}_+^m$, as claimed in (ii.a).

(ii.b): Let $\alpha \in \boldsymbol{R}^k$, $\|\alpha\|_1 = 1$, $\beta = \left[\begin{smallmatrix}\alpha\\ -\alpha\end{smallmatrix}\right] \in \boldsymbol{R}^{2k}$, and $\xi \sim \mathcal{N}(0, I_{2k})$. Setting

$$J = \int\left|\sum_{i=1}^{k} u_i^2\alpha_i\right| p_k(u)du,$$

we have

$$\boldsymbol{E}\left\{\left|\sum_{i=1}^{2k}\xi_i^2\beta_i\right|\right\} \le \boldsymbol{E}\left\{\left|\sum_{i=1}^{k}\xi_i^2\alpha_i\right| + \left|\sum_{i=k+1}^{2k}\xi_i^2\alpha_{i-k}\right|\right\} = 2J. \tag{13}$$

On the other hand, setting $\eta_i = (\xi_i - \xi_{i+k})/\sqrt{2}$, $\zeta_i = (\xi_i + \xi_{i+k})/\sqrt{2}$, we get

$$\left|\sum_{i=1}^{2k} \xi_i^2 \beta_i\right| = \left|\sum_{i=1}^{k} 2\alpha_i \eta_i \zeta_i\right| = 2\left|\widehat{\eta}^T \zeta\right|, \qquad \widehat{\eta} = \begin{bmatrix} \alpha_1\eta_1 \\ \vdots \\ \alpha_k\eta_k \end{bmatrix}, \qquad \zeta = \begin{bmatrix} \zeta_1 \\ \vdots \\ \zeta_k \end{bmatrix}. \tag{14}$$

Note that $\zeta \sim \mathcal{N}(0, I_k)$ and $\widehat{\eta}$, ζ are independent. Setting

$$\widetilde{\eta} = \begin{bmatrix} |\alpha_1\eta_1| \\ \vdots \\ |\alpha_k\eta_k| \end{bmatrix},$$

we have

$$\begin{aligned} \boldsymbol{E}\left\{|\widehat{\eta}^T\zeta|\right\} &= \boldsymbol{E}\left\{\|\widehat{\eta}\|_2\right\} \int |t| p_1(t)dt \\ &\qquad [\text{since } \widehat{\eta},\ \zeta \text{ are independent and } \zeta \sim \mathcal{N}(0, I_k)] \\ &= \boldsymbol{E}\left\{\|\widehat{\eta}\|_2\right\} \frac{2}{\sqrt{2\pi}} = \frac{2}{\sqrt{2\pi}} \boldsymbol{E}\left\{\|\widetilde{\eta}\|_2\right\} \\ &\geq \frac{2}{\sqrt{2\pi}} \|\boldsymbol{E}\left\{\widetilde{\eta}\right\}\|_2 = \frac{2}{\sqrt{2\pi}} \sqrt{\sum_{i=1}^{k} \alpha_i^2 \left(\frac{2}{\sqrt{2\pi}}\right)^2} \geq \frac{2}{\pi\sqrt{k}}. \end{aligned} \tag{15}$$

Combining (13), (14), and (15), we get $2J \geq \frac{4}{\pi\sqrt{k}}$, i.e., $\frac{1}{J} \leq \frac{\pi\sqrt{k}}{2}$, which yields the first relation in (10).

The second relation in (10) is given by the following computation:

$$\begin{aligned} \frac{1}{\vartheta(2)} &= \min_{\substack{\alpha \in \boldsymbol{R}^2, \\ \|\alpha\|_1 = 1}} \left\{ \int \left|\alpha_1 u_1^2 + \alpha_2 u_2^2\right| p_2(u)du \right\} = \min_{\theta \in [0,1]} \int \left|\theta u_1^2 - (1-\theta) u_2^2\right| p_2(u)du \\ &= \frac{1}{2} \int \left|u_1^2 - u_2^2\right| p_2(u)du \end{aligned}$$

[since the function to be minimized is convex in θ and symmetric w.r.t. $\theta = 1/2$]

$$= \left[\int |t| p_1(t)dt\right]^2 = \frac{2}{\pi}. \qquad \square$$

Let us reformulate Theorem 2.1 in a more convenient form as follows.

COROLLARY 2.4. *Consider a semi-infinite system of LMIs* (2) *with interval data (see* (3)*)*

$$A_0 + \sum_{j=1}^{n} x_j A_j \succeq 0 \quad \forall [A_0, A_1, \ldots, A_n] \in \mathcal{U}_\rho, \tag{Sys[ρ]}$$

$$\mathcal{U}_\rho = \left\{ [A_0, A_1, \ldots, A_n] = [A_0^0, A_1^0, \ldots, A_n^0] + \sum_{\ell=1}^{L} u_\ell [A_0^\ell, A_1^\ell, \ldots, A_n^\ell] \middle| \|u\|_\infty \leq \rho \right\},$$

and let

$$B^\ell[x] = A_0^\ell + \sum_{j=1}^n x_j A_j^\ell, \qquad \ell = 0, 1, \ldots, L,$$
$$\mu = \max_{\substack{x,\\ 1\le\ell\le L}} \operatorname{rank}(B^\ell[x])$$

(note $1 \le \ell$ *in the* max*!).*

The system of LMIs in variables $x, \{X_\ell\}$

$$X_\ell \succeq \pm\rho B^\ell[x], \qquad \ell = 1, \ldots, L,$$

(Appr[ρ])
$$\sum_{\ell=1}^L X_\ell \preceq B^0[x]$$

is a $\vartheta(\mu)$*-tight approximation of* (Sys[ρ])*, i.e.,*

(i) *if* x *can be extended to a feasible solution of* (Appr[ρ])*, then* x *is feasible for* (Sys[ρ])*;*

(ii) *if* x *cannot be extended to a feasible solution of* (Appr[ρ])*, then* x *is not feasible for* (Sys[$\vartheta(\mu)\rho$])*.*

2.1. Simplification of (7). From the computational viewpoint, a shortcoming of the sufficient condition (7) for the inclusion $\mathcal{C}[\rho] \subset \boldsymbol{S}_+^m$ is that the sizes of the LMI system (7), although polynomial in the sizes of MatrCube, are "large": The system has $2L+1$ "big" $(m \times m)$ LMIs and has $\frac{Lm(m+1)}{2}$ scalar decision variables. Our local goal is to demonstrate that in the case in which $\mu \equiv \max_{1\le\ell\le L} \operatorname{rank}(B^\ell)$ is small, (7) can be reduced to a much smaller system of LMIs.

PROPOSITION 2.5. (i) *Let* $S \in \boldsymbol{S}^m$ *be a matrix of rank* $k > 0$*, so that*

$$S = P^T R P$$

with invertible $k \times k$ *symmetric matrix* R *and* $k \times m$ *matrix* P *of rank* k*.*

(i.1) *A matrix* $X \in \boldsymbol{S}^m$ *satisfies the relation* $X \succeq \pm S$ *if and only if there exist* $k \times k$ *symmetric matrices* Y*,* Z *satisfying the relations*

(16)
$$\begin{aligned} &\text{(a)} \quad X \succeq \tfrac{1}{2} P^T (Y + Z) P, \\ &\text{(b)} \quad \begin{bmatrix} Y & R \\ R & Z \end{bmatrix} \succeq 0. \end{aligned}$$

(i.2) *In particular,* $X \succeq \pm S$ *if and only if there exists* $U \succeq \pm R$ *such that* $X \succeq P^T U P$*.*

(ii) *Consequently, the solvability of* (7) *is equivalent to the solvability of the system of LMIs*

(17)
$$\begin{aligned} &\text{(a)} \quad \begin{bmatrix} Y_\ell & \rho R_\ell \\ \rho R_\ell & Z_\ell \end{bmatrix} \succeq 0, \qquad \ell = 1, \ldots, L, \\ &\text{(b)} \quad \sum_{\ell=1}^L P_\ell^T (Y_\ell + Z_\ell) P_\ell \preceq 2B^0 \end{aligned}$$

in matrix variables $Y_\ell, Z_\ell \in \boldsymbol{S}^{k_\ell}$, $\ell = 1, \ldots, L$. *Here* $k_\ell = \mathrm{rank}(B^\ell)$ *(without loss of generality, we can assume that* $k_\ell > 0$*), and* $P_\ell, R_\ell = R_\ell^T$ *are* $k_\ell \times m$ *and* $k_\ell \times k_\ell$ *matrices of rank* k_ℓ *such that* $B^\ell = P_\ell^T R_\ell P_\ell$, $\ell = 1, \ldots, L$.

Proof. (i.1), "if" part: Assume that X, Y, Z satisfy (16); we should prove that then $X \succeq \pm S$. To this end it suffices to verify that if Y, Z satisfy (16.b), then $\frac{1}{2}(Y + Z) \succeq \pm R$, which is immediate:

$$\begin{aligned}
&(16.b) \Rightarrow \left\{ 0 \le \begin{bmatrix} \xi \\ \epsilon\xi \end{bmatrix}^T \begin{bmatrix} Y & R \\ R & Z \end{bmatrix} \begin{bmatrix} \xi \\ \epsilon\xi \end{bmatrix} \quad \forall(\xi \in \boldsymbol{R}^k, \epsilon = \pm 1) \right\} \\
&\Leftrightarrow \left\{ 0 \le \xi^T(Y + Z)\xi + 2\epsilon\xi^T R\xi \quad \forall(\xi \in \boldsymbol{R}^k, \epsilon = \pm 1) \right\} \Rightarrow \tfrac{1}{2}(Y + Z) \succeq \pm R.
\end{aligned}$$

(i.1), "only if" part: Let $X \succeq \pm S$. We should prove that there exist Y, Z satisfying (16). Assume, on the contrary, that the system of LMIs (16) in variables Y, Z is unsolvable, and consider the semidefinite program

$$t^* = \min_{t,Y,Z} \left\{ t : \begin{array}{c} tI + 2X - P^T(Y + Z)P \succeq 0, \\ \begin{bmatrix} Y & R \\ R & Z \end{bmatrix} \succeq 0 \end{array} \right\}. \tag{18}$$

Since P is of rank k, the intersections of the levels sets of the objective with the (nonempty!) feasible set of the problem are bounded, whence the problem is solvable; unsolvability of (16) implies that the optimal value t^* in the problem is positive. Since (18) clearly is strictly feasible, it follows that the semidefinite dual of (18), which is the semidefinite program

$$\min_{U,V,W,Q} \left\{ -2\,\mathrm{Tr}(UX) - 2\,\mathrm{Tr}(RQ^T) : \begin{array}{c} V = PUP^T \\ W = PUP^T \\ \begin{bmatrix} V & Q \\ Q^T & W \end{bmatrix} \succeq 0 \\ \mathrm{Tr}(U) = 1 \\ U, V, W \succeq 0 \end{array} \right\}, \tag{19}$$

is solvable with the same positive optimal value t^*. In other words, there exist $U \succeq 0$ and Q such that

$$\begin{array}{lrcl} \text{(a)} & \mathrm{Tr}(UX) & < & \mathrm{Tr}(RQ^T), \\ \text{(b)} & \begin{bmatrix} PUP^T & Q \\ Q^T & PUP^T \end{bmatrix} & \succeq & 0. \end{array} \tag{20}$$

From (20.b), by standard arguments, it follows that $Q = PU^{1/2}MU^{1/2}P^T$ for appropriately chosen M such that $M^TM \le I$. Consequently, (20.a) reads

$$\mathrm{Tr}(\underbrace{U^{1/2}XU^{1/2}}_{\bar{X}}) < \mathrm{Tr}(RP^TU^{1/2}M^TU^{1/2}P) = \mathrm{Tr}((\underbrace{U^{1/2}SU^{1/2}}_{\bar{S}})M^T).$$

Since $M^TM \le I$, the quantity $\mathrm{Tr}(\bar{S}M^T)$ is $\le \|\lambda(\bar{S})\|_1$, and we come to the relation $\mathrm{Tr}(\bar{X}) < \|\lambda(\bar{S})\|_1$. This is the desired contradiction, since from $X \succeq \pm S$ it follows that $\bar{X} \succeq \pm\bar{S}$, whence $\mathrm{Tr}(\bar{X}) \ge \|\lambda(\bar{S})\|_1$. (Notice what happens in the orthonormal basis, where $\bar{S}$ becomes diagonal.) Thus (i.1) is proved.

(i.2): If $X \succeq P^TUP$ with $U \succeq \pm R$, then of course $X \succeq \pm P^TRP = \pm S$. Conversely, if $X \succeq \pm S$, then by (i.2) there exist Y, Z satisfying (16). Setting $U = \frac{1}{2}(Y+Z)$, we have $X \succeq P^TUP$ by (16.a), and applying (i.1) to R rather than to S, we have $U \succeq \pm R$.

(ii) is an immediate consequence of (i). □

Note that when the ranks k_ℓ of the matrices B^ℓ, $\ell = 1, \dots, L$, are much less than the size m of these matrices, system (17) is much better suited for numerical processing than (7). Indeed, the latter system has $2L+1$ "big" $(m \times m)$ LMIs and totally $\frac{Lm(m+1)}{2}$ scalar decision variables, while the former system has a single "big" LMI, L "small" ones (of the sizes at most $2\mu \times 2\mu$, $\mu = \max_{1\le\ell\le L} k_\ell$), and no more than $L\mu(\mu+1)$ scalar decision variables. A shortcoming of the reformulated system, when compared with the original one, is that when the matrices B^ℓ depend affinely on certain vectors of parameters x (as is the case in the semi-infinite LMI (2) with interval uncertainty (3)), system (7) always is a system of LMIs in variables x, $\{X_\ell\}$ (cf. $(A[\rho])$), while (17) is a system of LMIs in x, Y_ℓ, Z_ℓ only under the additional (and restrictive) assumption that the matrices P_ℓ are independent of x. In section 3.2 we shall see that in certain important applications this shortcoming can be avoided.

3. Application I: Quadratic Lyapunov stability analysis and synthesis.

3.1. Lyapunov stability analysis/synthesis. Consider a controlled time-varying linear dynamic system

$$\begin{array}{lrcll} \text{(a)} & \frac{d}{dt}x(t) & = & A(t)x(t) + B(t)u(t) & \text{[open-loop system]}, \\ \text{(b)} & u(t) & = & Kx(t) & \text{[feedback]}, \\ & & & \Downarrow & \\ \text{(c)} & \frac{d}{dt}x(t) & = & [A(t) + B(t)K]x(t) & \text{[closed-loop system]} \end{array} \tag{21}$$

(x is n-dimensional, u is m-dimensional), which is uncertain in the sense that the dependency $t \mapsto [A(t), B(t)]$ is not known in advance; all we know is that

$$\forall t: \quad [A(t), B(t)] \in \mathcal{U}_\rho = \left\{[A, B] \mid |A_{ij} - B^*_{ij}| \le \rho C_{ij}, |B_{i\ell} - B^*_{i\ell}| \le \rho D_{i\ell} \quad \forall i, j, \ell\right\}. \tag{22}$$

Here A^*, B^* are given "nominal" data; C, D are given "scale matrices" with nonnegative entries; and $\rho \ge 0$ is the "perturbation level."

Consider the following pair of problems.

Lyapunov stability analysis: Given A^*, B^*, C, D, and a feedback K, find the supremum R^{a}_* of those $\rho \ge 0$ for which all instances $A + BK$, $[A, B] \in \mathcal{U}_\rho$, of the closed-loop system matrix (21.c) share a common quadratic Lyapunov function:

(LA)
$$R_*^{\rm a} = \sup_{\rho,X}\left\{\rho : X \succeq I, [A+BK]X + X[A+BK]^T \preceq -I \ \forall [A,B] \in \mathcal{U}_\rho\right\}$$
$$= \sup_{\rho,X}\left\{\rho : \begin{array}{l} X \succeq I \\ \forall (u_{ij}, |u_{ij}| \le \rho, u^{i\ell}, |u^{i\ell}| \le \rho): \\ \sum_{i,j} u_{ij} \underbrace{C_{ij}[E^{ij}X + XE^{ij}]}_{A_{ij}[X]} + \sum_{i,\ell} u^{i\ell} \underbrace{D_{i\ell}[F^{i\ell}KX + XK^T(F^{i\ell})^T]}_{A^{i\ell}[X]} \\ \preceq \underbrace{[-I - (A^* + B^*K)X - X(A^* + B^*K)^T]}_{A[X]} \end{array}\right\},$$

where E^{ij} are the basic $n \times n$ matrices (1 in cell ij, zeros in other cells), and $F^{i\ell}$ are the basic $n \times m$ matrices.

Lyapunov stability synthesis: Given A^*, B^*, C, D, find the supremum $R_*^{\rm s}$ of those $\rho \ge 0$ for which there exists a feedback K such that all instances $A+BK$, $[A,B] \in \mathcal{U}_\rho$, of the closed-loop system matrix (21.c) share a common quadratic Lyapunov function:

(LS)
$$R_*^{\rm s} = \sup_{\rho,X,K}\left\{\rho : X \succeq I, [A+BK]X + X[A+BK]^T \preceq -I \ \forall [A,B] \in \mathcal{U}_\rho\right\}$$
$$= \sup_{\rho,X,Z}\left\{\rho : X \succeq I, AX + XA^T + BZ + Z^TB^T \preceq -I \ \forall [A,B] \in \mathcal{U}_\rho\right\}, \ [Z = KX]$$
$$= \sup_{\rho,X,Z}\left\{\rho : \begin{array}{l} X \succeq I \\ \forall (u_{ij}, |u_{ij}| \le \rho, u^{i\ell}, |u^{i\ell}| \le \rho): \\ \sum_{i,j} u_{ij} \underbrace{C_{ij}[E^{ij}X + XE^{ij}]}_{B_{ij}[X]} + \sum_{i,\ell} u^{i\ell} \underbrace{D_{i\ell}[F^{i\ell}Z + Z^T(F^{i\ell})^T]}_{B^{i\ell}[Z]} \\ \preceq \underbrace{[-I - A^*X - X(A^*)^T - B^*Z - Z(B^*)^T]}_{B[X,Z]} \end{array}\right\},$$

where E^{ij} are the basic $n \times n$, and $F^{i\ell}$ are the basic $n \times m$ matrices.

As we can see, both problems (LA) and (LS) deal with *solvability of semi-infinite systems of LMIs*. Consider the approximations of these systems as follow:

(ALA)
$$\rho_*^{\rm a} = \max_{\rho,X,\{X^{ij},Y^{i\ell}\}}\left\{\rho : \begin{array}{c} A_{ij}[X] \le X^{ij}, -A_{ij}[X] \le X^{ij} \ \forall i,j, \\ A^{i\ell}[X] \le Y^{i\ell}, -A^{i\ell}[X] \le Y^{i\ell} \ \forall i,\ell, \\ \rho\left[\sum_{i,j} X^{ij} + \sum_{i,\ell} Y^{i\ell}\right] \preceq A[X], \end{array}\right\}$$

(ALS)
$$\rho_*^{\rm s} = \max_{\rho,X,Z,\{X^{ij},Y^{i\ell}\}}\left\{\rho : \begin{array}{c} B_{ij}[X] \le X^{ij}, -B_{ij}[X] \le X^{ij} \ \forall i,j, \\ B^{i\ell}[Z] \le Y^{i\ell}, -B^{i\ell}[Z] \le Y^{i\ell} \ \forall i,\ell, \\ \rho\left[\sum_{i,j} X^{ij} + \sum_{i,\ell} Y^{i\ell}\right] \preceq B[X,Z]. \end{array}\right\}$$

Note that both (ALA) and (ALS) are generalized eigenvalue problems (see [3, 10]) and as such are "computationally tractable."

Taking into account that the ranks of the matrices $A_{ij}[X]$, $A^{i\ell}[X]$, $B_{ij}[X]$, $B^{i\ell}[Z]$ never exceed 2 and applying Corollary 2.4, we come to the following result.

THEOREM 3.1. (i) *Consider the Lyapunov stability analysis problem and assume that the matrix $A^* + B^*K$ of the nominal closed-loop system is stable (i.e., all its eigenvalues are in the open left half-plane). Then problem* (ALA) *is an approximation of* (LA) *(i.e., the ρ, X-component of a feasible solution of* (ALA) *is a feasible solution of* (LA)*), and the optimal value of* (ALA) *coincides with the one of* (LA) *within the factor $\frac{\pi}{2}$:*

$$\rho_*^{\mathrm{a}} \le R_*^{\mathrm{a}} \le \frac{\pi}{2}\rho_*^{\mathrm{a}}.$$

(ii) *Consider the Lyapunov stability synthesis problem and assume that the nominal system is stabilizable (i.e., there exists a feedback K^* such that the matrix $A^* + B^*K^*$ is stable). Then problem* (ALS) *is an approximation of* (LS) *(i.e., the ρ, X, Z-component of a feasible solution of* (ALS) *is a feasible solution of* (LS)*), and the optimal value in* (ALS) *coincides with the one of* (LS) *within the factor $\frac{\pi}{2}$:*

$$\rho_*^{\mathrm{s}} \le R_*^{\mathrm{s}} \le \frac{\pi}{2}\rho_*^{\mathrm{s}}.$$

3.2. Simplifications of (ALA) and (ALS). Although the dimensions of the approximating semidefinite problems (ALA) and (ALS) are polynomial in the dimensions of the original system (21), they are nevertheless of huge design dimension. (They have a matrix variable per every uncertain entry in the data of (21).) This fact may render the approximating problems too difficult for practical use. We are about to demonstrate that the design dimensions of (ALA) and (ALS) can be reduced dramatically.

Consider a "generic problem" of the same structure as (ALA), (ALS):

We are given $\rho > 0$ and $L + 1$ symmetric $m \times m$ matrices $B^0[x]$, $B^1[x], \ldots, B^L[x]$ affinely depending on vector x of design variables, with $B^\ell[x]$, $\ell \ge 1$, of the form

$$B^\ell[x] = a_\ell b_\ell^T[x] + b_\ell[x]a_\ell^T, \tag{23}$$

where $a_\ell \ne 0$ and the vectors $b_\ell[x] \not\equiv 0$ are affine in x. We associate with these data the semi-infinite system of LMIs in variables x, u

$$B^0[x] + \sum_{\ell=1}^{L} u_\ell B^\ell[x] \succeq 0 \quad \forall (u : \|u\|_\infty \le \rho), \tag{24}$$

along with its "tractable conservative approximation"—the system of LMIs in variables x and additional matrix variables $X_1, \ldots, X_\ell$ as follows:

$$(\mathcal{P}[\rho]) \qquad \begin{array}{lrcl} \text{(a)} & X_\ell & \succeq & \pm\rho B^\ell[x], \qquad \ell = 1, \ldots, L, \\ \text{(b)} & \displaystyle\sum_{\ell=1}^{L} X_\ell & \preceq & B^0[x]. \end{array}$$

The problem is to simplify $(\mathcal{P}[\rho])$, i.e., to pass from this system to a system of LMIs in variables x and, perhaps, additional variables λ in

such a way that the new system (let it be called $(\mathcal{S}[\rho])$) is of smaller design dimension than $(\mathcal{P}[\rho])$ and is equivalent to $(\mathcal{P}[\rho])$ in the sense that an x can be extended to a feasible solution of $(\mathcal{S}[\rho])$ if and only if x can be extended to a feasible solution of $(\mathcal{P}[\rho])$.

Note that both (ALA) and (ALS) are of the form of $(\mathcal{P}[\rho])$.

The simplification of $(\mathcal{P}[\rho])$ to follow is similar to the construction presented in Proposition 2.5; it turns out that the specific form (23) of the dependence of B^ℓ on x allows us to end up with an analogy of (17) which is a system of LMIs in x and additional variables. The key to our simplification is the following simple fact (which can be viewed as certain strengthening of Proposition 2.5(i) for the case in which $S = ab^T + ba^T$).

LEMMA 3.2. *Let $a, b \in \boldsymbol{R}^m$ be two nonzero vectors, and let X be an $m \times m$ symmetric matrix. Then $X \succeq \pm[ab^T + ba^T]$ if and only if there exists a positive real λ such that*

$$X \succeq \lambda aa^T + \frac{1}{\lambda} bb^T. \tag{25}$$

Proof. "If" part: It suffices to prove that if $\lambda > 0$, then $\lambda aa^T + \frac{1}{\lambda} \succeq \pm[ab^T + ba^T]$, which is immediate:

$$\forall \xi : \xi^T \left[\lambda aa^T + \frac{1}{\lambda} bb^T\right] \xi = \lambda (a^T\xi)^2 + \frac{1}{\lambda}(b^T\xi)^2 \geq 2|a^T\xi||b^T\xi| \geq |\xi^T[ab^T + ba^T]\xi|.$$

"Only if" part: Assume that $a, b \neq 0$ and $X \succeq \pm[ab^T + ba^T]$; we should prove that there exists $\lambda > 0$ such that $X \succeq [\lambda aa^T + \frac{1}{\lambda} bb^T]$, or, a statement which is clearly equivalent, that the system of LMIs

$$\begin{aligned} X &\succeq \lambda aa^T + \mu bb^T, \\ \begin{bmatrix} \mu & 1 \\ 1 & \lambda \end{bmatrix} &\succeq 0 \end{aligned} \tag{26}$$

is solvable. Assume, on the contrary, that the system is unsolvable. Since $a, b \neq 0$, the semidefinite problem

$$\min_{t,\lambda,\mu} \left\{ t : \begin{array}{c} tI + X \succeq \lambda aa^T + \mu bb^T, \\ \begin{bmatrix} \mu & 1 \\ 1 & \lambda \end{bmatrix} \succeq 0 \end{array} \right\} \tag{27}$$

clearly is solvable; but then the infeasibility of (27) means that the optimal value in problem (27) is positive. Since the problem clearly is strictly feasible, the problem

$$\max_{U,p,q,r} \left\{ -\mathrm{Tr}(UX) - 2r : \begin{array}{rc} p &= b^T U b \\ q &= a^T U a \\ \mathrm{Tr}(U) &= 1 \\ \begin{bmatrix} p & r \\ r & q \end{bmatrix} &\succeq 0 \\ U &\succeq 0 \end{array} \right\}, \tag{28}$$

which is the semidefinite dual of (27), is solvable with positive optimal value. Since at a feasible solution to this problem one clearly has $|r| \leq \sqrt{pq} = \sqrt{(a^T U a)(b^T U b)}$,

the latter fact is equivalent to the existence of $U \succeq 0$ such that

$$2\sqrt{(a^TUa)(b^TUb)} > \mathrm{Tr}(UX). \tag{29}$$

Setting $\bar{a} = U^{1/2}a$, $\bar{b} = U^{1/2}b$, $\bar{X} = U^{1/2}XU^{1/2}$ and taking into account that $X \succeq \pm[ab^T + ba^T]$, we get

$$\begin{array}{lrcl} \text{(a)} & \bar{X} & \succeq & \pm Q, \qquad Q = \bar{a}\bar{b}^T + \bar{b}\bar{a}^T, \\ \text{(b)} & \mathrm{Tr}(\bar{X}) & < & 2\|\bar{a}\|_2\|\bar{b}\|_2. \end{array} \tag{30}$$

This is the desired contradiction. Indeed, from (30.a) it follows that $\mathrm{Tr}(\bar{X}) \geq \|\lambda(Q)\|_1$ (pass to the orthonormal basis where Q is diagonal); on the other hand, an immediate computation demonstrates that $\|\lambda(Q)\|_1 = 2\|\bar{a}\|_2\|\bar{b}\|_2$, which is $> \mathrm{Tr}(\bar{X})$ by (30.b). □

Lemma 3.2 underlies the following proposition.

PROPOSITION 3.3. *The LMI system* $(\mathcal{P}[\rho])$ *is equivalent to the following system of LMIs in variables* x *and additional variables* $Y \in \boldsymbol{S}^m$, $\lambda \in \boldsymbol{R}^L$:

$$\text{(a)} \quad \begin{bmatrix} Y - \sum_{\ell=1}^{L} \lambda_\ell a_\ell a_\ell^T & b_1[x] & b_2[x] & \dots & b_\ell[x] \\ b_1^T[x] & \lambda_1 & & & \\ b_2^T[x] & & \lambda_2 & & \\ \vdots & & & \ddots & \\ b_L^T[x] & & & & \lambda_L \end{bmatrix} \succeq 0, \tag{31}$$

$$\text{(b)} \quad \rho Y \preceq B^0[x].$$

Proof. We should prove that if x can be extended to a feasible solution of $(\mathcal{P}[\rho])$, then x can be extended to a feasible solution of (31), and vice versa.

1^0. Assume that $x, \{X_\ell\}$ is a feasible solution of $(\mathcal{P}[\rho])$, and let $J(x)$ be the set of those ℓ for which $b_\ell[x] = 0$. Let us extend x to a feasible solution of (31) as follows:

1. For $\ell \in J(x)$, we set $\lambda_\ell = 0$.
2. For $\ell \notin J(x)$, we have $a_\ell \neq 0$, $b_\ell[x] \neq 0$, and $\rho^{-1}X_\ell \succeq \pm[a_\ell b_\ell^T[x] + b_\ell[x]a_\ell^T]$. Applying Lemma 3.2, we can find $\lambda_\ell > 0$ such that $\rho^{-1}X_\ell \succeq \lambda_\ell a_\ell a_\ell^T + \lambda_\ell^{-1} b_\ell[x] b_\ell^T[x]$.
3. After we have defined $\lambda_\ell \geq 0$ for all $\ell = 1, \dots, L$, we set

$$Y = \sum_{\ell \notin J(x)} \left[\lambda_\ell a_\ell a_\ell^T + \lambda_\ell^{-1} b_\ell[x] b_\ell^T[x]\right].$$

Let us prove that $x, Y, \{\lambda_\ell\}$ is feasible for (31). Indeed, (31.a) is readily given by the definition of Y and the Schur complement lemma. (Note that a zero λ_ℓ on the diagonal of the left-hand side matrix in (31.a) corresponds to a zero row and a zero column.) Further, from the origin of λ_ℓ, $\ell \notin J(x)$, it follows that $Y \preceq \sum_{\ell \notin J(x)} \rho^{-1}X_\ell \preceq \rho^{-1}\sum_{\ell=1}^{L} X_\ell$, and since $x, \{X_\ell\}$ is feasible for $(\mathcal{P}[\rho])$, we conclude that $\rho Y \preceq B^0[x]$, i.e., (31.b) is valid. Thus, $x, Y, \{\lambda_\ell\}$ is feasible for (31).

2^0. Now assume that $x, Y, \{\lambda_\ell\}$ is feasible for (31), and let us prove that x can be extended to a feasible solution of $(\mathcal{P}[\rho])$. Let, as above, $J(x)$ be the set of those ℓ for

which $b_\ell[x] = 0$. Note that from (31.a) it follows that $\lambda_\ell \geq 0$ for all ℓ, and $\lambda_\ell > 0$ for $\ell \notin J(x)$. Let us set

$$X_\ell = \rho \begin{cases} 0, & \ell \in J(x), \\ \lambda_\ell a_\ell a_\ell^T + \lambda_\ell^{-1} b_\ell[x] b_\ell^T[x], & \ell \notin J(x). \end{cases}$$

Applying Lemma 3.2, we see that ($\mathcal{P}[\rho]$.a) holds true. Now, by the Schur complement lemma from (31.a), it follows that

$$\rho^{-1} \sum_\ell X_\ell = \sum_{\ell \notin J(x)} \left[\lambda_\ell a_\ell a_\ell^T + \lambda_\ell^{-1} b_\ell[x] b_\ell^T[x]\right] \preceq Y;$$

this observation combined with (31.b) implies the validity of ($\mathcal{P}[\rho]$.b). Thus, $x, \{X_\ell\}$ is feasible for ($\mathcal{P}[\rho]$). □

We have reduced the system of LMIs ($\mathcal{P}[\rho]$) to (31). In the original system, there are $(\dim x + L \dim X)$ scalar design variables, while in the resulting system there are just $(\dim x + \dim X + L)$ design variables. To realize how large the reduction in the design dimension can be, consider the case in which ($\mathcal{P}[\rho]$) is the problem (ALS). Here $x = X$ is a symmetric $n \times n$ matrix, and L is the total number of uncertain entries in the underlying uncertain interval matrix $[A, B]$. Here the original system ($\mathcal{P}[\rho]$) has $L + 1$ symmetric $n \times n$ matrix variables, i.e., totally $\frac{(L+1)n(n+1)}{2}$ scalar design variables, and $(2L + 1)$ "large" $(n \times n)$ LMIs. The reformulated system (31) has just two symmetric $n \times n$ matrix variables X, Y, and $L \leq n^2 + nm$ scalar variables λ_ℓ, i.e., totally $L + n(n + 1) \leq 2n^2 + n(m + 1)$ scalar design variables. As for LMIs, system (31) has one "large" $(n \times n)$ LMI (b) and one "very large" $((n+L) \times (n+L))$ LMI (a); note, however, that this LMI is of very simple "arrow" structure and is very sparse. Thus, (31) seems to be much better suited for numerical processing than ($\mathcal{P}[\rho]$).

3.3. Extensions. *An LMI region* is a set $\mathcal{H}$ in the complex plane $\boldsymbol{C}$ representable as

$$\mathcal{H} = \{z \in \boldsymbol{C} \mid f_\mathcal{H}(z) \equiv P + Qz + Q^T \bar{z} \prec 0\},$$

where $P = P^T$ and Q are real $k \times k$ matrices and $\bar{z}$ is the complex conjugate of z. The simplest examples of LMI regions are

1. *open left half-plane*: $f_\mathcal{H}(z) = z + \bar{z}$;
2. *open disk* $\{z \mid |z + q| \leq r\}$, $q \in \boldsymbol{R}, r > 0$: $f_\mathcal{H}(z) = \begin{pmatrix} -r & \bar{z}+q \\ z+q & -r \end{pmatrix}$;
3. *the interior of the sector* $\{z \mid \pi - \theta \leq |\arg(z)| \leq \pi\}$ $(-\pi < \arg(z) \leq \pi$, $0 < \theta < \frac{\pi}{2})$:

$$f_\mathcal{H}(z) = \begin{pmatrix} (z + \bar{z}) \sin\theta & -(z - \bar{z}) \cos\theta \\ (z - \bar{z}) \cos\theta & (z + \bar{z}) \sin\theta \end{pmatrix};$$

4. *the stripe* $\{z \mid h_1 < \Re(z) < h_2\}$: $f_\mathcal{H}(z) = \begin{pmatrix} 2h_1 - (z + \bar{z}) & 0 \\ 0 & (z + \bar{z}) - 2h_2 \end{pmatrix}$.

It is known (see, e.g., [5]) that the spectrum $\Sigma(A)$ of a real $n \times n$ matrix A belongs to $\mathcal{H}$ if and only if there exists $Y \in \boldsymbol{S}^m$, $Y \succ 0$, such that the $k \times k$ block matrix $\mathcal{M}[X, A]$ with the $m \times m$ blocks

$$\mathcal{M}_{ij}[X, A] = P_{ij} X + Q_{ij} A X + Q_{ji} X A^T, \qquad i, j = 1, \ldots, m,$$

is negative definite. We can treat such an X as a certificate of the inclusion $\Sigma(A) \subset \mathcal{H}$, and for homogeneity reasons we can normalize this certificate to satisfy the relations $X \succeq I$, $\mathcal{M}[X, A] \preceq -I$. From now on, we speak about normalized certificates only.

The problem we are interested in now is as follows. Given an LMI region $\mathcal{H}$ and an "uncertain interval matrix"

$$\begin{aligned}\mathcal{U}_\rho = \quad & \{[A, B] \in \boldsymbol{R}^{n\times n} \times \boldsymbol{R}^{n\times m} \mid \\ & |A_{ij} - A^*_{ij}| \le \rho C_{ij},\ |B_{i\ell} - B^*_{i\ell}| \le \rho D_{i\ell},\ 1 \le i, j \le n, 1 \le \ell \le m\}\end{aligned}$$

of the open-loop system (21), we ask what is the supremum R_* of those $\rho \ge 0$ for which there exists a linear feedback $K \in \boldsymbol{R}^{m\times n}$ such that all instances $A + BK$, $[A, B] \in \mathcal{U}_\rho$, of the uncertain matrix of the closed-loop system share a common certificate X of the inclusion $\Sigma(A + BK) \subset \mathcal{H}$. This important problem in control is a natural extension of the Lyapunov stability synthesis problem. The problem can be treated in the same fashion as Lyapunov analysis/synthesis. Indeed, $X \succeq I$ certifies the inclusion $\Sigma(A + BK) \subset \mathcal{H}$ for all $[A, B] \in \mathcal{U}_\rho$ if and only if (X, K) solves the semi-infinite system of matrix inequalities

$$\mathcal{M}[X, A + BK] \preceq -I \quad \forall [A, B] \in \mathcal{U}_\rho.$$

Passing from the variables X, K to $X, Z = KX$, we convert this system to the semi-infinite system of LMIs

$$\begin{aligned}&\forall\,([A, B] \in \mathcal{U}_\rho):\\ &\mathcal{N}(X, Z, A, B) \equiv \left[P_{ij}X + Q_{ij}AX + Q_{ij}BZ + Q_{ji}XA^T + Q_{ji}Z^TB^T\right]_{1\le i,j\le k} \preceq -I,\end{aligned} \tag{32}$$

where $[M_{ij}]_{1\le i,j\le k}$ denotes block matrix with blocks M_{ij}. We see that (X, Z) solves (32) if and only if (X, Z) solves the semi-infinite system of LMIs

$(\mathcal{I}[\rho])$

$$\begin{aligned}&\forall\,(\{u_{ij}, |u_{ij}| \le 1\}, \{v_{i\ell}, |v_{i\ell}| \le 1\}):\\ &\rho\left(\sum_{(i,j):C_{ij}>0} u_{ij}\mathcal{N}^0(X, Z, C_{ij}E^{ij}, 0) + \sum_{(i,\ell):D_{i\ell}>0} v_{i\ell}\mathcal{N}^0(X, Z, 0, D_{i\ell}F^{i\ell})\right)\\ &\qquad -I - \mathcal{N}(X, Z, A^*, B^*) \quad \succeq \quad 0,\\ &\mathcal{N}^0(X, Z, A, B) = \left[Q_{ij}AX + Q_{ij}BZ + Q_{ji}XA^T + Q_{ji}Z^TB^T\right]_{1\le i,j\le k},\end{aligned}$$

where the basic matrices E^{ij}, $F^{i\ell}$ are the same as in (LA), (LS). As before, an evident sufficient condition for $X \succeq I$ and Z to solve $(\mathcal{I}[\rho])$ is the existence of matrices X^{ij}, $(i, j) \in \mathcal{C} = \{(i, j) \mid C_{ij} > 0\}$, and $Z^{i\ell}$, $(i, \ell) \in \mathcal{D} = \{(i, \ell) \mid D_{i\ell} > 0\}$, such that $(X, Z, X^{ij}, Z^{i\ell})$ solves the system of LMIs

$$\begin{aligned}&X^{ij} \succeq \mathcal{N}^0(X, Z, C_{ij}E^{ij}, 0), \quad X^{ij} \succeq -\mathcal{N}^0(X, Z, C_{ij}E^{ij}, 0), \qquad (i, j) \in \mathcal{C},\\ &Z^{i\ell} \succeq \mathcal{N}^0(X, Z, 0, D_{i\ell}F^{i\ell}), \quad Z^{i\ell} \succeq -\mathcal{N}^0(X, Z, 0, D_{i\ell}F^{i\ell}), \qquad (i, \ell) \in \mathcal{D},\end{aligned}$$

$(\mathcal{II}[\rho])$

$$\begin{aligned}&\rho\left(\sum_{(i,j)\in\mathcal{C}} X^{ij} + \sum_{(i,\ell)\in\mathcal{D}} Z^{i\ell}\right) \preceq -I - \mathcal{N}(X, Z, A^*, B^*),\\ &X \succeq I.\end{aligned}$$

Invoking Theorem 2.1, we arrive at the following result.

THEOREM 3.4. *Let the system* $(\mathcal{I}[0])$ *(or, which is the same,* $(\mathcal{II}[0])$*) be solvable, and let*

$$\mu = \max\left[\max_{X,Z,i,j} \operatorname{rank}(\mathcal{N}^0(X,Z,C_{ij}E^{ij},0)),\ \max_{X,Z,i,\ell} \operatorname{rank}(\mathcal{N}^0(X,Z,0,D_{i\ell}F^{i\ell}))\right].$$

Then

(i) *if* $(\mathcal{II}[\rho])$ *is solvable, then so is* $(\mathcal{I}[\rho])$*, and the* (X,Z)*-component of a solution of the former system solves the latter system;*

(ii) *if* $(\mathcal{II}[\rho])$ *is unsolvable, then so is* $(\mathcal{I}[\vartheta(\mu)\rho])$*, where* $\vartheta(\cdot)$ *is the function given in* (9)*.*

In particular,

$$\frac{\sup\{\rho:\ (\mathcal{I}[\rho]) \text{ is solvable}\}}{\sup\{\rho:\ (\mathcal{II}[\rho]) \text{ is solvable}\}} \le \vartheta(\mu). \tag{33}$$

Note that the denominator in (33) is the optimal value in an explicit generalized eigenvalue problem and thus is efficiently computable. Note also that one always has $\mu \le 2k$, and that, for our list of the 4 simple LMI regions, $\mu = 2$ in cases 1 and 2 ("half-plane" and "disk"), and $\mu = 4$ in cases 3 and 4 ("sector" and "stripe").

There are many other applications of Theorem 2.1 to semi-infinite systems of LMIs (2) arising in control, provided that the uncertainty set $\mathcal{U}$ in (2) is an interval uncertainty. In a typical control application, all matrices A_j in (2) share a common block-diagonal structure and are such that when perturbing a single data entry, every diagonal block in the matrix $A_0+\sum_j x_j A_j$ is perturbed by a small rank matrix, which is exactly the case considered in Theorem 2.1.

4. Application II: Quadratic maximization over the unit cube. Here we demonstrate that the MatrCube problem in its simplest form, where all the edge matrices B^ℓ are very specific matrices of ranks ≤ 2, is equivalent to the problem

$$\omega_*(Q) = \max_x \left\{x^TQx : \|x\|_\infty \le 1\right\} \quad [Q \succ 0] \tag{34}$$

of maximizing a positive definite quadratic form over the unit cube. On one hand, this observation says that MatrCube (already in the case of "rank 2 edges") is NP-hard (since (34) is). On the other hand, our observation allows us to extract from Theorem 2.1 a certain statement about the possibility of building efficiently a tight bound on the optimal value in (34). As it turns out, this bound is exactly the one given by the standard semidefinite relaxation of (34), and the corresponding "tightness" statement coming from Theorem 2.1 is nothing but the "$\frac{\pi}{2}$ Theorem" of Nesterov [12].

The link between the quadratic maximization over the unit cube and the matrix cube problem is given by the following simple observation.

PROPOSITION 4.1. *Assume that* Q *in* (34) *is positive definite. Then*

$$\begin{array}{ll} \text{(a)} & \omega \ge \omega_*(Q) \equiv \max\limits_{x:\|x\|_\infty\le 1} x^TQx, \\ \text{(i)} & \Updownarrow \\ \text{(b)} & \omega\xi^TQ^{-1}\xi \ge \|\xi\|_1^2 \quad \forall \xi, \\ \text{(ii)} & \Updownarrow \\ \text{(c)} & \omega Q^{-1} + \{A \in \boldsymbol{S}^m : |A_{ij}| \le 1,\ 1 \le i,j \le m\} \subset \boldsymbol{S}^m_+. \end{array} \tag{35}$$

Proof. Relation (a) means that the ellipsoid $\{x : x^TQx \le \omega\}$ contains the unit cube $\{x : \|x\|_\infty \le 1\}$. Passing to polars, this is exactly the same as saying that the polar of the ellipsoid, which is the ellipsoid $\{\xi : \xi^TQ^{-1}\xi \le \omega^{-1}\}$, is contained in the polar of the unit cube, which is the set $\{\xi : \|\xi\|_1 \le 1\}$. But the latter inclusion is exactly what is stated in (b). Thus we have proved the equivalence (i).

Now, (c) says exactly that

$$\omega\xi^TQ^{-1}\xi + \min_A \left\{\xi^TA\xi : A = A^T, |A_{ij}| \le 1\right\} \ge 0 \quad \forall\xi. \tag{36}$$

The minimum in the left-hand side of this relation is equal to $-\|\xi\|_1^2$. (Indeed, $\xi^TA\xi \ge -\|\xi\|_1^2$ whenever $|A_{ij}| \le 1$ for all i, j, and $\xi^TA\xi = -\|\xi\|_1^2$ for $A_{ij} = -\mathrm{sign}(\xi_i)\mathrm{sign}(\xi_j)$, $i, j = 1, \dots, m$.) Thus, (36) is equivalent to the relation $\omega\xi^TQ^{-1}\xi - \|\xi\|_1^2 \ge 0$ for all ξ, which is nothing but (b). Thus we have proved the equivalence (ii). □

Now, let S^{ij} be the basic symmetric matrices (so that S^{ii} has a single nonzero entry, equal to 1, in the cell (i, i), and S^{ij}, $i \ne j$, has exactly two nonzero entries, both equal to 1, in the cells (i, j) and (j, i)). Relation (35.b) says exactly that the matrix box

$$\mathcal{C}\left[\frac{1}{\omega}\right] = \left\{Q^{-1} + \sum_{1\le i\le j\le m} u_{ij}S^{ij} : \|u\|_\infty \le \frac{1}{\omega}\right\}$$

is contained in the positive semidefinite cone. According to Theorem 2.1, a *sufficient* condition for this inclusion is the solvability of the system of LMIs as follows:

$$\begin{aligned} X^{ij} &\succeq \pm\rho S^{ij}, \quad 1 \le i \le j \le m, \\ \sum_{1\le i\le j\le m} X^{ij} &\preceq Q^{-1}, \qquad \rho = \frac{1}{\omega}. \end{aligned} \tag{37}$$

Moreover, since the ranks of the edge matrices S^{ij} are ≤ 2, Theorem 2.1 says that the solvability of (37) is a "tight, within the factor $\frac{\pi}{2}$" sufficient condition for the validity of (35.b). Taking into account that the smallest value of ω for which (35.b) is valid is exactly $\omega_*(Q)$ (Proposition 4.1), we arrive at the following.

PROPOSITION 4.2. *Let $Q \succ 0$. Consider the semidefinite program*

$$\rho(Q) = \max_{\rho, X^{ij}} \left\{\rho : \begin{array}{rcl} X^{ij} &\succeq& \pm\rho S^{ij},\ 1 \le i \le j \le m \\ \sum\limits_{1\le i\le j\le m} X^{ij} &\preceq& Q^{-1} \end{array}\right\}. \tag{38}$$

The reciprocal of the optimal value in this problem is an upper bound on the optimal value $\omega_*(Q)$ in the problem of quadratic maximization (34)*, and this bound is tight within the factor $\frac{\pi}{2}$:*

$$\omega_*(Q) \le \frac{1}{\rho(Q)} \le \frac{\pi}{2}\omega_*(Q). \tag{39}$$

Proposition 4.2 says that a certain quantity which is efficiently computable via semidefinite programming (namely, $1/\rho(Q)$) is a tight, within the factor $\pi/2$, upper

bound on the maximum $\omega_*(Q)$ of the positive definite quadratic form x^TQx over the unit cube. We are about to demonstrate that our bound is nothing but the standard semidefinite upper bound

$$\begin{array}{rcl}\omega^*(Q) & = & \max_X \{\mathrm{Tr}(QX) : X_{ii} \le 1,\, i = 1, \ldots, m, X \succeq 0\} \\ & = & \min\limits_{\lambda} \left\{ \sum\limits_{i=1}^m \lambda_i : \mathrm{Diag}\{\lambda\} \succeq Q \right\}\end{array} \tag{40}$$

on $\omega_*(Q)$.

PROPOSITION 4.3. *For $Q \succ 0$, one has $\frac{1}{\rho(Q)} = \omega^*(Q)$.*

Proof. Let e_i be the standard basic *orths* in $\boldsymbol{R}^m$, so that $S^{ij} = \frac{1}{1+\delta_{ij}}[e_ie_j^T + e_je_i^T]$, where δ_{ij} are the Kronecker symbols. Applying Lemma 3.2, we see that

$$\begin{array}{l}\rho(Q) = \max \left\{ \rho : \exists X^{ij} : \begin{array}{c} X^{ij} \succeq \pm \frac{\rho}{1+\delta_{ij}}[e_ie_j^T + e_je_i^T],\, 1 \le i \le j \le m \\ \sum\limits_{1 \le i \le j \le m} X^{ij} \preceq Q^{-1} \end{array} \right\} \\ = \max \left\{ \rho : \exists \{H_{ij} > 0\}_{1 \le i \le j \le m} : \sum\limits_{1 \le i \le j \le m} \frac{1}{1+\delta_{ij}} \left[H_{ij} e_i e_i^T + H_{ij}^{-1} e_j e_j^T \right] \preceq \frac{1}{\rho} Q^{-1} \right\}.\end{array} \tag{41}$$

Let $\mathcal{H}$ be the set of all $m \times m$ matrices $H = [H_{ij}]$ with positive entries such that $H_{ij}H_{ji} \ge 1$ for all i, j. It is immediately seen that (41) can be rewritten as

$$\begin{array}{rcl}\rho^{-1}(Q) & = & \min \left\{ \omega : \exists (H \in \mathcal{H}) : \Lambda(H) \preceq \omega Q^{-1} \right\} \\ & = & \min \left\{ \omega : \exists (H \in \mathcal{H}) : Q \preceq \omega \Lambda^{-1}(H) \right\},\end{array} \tag{42}$$

where $\Lambda(H)$ is the diagonal matrix with the diagonal entries

$$\Lambda_{ii}(H) = \sum_{j=1}^m H_{ij}, \qquad i = 1, \ldots, m.$$

LEMMA 4.4. *The matrices which can be represented as $\Lambda^{-1}(H)$, $H \in \mathcal{H}$, are exactly the positive definite diagonal matrices with trace ≤ 1.*

Proof. A matrix $M = \mathrm{Diag}\{\mu_i\}$ with $\mu_i > 0$ and $s \equiv \sum_i \mu_i \le 1$ is $\Lambda^{-1}(H)$ for H given by $H_{ij} = \frac{\mu_j}{s\mu_i}$; note that $H \in \mathcal{H}$ due to $s \le 1$. It remains to prove that if $H \in \mathcal{H}$, then $\mathrm{Tr}(\Lambda^{-1}(H)) \le 1$. To this end observe that

(*) For positive reals $\mu_1, \ldots, \mu_m$, one has

$$\sum_i \mu_i \le 1 \Leftrightarrow \forall \{a_i > 0\} : \sum_i \frac{a_i^2}{\mu_i} \ge \left(\sum_i a_i \right)^2.$$

Indeed, $\Rightarrow$ is given by the evident relation $\min_{\mu_i > 0 : \sum_i \mu_i \le 1} \sum_i a_i^2/\mu_i = (\sum_i a_i)^2$ for all $a_i > 0$. To verify $\Leftarrow$, set $a_i = \mu_i$ in the inequality $\sum_i a_i^2/\mu_i \ge (\sum_i a_i)^2$.

In view of (*), in order to prove that $H \in \mathcal{H}$ implies $\mathrm{Tr}(\Lambda^{-1}(H)) \le 1$, it suffices

to verify that if $H \in \mathcal{H}$ and $a_i > 0$, then $\sum_i a_i^2 \Lambda_{ii}(H) \geq (\sum_i a_i)^2$, which is immediate:

$$\begin{aligned}\sum_{i=1}^m a_i^2 \Lambda_{ii}(H) &= \sum_{i,j=1}^m a_i^2 H_{ij} = \sum_{i=1}^m a_i^2 H_{ii} + \sum_{i<j} [a_i^2 H_{ij} + a_j^2 H_{ji}] \\ &\geq \sum_i a_i^2 + 2\sum_{i<j} a_i a_j \qquad [\text{since } H_{ij} > 0, H_{ij} H_{ji} \geq 1] \\ &= \left(\sum_i a_i\right)^2. \qquad \square\end{aligned}$$

By Lemma 4.4, as H runs through $\mathcal{H}$, the matrix $\Lambda^{-1}(H)$ runs through the entire set of positive definite diagonal matrices with trace ≤ 1, so that the matrix $\omega\Lambda^{-1}(H)$ runs through the entire set of positive definite diagonal matrices with trace $\leq \omega$. Consequently, (42) implies that

$$\rho^{-1}(Q) = \min\left\{\sum_{i=1}^m \lambda_i : Q \preceq \mathrm{Diag}\{\lambda\}\right\},$$

so that $\rho^{-1}(Q) = \rho^*(Q)$ by (40). $\square$

Note that the fact that the bound (40) on the optimal value $\omega_*(Q)$ of (34) is tight within the factor $\frac{\pi}{2}$ is known; it is the "$\frac{\pi}{2}$ Theorem" of Nesterov [12], established originally via a construction based on the famous MAXCUT-related "random hyperplane" technique of Goemans and Williamson [7]. Surprisingly, the alternative proof that we have developed, although it exploits randomization, seemingly uses nothing like the random hyperplane technique.

5. Maximizing a homogeneous polynomial of degree 3 over the unit cube. Let $B[x^1, x^2, x^3]$ be a symmetric 3-linear form on $\boldsymbol{R}^m$, and let $P[x] = B[x, x, x]$ be the associated homogeneous polynomial; i.e.,

$$P[x] = \sum_{j=1}^m x_j (x^T B_j x), \quad \text{where } B_j \in S^m.$$

Consider the problem of computing

$$\omega(P) = \max_x \{P[x] : \|x\|_\infty \leq 1\}$$

along with the semidefinite program

$$\omega^*(P) = \min_{\lambda, X^1, \ldots, X^m} \left\{\sum_{j=1}^m \lambda_j : \sum_j X^j \preceq \mathrm{Diag}\{\lambda\}, X^j \succeq \pm B_j,\ j = 1, \ldots, m\right\}, \tag{43}$$

where B_j are the matrices of the symmetric bilinear forms $B[e_j, \cdot, \cdot]$; here e_j, $j = 1, \ldots, m$, are the standard basic orths in $\boldsymbol{R}^m$. We intend to demonstrate that $\omega^*(P)$ is an upper bound on $\omega(P)$, and that the quality of this bound basically depends only on the "width"

$$d(P) = \max_{1 \leq j \leq m} \mathrm{rank}(B_j)$$

of P.

THEOREM 5.1. *One has*

$$\omega(P) \le \omega^*(P) \le 4.652\vartheta(d(P))\ln(m+1)\omega(P) \le 7.31\sqrt{d(P)}\ln(m+1)\omega(P), \tag{44}$$

where $\vartheta(\cdot)$ *is given by* (9).

Proof. The proof is very much in the spirit of the matrix cube Theorem 2.1; it uses a *probabilistic* argument in order to validate the solvability/unsolvability of a certain deterministic inequality system.

1^0. Let $\lambda, X_1, \ldots, X_m$ be a feasible solution of (43). We have

$$\begin{aligned} \|x\|_\infty \le 1 \Rightarrow P[x] &= \sum_j x_j(x^T B_j x) \le \sum_j |x_j|(x^T X_j x) \le \sum_j x^T X_j x \\ &\le x^T \mathrm{Diag}\{\lambda\} x \le \sum_j \lambda_j, \end{aligned}$$

which gives the first inequality in (44).

2^0. Let us prove the second inequality in (44); without loss of generality we may assume that $P \not\equiv 0$, so that $\omega(P) > 0$. Problem (43) is strictly feasible and bounded below, so that its optimal value $\omega^*(P)$ is equal to that of its (solvable) semidefinite dual problem:

$$\omega^*(P) = \max_{U, Y_j, Z_j} \left\{ \sum_j \mathrm{Tr}([Y_j - Z_j]B_j) : \begin{array}{l} U, Y_j, Z_j \succeq 0, \\ Y_j + Z_j = U, \\ U_{jj} = 1, j = 1, \ldots, m. \end{array} \right\} \tag{45}$$

Invoking Lemma 2.2, we see that there exists U such that

$$\begin{array}{lrcl} \text{(a)} & U & \succeq & 0, \\ \text{(b)} & U_{jj} & = & 1, \qquad j = 1, \ldots, m, \\ \text{(c)} & \sum_j \|\lambda(U^{1/2} B_j U^{1/2})\|_1 & = & \omega^*(P). \end{array} \tag{46}$$

Now let $V = U^{1/2}$ and let $\xi \sim \mathcal{N}(0, I_m)$. By Lemma 2.3 and (46.c) we have

$$\vartheta(d(P)) \boldsymbol{E}\left\{ \sum_j |\xi^T V B_j V \xi| \right\} \ge \sum_j \|\lambda(V B_j V)\|_1 = \omega^*(P). \tag{47}$$

At the same time, by (46.b) the Euclidean norms of the rows of V are equal to 1, so that

$$\|V\xi\|_\infty = \max_{1 \le j \le m} |\zeta_j|, \qquad \zeta_j \sim \mathcal{N}(0,1),$$

whence, as is well known,

$$\boldsymbol{E}\left\{ \|Y\xi\|_\infty^2 \right\} \le 2\ln(m+1). \tag{48}$$

To make the paper self-contained, here is a derivation of (48). We have

$$
\begin{aligned}
& t>0 \Rightarrow \\
& \quad \psi(t) \equiv \operatorname{Prob}\{|\zeta_j|>t\} = 2\int_t^\infty \frac{1}{\sqrt{2\pi}}\exp\left\{-\frac{\tau^2}{2}\right\}d\tau \\
& \qquad \le 2\int_t^\infty \frac{1}{\sqrt{2\pi}}\frac{\tau}{t}\exp\left\{-\frac{\tau^2}{2}\right\}d\tau = \sqrt{\frac{2}{\pi}}t^{-1}\exp\left\{-\frac{t^2}{2}\right\} \\
& \Rightarrow \\
& \operatorname{Prob}\left\{\max_{j\le m}|\zeta_j|>t\right\} \le \min[1, m\psi(t)] \le \min\left[1, m\sqrt{\frac{2}{\pi}}t^{-1}\exp\left\{-\frac{t^2}{2}\right\}\right] \\
& \Rightarrow \\
& \boldsymbol{E}\left\{\max_{j\le m}|\zeta_j|^2\right\} = 2\underbrace{\int_{t>0} t\min[1, m\psi(t)]dt}_{J_m} \le 2\int_{t>0}\min\left[t, m\sqrt{\frac{2}{\pi}}\exp\left\{-\frac{t^2}{2}\right\}\right]dt \\
& \qquad \le 2\int_0^\tau t dt + 2\sqrt{\frac{2}{\pi}}m\int_\tau^\infty \exp\left\{-\frac{t^2}{2}\right\}dt \\
& \qquad \le \tau^2 + 2\sqrt{\frac{2}{\pi}}m\tau^{-1}\exp\left\{-\frac{\tau^2}{2}\right\}.
\end{aligned}
$$

The resulting bound

$$
\boldsymbol{E}\left\{\max_{j\le m}|\zeta_j|^2\right\} \le 2J_m \le \tau^2 + 2\sqrt{\frac{2}{\pi}}m\tau^{-1}\exp\left\{-\frac{\tau^2}{2}\right\} \tag{49}
$$

is valid for all m and all $\tau>0$. Assuming $m\ge 3$ and setting $\tau=\sqrt{2\ln(m/2)}$, one can easily conclude from (49) that $2J_m \le 2\ln(m+1)$ for all $m\ge 25$. Numerical computation of J_m for $m\le 25$ demonstrates that the latter inequality holds true for all m.

Combining (47), (48), we get

$$
\boldsymbol{E}\left\{\sum_j |\xi^T V B_j V\xi|\right\} \ge \frac{\omega^*(P)}{2\vartheta(d(P))\ln(m+1)}\boldsymbol{E}\left\{\|V\xi\|_\infty^2\right\}, \tag{50}
$$

and the left-hand side in this inequality is positive. It follows that there exist $\eta\in\boldsymbol{R}^m$ and a vector $\epsilon\in\boldsymbol{R}^m$ with entries ± 1 such that

$$
\eta^T\left[\sum_j \epsilon_j B_j\right]\eta \ge \frac{\omega^*(P)}{2\vartheta(d(P))\ln(m+1)} \quad \text{and} \quad \|\eta\|_\infty = 1,
$$

whence

$$
\max_{\epsilon,\eta}\{B[\epsilon,\eta,\eta] : \|\epsilon\|_\infty\le 1, \|\eta\|_\infty\le 1\} \ge \frac{\omega^*(P)}{2\vartheta(d(P))\ln(m+1)}. \tag{51}
$$

On the other hand,

$$
B[x+ty, x+ty, x+ty] = B[x,x,x] + 3tB[x,x,y] + 3t^2B[x,y,y] + t^3B[y,y,y],
$$

whence

$$\forall t \neq 0 \quad \forall x, y:$$
$$B[x,y,y] = \frac{B[x+ty, x+ty, x+ty] + B[x-ty, x-ty, x-ty] - 2B[x,x,x]}{6t^2}.$$

It follows that

$$\max_{\epsilon,\eta} \{B[\epsilon,\eta,\eta] : \|\epsilon\|_\infty \leq 1, \|\eta\|_\infty \leq 1\} \leq \frac{(1+t)^3+1}{3t^2}\omega(P) \quad \forall t > 0,$$

which combines with (51) to yield the relation

$$\frac{\omega^*(P)}{2\vartheta(d(P))\ln(m+1)} \leq \min_{t>0} \frac{(1+t)^3+1}{3t^2}\omega(P) \leq 2.326\omega(P),$$

and the second inequality in (44) follows. The third inequality follows from $\vartheta(d) \leq \frac{\pi\sqrt{d}}{2}$; see (10). □

Remark. There are two simple cases in which $d(P)$ is small. The first is when $P[x]$ is "of small rank": $P[x] = \sum_{\ell=1}^{L}(p_\ell^T x)^3$ with a small L (since clearly $d(P) \leq L$). The second case is when P is of a "band" structure; i.e., the quantity

$$\kappa = \max_{1\leq i\leq j\leq k\leq m} \{k-i : B[e_i, e_j, e_k] \neq 0\}$$

is small (since clearly $d(P) \leq 2\kappa + 1$).

REFERENCES

[1] A. Ben-Tal and A. Nemirovski, *Robust truss topology design via semidefinite programming*, SIAM J. Optim., 7 (1997), pp. 991–1016.

[2] A. Ben-Tal and A. Nemirovski, *Robust convex optimization*, Math. Oper. Res., 23 (1998), pp. 769–805.

[3] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, SIAM Stud. Appl. Math. 15, SIAM, Philadelphia, 1994.

[4] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust semidefinite programming*, in Handbook on Semidefinite Programming, R. Saigal, H. Wolkowicz, and L. Vandenberghe, eds., Kluwer Academic Publishers, Norwell, MA, 2000, pp. 139–162.

[5] M. Chilali and P. Gahinet, *H_∞ design with pole placement constraints: An LMI approach*, IEEE Trans. Automat. Control, 41 (1996), pp. 358–367.

[6] L. El Ghaoui, F. Oustry, and H. Lebret, *Robust solutions to uncertain semidefinite programs*, SIAM J. Optim., 9 (1998), pp. 33–52.

[7] M. X. Goemans and D. P. Williamson, *Improved approximation algorithms for Maximum Cut and satisfiability problems using semidefinite programming*, J. ACM, 42 (1995), pp. 1115–1145.

[8] M. Grotschel, L. Lovasz, and A. Schrijver, *The Ellipsoid Method and Combinatorial Optimization*, Springer, Heidelberg, 1988.

[9] A. Nemirovski, *Several NP-hard problems arising in robust stability analysis*, Math. Control Signals Systems, 6 (1993), pp. 99–105.

[10] A. Nemirovski, *On polynomiality of the method of analytic centers for fractional problems*, Math. Programming, 73 (1996), pp. 175–198.

[11] Yu. Nesterov and A. Nemirovski, *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM Stud. Appl. Math., SIAM, Philadelphia, 1994.

[12] Yu. Nesterov, *Semidefinite relaxation and non-convex quadratic optimization*, Optim. Methods Softw., 12 (1997), pp. 1–20.

SIAM J. OPTIM.
Vol. 12, No. 3, pp. 834–860

A TIME-STEPPING COMPLEMENTARITY APPROACH FOR FRICTIONLESS SYSTEMS OF RIGID BODIES*

JAMES A. TZITZOURIS† AND JONG-SHI PANG†

Abstract. This paper presents a time-stepping complementarity approach for computing a numerical trajectory of a system of rigid bodies under frictionless contact. Such a system is formulated as a time-dependent (ordinary) differential complementarity problem (DCP), which consists of an ordinary differential equation (ODE) coupled with a complementarity condition. The phenomenon of impact plays an essential role when modeling a mechanical system of this type. Within the time-stepping scheme, which uses a high-order ODE discretization, we include an impact detection routine and an impact law governing the behavior of the system over the duration of an impact. We report computational results for the application of the overall numerical scheme to computing the solution trajectories of several realistic mechanical systems.

Key words. contact problems, complementarity, numerical methods, time-stepping

AMS subject classifications. 74M15, 74M10, 74M20, 90C33, 74S20

PII. S1052623400370369

1. Introduction.

Inspired by the pioneering work of Lötstedt [17, 18], the study of rigid-body contact mechanics has sparked substantial interest recently; see [1, 2, 5, 11, 10, 22, 21, 23, 24, 25, 28, 27, 30, 31, 29, 33]. A general mathematical model for such mechanical problems leads to a differential complementarity problem (DCP) over time, consisting of an ordinary differential equation (ODE) in time coupled with an instantaneous complementarity problem; for more details, see the cited references. Complementarity problems involving derivatives of functions are not unprecedented in the mathematical programming literature. For example, see [14, 13] for a treatment of a parameterized complementarity problem involving derivatives that arises when modeling a particular problem in the field of structural mechanics. Such DCPs can be viewed as extensions of differential algebraic equations (DAEs), whereby bilateral constraints (equations) of the DAE are augmented by unilateral constraints (inequalities); see [4]. Unlike the contact problems involving elastic bodies (for an overview, see [16, 15]), when modeling rigid bodies the phenomenon of impact plays an essential role. Several of the papers cited above (e.g., [25, 22, 5]) utilize the power of mathematical programming to solve for the behavior of the system during impact. These theoretical models necessitate the development of numerical schemes for their solution.

Our paper is closely related to several recent works. The theoretical results in section 2, which we developed independently, closely parallel those in Chapter 6 of [10]; see also the references cited therein. The numerical algorithm presented in section 6 shares common ideas with the time-stepping schemes of Stewart and Trinkle [30, 31] and Anitescu and Potra [1]. Our work can therefore be considered a synthesis of the recent work by these authors, whereby we present both the theoretical and numerical treatment in a coherent framework. Specifically, our theory defines a "numerical solution" that is computed by our proposed algorithm.

*Received by the editors April 7, 2000; accepted for publication (in revised form) June 18, 2001; published electronically February 27, 2002.
http://www.siam.org/journals/siopt/12-3/37036.html
†Department of Mathematical Sciences, The Johns Hopkins University, Baltimore, MD (jimt2@mts.jhu.edu, jsp@vicp1.mts.jhu.edu).

Previous numerical methods for the solution of the rigid-body contact problem (e.g., [28, 22, 19, 7, 2]) mostly involve only basic low-order methods (typically related to Euler's method). A notable exception is [27], in which Stewart presents a framework for a high-order numerical scheme for ODEs with discontinuous right-hand sides. As Stewart's algorithm is designed to handle general ODEs with discontinuous right-hand sides, he does not exploit structure inherent in those ODEs resulting from models of rigid bodies. A recent work by Heemels, Schumacher, and Weiland [11] presents a time-stepping approach to solving a linear time-invariant DCP pertaining to the frictionless contact problem with affine functions (in the state variables), which describes the system dynamics and constraints.

There are several very important distinctions between our numerical scheme and those mentioned above. In particular, the time-stepping schemes of Stewart–Trinkle and Anitescu–Potra employ a fixed step-size. Recently, Stewart [28] established a convergence theory of the Stewart–Trinkle scheme as the step-size goes to zero. However, if one is interested in computing an approximate trajectory for a system of rigid bodies over time (with or without friction), there must be some flexibility in the choice of step-size from one iteration to the next.

In particular, by fixing the step-size h at the beginning of execution, such an algorithm cannot guarantee any better than $h/2$ accuracy in determining the times at which impacts occur. The error bound for the numerical calculation of impact times is first-order, regardless of the theoretical global error properties of the time-stepping scheme employed. For the purposes of a convergence analysis, the behavior of the approximate solution produced by the time-stepping scheme in the limit as $h \downarrow 0$ is of paramount interest. As a result, this first-order error in determining an impact time does not present a problem. However, if one's goal is to calculate an approximate trajectory via a high-order time-stepping scheme, this first-order error bound can become problematic.

One remedy for this difficulty is to attempt to predict or detect impact. In order to predict impact, an algorithm must estimate the state variables at the next time-step without actually executing a time-step. If an impact is detected, a new step-size is chosen (smaller than the original step-size) so that the next time-step is as close as possible to the impact time. In order to detect impact, an algorithm uses the previous and current state iterates to estimate the impact time (usually with some kind of interpolating polynomial and root-finding subroutine). After the impact time is estimated, the current state iterate is discarded and the algorithm backtracks to the previous state iterate. Then, after adjusting the step-size so that the next state iterate will occur at the estimated impact time, the algorithm steps forward. In both prediction and detection, adaptive step-sizing (at least in the neighborhood of an impact time) is vital to obtaining accurate estimates of impact times.

Although briefly mentioned in [29], such a prediction/detection/correction procedure was not part of the overall Stewart–Trinkle algorithm presented in [30]. This issue is something that plagues any attempt to calculate trajectories of systems of rigid bodies. Allowing for adaptive step-sizing allows for the possibility of impact prediction/detection (referred to as "event detection" in [27] and [11]), which can dramatically reduce the possibility of various numerical difficulties. Furthermore, almost all commercial-quality ODE solvers use adaptive step-sizing in some way to add robustness to the solver. A reasonable conclusion is that adaptive step-sizing should be a component of any practical time-stepping scheme for the calculation of trajectories of a system of rigid bodies.

In this work, we present a time-stepping framework that allows for a high-order discretization of the ODE component of the DCP used to model frictionless systems of rigid bodies. We focus on frictionless contact here as a first step toward a comprehensive investigation of rigid-body systems. A subsequent work will deal with the frictional problems. The algorithm presented in this paper relies heavily on adaptive step-sizing techniques, as we focus on obtaining physically meaningful iterates rather than on the convergence properties of the underlying time-stepping scheme as the step-size tends to zero. In order to model impact more accurately, we use an impact law related to the widely accepted impact models presented in [22, 11, 25], which are also used in [29, 1]. In [29], Stewart embeds his impact model directly into his time-stepping scheme. In this work, we draw a distinction between the impact model and the time-stepping scheme, thereby allowing end users the option of replacing our impact model with any other model of their choosing, so long as certain properties hold for the alternate model. The effectiveness of our algorithm is demonstrated on six realistic mechanical models using the trapezoidal ODE discretization technique, for which the global error is quadratic in step-size. The first two involve a sliding rod, a common example for problems of this type; e.g., see [5] and [29]. The second two examples involve a rigid double pendulum and a rigid wall. Finally, the last two examples involve two rigid carts, a rigid wall, and a hook and are borrowed from [11, 5].

Next, we discuss some notational issues. By $\|\cdot\|$ we denote the standard Euclidean norm of vectors on $\mathbb{R}^k$. By $\mathbb{R}^k_+ \subseteq \mathbb{R}^k$ we denote the nonnegative orthant in k dimensions. Given some vector $c \in \mathbb{R}^k$, by $\operatorname{diag} c$ we denote the diagonal matrix for which the ith diagonal element is given by c_i for $i = 1, \ldots, k$. Given an index set $\alpha \subseteq \{1, \ldots, k\}$, we denote the complement of α in $\{1, \ldots, k\}$ as $\bar{\alpha}$. Given a vector $q \in \mathbb{R}^k$, we denote the subvector of q that lies in the rows indexed by $\alpha \subset \{1, \ldots, k\}$ as $q_\alpha \in \mathbb{R}^{|\alpha|}$, where $|\alpha|$ is the cardinality of the index set α. We use a similar shorthand to denote the submatrices of a matrix [12]. Given a matrix $M \in \mathbb{R}^{k \times k}$ and index sets $\alpha \subseteq \{1, \ldots, k\}$ and $\beta \subseteq \{1, \ldots, k\}$, we denote the submatrix that lies in the rows of M indexed by α and the columns indexed by β as $M_{\alpha\beta}$. We say the submatrix $M_{\alpha\alpha}$ is a *principal submatrix*. By $M_{\cdot\beta}$ we mean $M_{\alpha\beta}$, where $\alpha = \{1, \ldots, k\}$, and similarly, by $M_{\alpha\cdot}$ we mean $M_{\alpha\beta}$, where $\beta = \{1, \ldots, k\}$. Note that indexing takes precedence over transposition. That is, $G^T_{\alpha\beta} = (G_{\alpha\beta})^T$. Throughout this work, we use the labeling convention $v_{i_{\mathrm{n}}}$ to represent the ith element of the vector v_{n}, where "n" is simply a label, not an index. By $v_{\alpha_{\mathrm{n}}}$ we denote the vector consisting of the entries of v_{n} lying in the index set α. Similarly, by $W_{\alpha_{\mathrm{n}}}$ we denote the matrix composed of those rows of W_{n} lying in α.

Given a function $g : \mathbb{R}^k \to \mathbb{R}^m$ that is twice differentiable on $\mathbb{R}^k$, we have the following three notational conventions. For each $l = 1, \ldots, m$, $\operatorname{grad} g_l : \mathbb{R}^k \to \mathbb{R}^k$ denotes the gradient of g_l given by $[\operatorname{grad} g_l(q)]_j = \frac{\partial g_l(q)}{\partial q_j}$ for all $q \in \mathbb{R}^k$ and $j = 1, \ldots, k$, and $\nabla^{\mathbf{2}} g_l : \mathbb{R}^k \to \mathbb{R}^{k \times k}$ denotes the Hessian of g_l given by $\left[\nabla^{\mathbf{2}} g_l(q)\right]_{ij} = \frac{\partial^2 g_l(q)}{\partial q_i \partial q_j}$ for all $q \in \mathbb{R}^k$, $i = 1, \ldots, k$, and $j = 1, \ldots, k$. Finally, $\mathbf{J}g : \mathbb{R}^k \to \mathbb{R}^{m \times k}$ denotes the Jacobian of g given by $[\mathbf{J}g(q)]_{i,j} = \frac{\partial g_i(q)}{\partial q_j}$ for all $q \in \mathbb{R}^k$, $i = 1, \ldots, m$, and $j = 1, \ldots, k$. Note that the ith row of the matrix $\mathbf{J}g(q)$ is given by the row vector $\operatorname{grad} g_i(q)^T$ for all $q \in \mathbb{R}^k$.

2. A dynamics model. In this section, we present a well-established model for the dynamics of a frictionless multi-rigid-body system with n degrees of freedom (see, for example, [5, 29]). We begin with the classical Newton–Euler ODE, given by

$$M(q)\dot{\nu} = f(q, \nu, t),$$

where $q : \mathbb{R} \to \mathbb{R}^{n_q}$ denotes the system orientation; $\nu : \mathbb{R} \to \mathbb{R}^{6n}$ denotes the system velocity; the function $f(q, \nu, t)$ represents the sum of all external, noncontact forces acting on the system; and the function $M(q)$ is the mass matrix. Additionally, we must parameterize the orientation. Letting n_q denote the number of parameters used, we define a function $G : \mathbb{R}^{n_q} \to \mathbb{R}^{6n \times n_q}$ mapping ν to $\dot{q}$ as follows:

$$\dot{q} = G(q)\nu.$$

When modeling mechanical systems with frictionless contact, the classical approach proves insufficient. One must impose unilateral nonpenetration conditions on the bodies in the system and introduce the unilateral contact forces required to maintain these conditions. Assuming that there are n_c possible contacts in the system, we define a function $\Psi_{\mathrm{n}} : \mathbb{R}^{n_q} \to \mathbb{R}^{n_c}$ and require that $\Psi_{i\mathrm{n}}$ be positive if there is a gap between the bodies at the ith contact point, zero if the bodies are touching, or negative if the bodies are interpenetrating each other. We denote the normal contact force vector by $c_{\mathrm{n}} : \mathbb{R} \to \mathbb{R}^{n_c}$. Since the normal force between two bodies should be zero when the bodies are separated, we require that $c_{i\mathrm{n}}$ be zero in this case. Since we wish to explicitly forbid interpenetration, we require that Ψ_{n} be nonnegative always. Thus, we have a complementarity relationship between Ψ_{n} and c_{n}, given by

$$0 \le \Psi_{\mathrm{n}}(q) \ \perp \ c_{\mathrm{n}} \ge 0.$$

With the inclusion of these contact forces, the Newton–Euler ODE must be modified as follows:

$$M(q)\dot{\nu} = f(q, \nu, t) + W_{\mathrm{n}}(q)^T c_{\mathrm{n}},$$

where

$$W_{\mathrm{n}}(q) = \mathbf{J}\Psi_{\mathrm{n}}(q)G(q).$$

Combining these results, the general model for a frictionless multi-rigid-body system is given by

$$\left.\begin{aligned} M(q)\dot{\nu} &= f(q, \nu, t) + W_{\mathrm{n}}(q)^T c_{\mathrm{n}}, \\ \dot{q} &= G(q)\nu, \\ 0 \le \Psi_{\mathrm{n}}(q) \ &\perp \ c_{\mathrm{n}} \ge 0 \end{aligned}\right\} \forall t \in [t_0, T], \tag{1}$$

subject to conditions given by

$$q = q^0 + \int_{t_0}^{t} \dot{q} ds \quad \text{and} \quad \nu = \nu^0 + \int_{t_0}^{t} \dot{\nu} ds, \tag{2}$$

where t_0 is the initial time, $T > t_0$ is the horizon (or final time), and the pair $\{q^0, \nu^0\}$ is the initial data.

The model given by (1) is a special kind of DCP used in the literature (for example, see [17, 5, 29, 33]) to model frictionless systems of rigid bodies. We make the following assumptions regarding DCP (1).

(A1) The function $M : \mathbb{R}^{n_q} \to \mathbb{R}^{6n\times 6n}$ is symmetric, and there exist positive constants c and c' such that $c' \|\nu\|^2 \leq \nu^T M(q)\nu \leq c \|\nu\|^2$ for all $\nu \in \mathbb{R}^{6n}$ and $q \in \mathbb{R}^{n_q}$.
(A2) The functions M^{-1} and G are globally Lipschitz, $f(\cdot,\cdot,t) : \mathbb{R}^{n_q} \times \mathbb{R}^{6n} \to \mathbb{R}^{6n}$ is globally Lipschitz for all $t \in [t_0, T]$ (with Lipschitz constant independent of t), and f is globally continuous.
(A3) The functions $\Psi_{\rm n}$, $\mathbf{J}\Psi_{\rm n}$, and $\boldsymbol{\nabla^2}\Psi_{i_{\rm n}}$ (for $i = 1, \ldots, n_c$) are globally Lipschitz.
(A4) The function $W_{\rm n}$ has full row rank uniformly on $\mathbb{R}^{n_q}$.
(A5) The initial orientation is feasible, i.e., $\Psi_{\rm n}(q^0) \geq 0$.
(A6) For $\alpha = \{i : \Psi_{i_{\rm n}}(q^0) = 0\}$, either $W_{\alpha {\rm n}}(q^0)\nu^0 \geq 0$ or $W_{\alpha {\rm n}}(q^0)\nu^0 \leq 0$.

Note that (A1) is a consequence of energy considerations, and the conditions (A2) and (A3) are common sufficiency results in order for the classical equation of motion $M(q)\dot{\nu} = f(q, \nu, t)$ to have a unique solution on $[t_0, T]$. The upper bound on the quadratic form in (A1) implies that the matrix $M(q)$ is uniformly nonsingular for all $q \in \mathbb{R}^{n_q}$. Thus the function M admits a matrix-inverse function $M^{-1} : \mathbb{R}^{n_q} \to \mathbb{R}^{6n\times 6n}$ such that $M(q)M^{-1}(q) = I$ for all $q \in R^{n_q}$, where I is the $6n \times 6n$ identity matrix. Furthermore, the upper and lower bounds on the quadratic form in (A1) imply that M and M^{-1}, respectively, are uniformly positive definite on $\mathbb{R}^{n_q}$. The condition (A4) is a constraint qualification on the nonpenetration constraint. Condition (A5) is self-explanatory, while (A6) states that, over the set of active contacts, either all normal relative velocities are nonnegative or all are nonpositive. This condition will allow us to prove some existence results later in this paper.

In the interest of rigor and consistency with Stewart (see [28]), we postulate that q is continuous and that $\dot{q}(t)$ and $\nu(t)$ are piecewise continuous with jump-discontinuities. Consequently, the derivative $\dot{\nu}(t)$ does not exist in the classical sense as the limit of the slope of a secant. Instead, we must think of $\dot{\nu}(t)$ as a generalized function involving the Dirac delta function, denoted by $\delta(t)$, satisfying (2). This explains why we incorporate the initial conditions into (2) instead of writing them explicitly in the usual way: $q(t_0) = q^0$ and $\nu(t_0) = \nu^0$. Furthermore, we must generalize our notions of nonnegativity and orthogonality accordingly. Letting $\mathcal{C}^o_+$ denote the space of all nonnegative continuous scalar-valued functions, we say that "$c_{\rm n} \geq 0$" and "$c_{\rm n} \perp \Psi_{\rm n}$" if

$$\int_{t_o}^{t} \phi(s) c_{i_{\rm n}}(s) ds \geq 0 \quad \text{and} \quad \int_{t_o}^{t} \phi(s) c_{i_{\rm n}}(s) \Psi_{i_{\rm n}}(s) ds = 0, \tag{3}$$

respectively, for all $i = 1, \ldots, n_c$ and $\phi(t) \in \mathcal{C}^o_+$.

3. An impact model. Impact occurs when two rigid bodies collide. The notion of a rigid body is an idealization, obscuring the highly complex process of restitution that occurs during the collision. An impact law is a mathematical model for this process, reinitializing the system velocities at the time of impact.

Throughout the remainder of this section, we assume that an impact occurs at $t^* \in [t_0, T]$ and use the following shorthand notation:

$$\nu^{*+} = \lim_{\tau \downarrow 0} \nu(t^* + \tau), \qquad \nu^{*-} = \lim_{\tau \downarrow 0} \nu(t^* - \tau), \qquad \text{and} \qquad q^* = q(t^*).$$

More generally, we use the superscript * to denote the value of a function at t^*. Similarly, we use the $^+$ and $^-$ superscripts to indicate right-hand and left-hand limits, respectively, at t^*. Note that if $t^* = t_0$, then we take $\nu^{*-} = \nu^0$ and $\nu^{*+} = \nu^{0+}$.

When an impact occurs at the ith contact point, we must have $\Psi_{i\mathrm{n}}(q^*) = 0$. Consequently, we define the impact index set α at t^* as follows:

$$\alpha = \{i : \Psi_{i\mathrm{n}}(q^*) = 0\}.$$

In the language of mathematical programming, α denotes the active set of the nonpenetration constraint at time t^*. To simplify notation, we also define the normal relative velocity v_{n} as follows:

$$v_{\mathrm{n}}(q, \nu) = \frac{d}{dt}\Psi_{\mathrm{n}}(q) = W_{\mathrm{n}}(q)\nu.$$

At the time of impact, the normal contact force c_{n} contains an impulsive part so that

$$\lim_{\tau \downarrow 0} \int_{t^*-\tau}^{t^*+\tau} c_{\mathrm{n}} dt = \lambda_{\mathrm{n}} \in \mathbb{R}^{n_c}.$$

We refer to λ_{n} as the impulse coefficient vector.

In this section, we present an impact law similar to those appearing in [1, 27, 29, 30]. Our goal is to capture as much physical reality as possible without overburdening the model with a high degree of complexity. To this end, we require our impact law to satisfy each of the following properties.

(R1) Momentum is conserved at the time of impact.
(R2) The total change in kinetic energy is nonpositive at the time of impact.
(R3) The condition $0 \leq \Psi_{\mathrm{n}}(q^*) \perp \lambda_{\mathrm{n}} \geq 0$ holds at the time of impact.

Integrating the ODE given by

$$\dot{\nu} = M^{-1}(q)\left(f(q, \nu, t) + W_{\mathrm{n}}^T(q)c_{\mathrm{n}}\right)$$

with respect to t from $t^* - \tau$ to $t^* + \tau$, we have

$$\int_{t^*-\tau}^{t^*+\tau} \dot{\nu} dt = \nu(t^* + \tau) - \nu(t^* - \tau) = \int_{t^*-\tau}^{t^*+\tau} M^{-1}(q)\left(f(q, \nu, t) + W_{\mathrm{n}}^T(q)c_{\mathrm{n}}\right) dt.$$

Passing to the limit at $\tau \downarrow 0$ and observing that the only impulsive term on the right is the one containing c_{n}, we have

$$\nu^{*+} - \nu^{*-} = M^{-1}(q^*)W_{\mathrm{n}}^T(q^*)\lambda_{\mathrm{n}}.$$

After some algebraic manipulation, we have

$$M(q^*)\nu^{*+} = M(q^*)\nu^{*-} + W_{\mathrm{n}}^T(q^*)\lambda_{\mathrm{n}},$$

which, along with

$$\dot{q}^{*+} = G(q^*)\nu^{*+},$$

represents conservation of momentum at the time of impact.

In order to satisfy (R3), we must require explicitly that

$$\lambda_{\alpha\mathrm{n}} \geq 0 \quad \text{and} \quad \lambda_{\bar{\alpha}\mathrm{n}} = 0.$$

Furthermore, an impact law should model the impact restitution process. To this end, for each contact $i = 1, \ldots, n_c$, we introduce the ith coefficient of restitution $\varepsilon_i \in [0, 1]$ and the unilateral restitution law

$$v_{\alpha\mathrm{n}}^{*+} + \operatorname{diag} \varepsilon_\alpha v_{\alpha\mathrm{n}}^{*-} \geq 0.$$

If $\lambda_{i_n} > 0$, we require that

$$v_{i_n}^{*+} = -\varepsilon_i v_{i_n}^{*-}.$$

We summarize these three requirements via the following complementarity problem:

$$\lambda_{\bar{\alpha}_n} = 0 \quad \text{and} \quad 0 \leq v_{\alpha_n}^{*+} + \operatorname{diag} \varepsilon_\alpha v_{\alpha_n}^{*-} \perp \lambda_{\alpha_n} \geq 0.$$

Combining the above complementarity problem with conservation of momentum, we obtain our impact law given by

$$\left.\begin{aligned} & M(q^*)\nu^{*+} = M(q^*)\nu^{*-} + W_n^T(q^*)\lambda_n, \\ & \dot{q}^{*+} = G(q^*)\nu^{*+}, \\ & \lambda_{\bar{\alpha}_n} = 0, \text{ and } 0 \leq v_{\alpha_n}^{*+} + \operatorname{diag} \varepsilon_\alpha v_{\alpha_n}^{*-} \perp \lambda_{\alpha_n} \geq 0. \end{aligned}\right\} \tag{4}$$

We now have the following theorem.

THEOREM 3.1. *The impact law given by* (4) *has a unique solution* $\{\dot{q}^{*+}, \nu^{*+}, \lambda_n\}$.

Proof. As a consequence of conservation of momentum, we have

$$v_{\alpha_n}^{*+} = v_{\alpha_n}^{*-} + W_{\alpha_n}(q^*)M^{-1}(q^*)W_{\alpha_n}^T(q^*)\lambda_{\alpha_n}.$$

Substituting for $v_{\alpha_n}^{*+}$ in (4), we obtain the following linear complementarity problem (LCP):

$$0 \leq (I + \operatorname{diag} \varepsilon_\alpha)\, v_{\alpha_n}^{*-} + W_{\alpha_n}(q^*)M^{-1}(q^*)W_{\alpha_n}^T(q^*)\lambda_{\alpha_n} \perp \lambda_{\alpha_n} \geq 0.$$

From (A3), $W_n(q^*)$ has full row rank. From (A1), $M^{-1}(q^*)$ is positive definite. Thus, for the above LCP, the defining matrix $W_{\alpha_n}(q^*)M^{-1}(q^*)W_{\alpha_n}^T(q^*)$ is positive definite. As a result, there exists a unique solution λ_{α_n} (see [8]). Thus, λ_n is uniquely determined. The uniqueness of $\dot{q}^{*+}$ and ν^{*+} follows immediately from (4). ∎

Before proceeding to the main results of this section, we prove the following proposition.

PROPOSITION 3.2. *For any impact time* $t^* > t_0$ *with nonempty impact set* α, $v_{\alpha n}^{*-} \leq 0$.

Proof. We must have that

$$\Psi_{\alpha_n}(q(t^*)) = \Psi_{\alpha_n}(q(t)) + \int_t^{t^*} v_{\alpha_n} ds = 0$$

for all $t \in [t_0, t^*]$. Since $\Psi_n \geq 0$ on $[t_0, t^*]$, we must have that

$$\int_t^{t^*} v_{\alpha_n} ds \leq 0$$

on $[t_0, t^*]$. In particular, the above inequality must hold for any arbitrarily small left-hand neighborhood of t^*. By assumption (A3), v_{α_n} is continuous on $[t_0, t^*]$, and thus we must have $v_{\alpha_n}^{*-} \leq 0$. ∎

Note that by (A6), either $v_{\alpha_n}^0 \leq 0$ or $v_{\alpha_n}^0 \geq 0$. We may disregard the latter case because, although technically an impact has occurred, $\lambda_{\alpha_n} = 0$. In effect, this signifies a purely degenerate impact.

THEOREM 3.3. *The unique solution to* (4) *satisfies* (R1), (R2), *and* (R3).

Proof. Clearly the impact law satisfies (R1) and (R3) as a result of its construction. All that remains to be shown is that (R2) is satisfied. Recall the following ODE:

$$M(q)\dot{\nu} = f(q,\nu,t) + W_{\mathrm{n}}^T(q)c_{\mathrm{n}}.$$

Taking the inner product of the above ODE with ν and then integrating with respect to t from $t^* - \tau$ to $t^* + \tau$, we have

$$\int_{t^*-\tau}^{t^*+\tau} \dot{\nu}^T M(q)\nu dt = \int_{t^*-\tau}^{t^*+\tau} \left(f(q,\nu,t)^T + c_{\mathrm{n}}^T W_{\mathrm{n}}(q)\right)\nu dt.$$

Taking the limit as $\tau \downarrow 0$ and observing that c_{n} is the only generalized function on the right-hand side, the above equation reduces to

$$\lim_{\tau\downarrow 0}\int_{t^*-\tau}^{t^*+\tau} \dot{\nu}^T M(q)\nu dt = \lim_{\tau\downarrow 0}\int_{t^*-\tau}^{t^*+\tau} c_{\mathrm{n}}^T W_{\mathrm{n}}\nu dt = \frac{1}{2}\lambda_{\mathrm{n}}^T W_{\mathrm{n}}(q^*)(\nu^{*+} + \nu^{*-});$$

see [20, 21] for details of the calculus of differential measures that yields the second equality in the above expression. Using the fact that

$$\dot{\nu}^T M(q)\nu = \frac{1}{2}\frac{d}{dt}\left(\nu^T M(q)\nu\right) - \nu^T\left(\frac{d}{dt}M(q)\right)\nu,$$

we arrive at the work-energy theorem for rigid-body impact given by

$$\frac{1}{2}(\nu^{*+})^T M(q^*)\nu^{*+} - \frac{1}{2}(\nu^{*-})^T M(q^*)\nu^{*-} = \frac{1}{2}\lambda_{\mathrm{n}}^T W_{\mathrm{n}}(q^*)\left(\nu^{*+} + \nu^{*-}\right).$$

Physically, the above equation states that over the duration of an impact, the change in kinetic energy is equal to the work done by the impulsive part of the normal force. Denoting the change in kinetic energy over the duration of the impact by ΔK and combining the work-restitution component of our impact law with the above work-energy relation, we see that the change in kinetic energy is given by

$$\Delta K = \frac{1}{2}\underbrace{\lambda_{\alpha\mathrm{n}}^T\left[I - \operatorname{diag}\varepsilon_\alpha\right]}_{\geq 0}\quad\underbrace{W_{\alpha\mathrm{n}}(q^*)\nu^{*-}}_{\leq 0,\text{ by Propostion 3.2}} \quad \leq\ 0,$$

so that the impact law given by (4) satisfies (R2). Note that Propostion 3.2 does not apply if $t^* = t_0$. However, for $t^* = t_0$, (A6) applies and either $v_{\alpha\mathrm{n}}^0 \leq 0$ or $v_{\alpha\mathrm{n}}^0 \geq 0$. In the latter case, this is a degenerate impact and $\lambda_{\alpha\mathrm{n}} = 0$, so that (R2) is still satisfied. This completes our proof. ∎

We now compare the impact law given by (4) with three well-known impact laws. If we take $\varepsilon = 0$ in (4), we recover the LCP formulation of the impact law presented in [11], which originated from Moreau (see [21]). This impact law is a special case of the impact law given by (4) and consequently satisfies (R1), (R2), and (R3). However, the impact law presented in the cited references has the unfortunate shortcoming of implicitly assuming that all coefficients of restitution are zero. Thus all rigid-body collisions are completely inelastic.

In a separate paper [22], Moreau presents an impact law that includes a dissipation index (denoted by δ) that functions like the coefficient of restitution ε. Using δ, he

defines the postimpact velocity to be a weighted average of the right-hand and left-hand limits of the velocity. This weighted average allows for the possibility of partially elastic impacts, as does our impact law. However, in [22], Moreau did not explicitly formulate a quadratic program or complementarity problem for the computation of the postimpact velocity involving δ, as is done in (4).

The third impact law, attributed to Newton and described in [25], relates the normal gap velocities before and after impact through a diagonal system of linear equations. One can then uniquely obtain the impulsive forces exerted on the system at impact by invoking conservation of momentum. Clearly Newton's law satisfies (R1) by construction. Pfeiffer and Glocker show that Newton's law satisfies (R2) but fails to guarantee nonnegative impulse coefficient vectors and therefore does not satisfy (R3); for more details, see [25].

4. A force-equilibrium model. For some fixed time t, whenever the system orientation and velocity vectors (denoted by q and ν, respectively) are known but the normal contact force (denoted by c_{n}) is not, we must rely on a force-equilibrium law to determine c_{n}. There are two situations in which this calculation is necessary. The first situation occurs at the initial time t_0. Without loss of generality, we assume that there is no impact initially. We use the same force-equilibrium approach to determine the initial value of the normal force (denoted by c_{n}^0). The second situation occurs immediately following impact. We use the impact law presented in the previous section to solve for the system velocity and the impulsive part of the normal force (denoted by λ_{n}) immediately following an impact. Since the system orientation is continuous on $[t_0, T]$, its value is the same immediately before and after impact. However, we still must determine the value of the nonimpulsive part of c_{n}. After using the impact law to reinitialize the system velocity, we must appeal to a force-equilibrium argument in order to determine the contact forces at the instant immediately following impact.

Since the same force-equilibrium approach is used in either of the above two situations, we restrict the remainder of this section to the case in which we wish to calculate the initial normal contact force. We assume that we are given initial conditions q^0 and ν^0 and that $v_{\alpha\mathrm{n}}^0 \geq 0$. We are not given initial values for the contact force vector c_{n}^0; rather, this must be calculated to be consistent with (1) at $t = t_0$ with $q(t_0) = q^0$ and $\nu(t_0) = \nu^0$. (Throughout this section, the ^{0} superscript indicates that the function in question is evaluated at $t = t_0$.) In this section, we use a limiting argument to derive a mathematical representation of this force-equilibrium law.

We begin by defining the contact index set β at time t_0 as follows:

$$\beta = \left\{ i \in \alpha : v_{i\mathrm{n}}^0 = 0 \right\} \subseteq \alpha,$$

where α is the impact index set and v_{n} is the normal relative velocity (both are defined in the previous section). Furthermore, we define the normal relative acceleration a_{n} as follows:

$$a_{\mathrm{n}} = \frac{d^2}{dt^2}\Psi_{\mathrm{n}} = \frac{d}{dt}v_{\mathrm{n}}.$$

Using a second-order Taylor series to expand the function $\Psi_{\mathrm{n}}(q(t))$ around $t = t_0$, we have

$$\Psi_{\mathrm{n}}(q(t)) = \Psi_{\mathrm{n}}(q^0) + (t - t_0)v_{\mathrm{n}}^0 + \frac{1}{2}(t - t_0)^2 a_{\mathrm{n}}^0 + o((t - t_0)^2).$$

Since $\Psi_{\bar{\alpha}\mathrm{n}}(q^0) > 0$, we must have that $c^0_{\bar{\alpha}\mathrm{n}} = 0$. For $i \in \alpha$ the above Taylor series reduces to

$$\Psi_{\alpha\mathrm{n}}(q(t)) = (t-t_0)v^0_{\alpha\mathrm{n}} + \frac{1}{2}(t-t_0)^2 a^0_{\alpha\mathrm{n}} + o((t-t_0)^2).$$

Since we have assumed $v^0_{\alpha\mathrm{n}} \geq 0$, there exists a positive scalar τ chosen sufficiently small such that $v_{\alpha\cap\bar{\beta},\mathrm{n}} > 0$ on $[t_0, t_0+\tau)$. As a result,

$$\Psi_{\alpha\cap\bar{\beta},\mathrm{n}}(q(t)) = \int_{t_0}^{t} v_{\alpha\cap\bar{\beta},\mathrm{n}} ds > 0$$

on $(t_0, t_0+\tau)$, which implies that $c_{\alpha\cap\bar{\beta},\mathrm{n}} = 0$ on $(t_0, t_0+\tau)$. Since $\Psi_{\alpha\cap\bar{\beta},\mathrm{n}}(q^0) = 0$, we are free to choose $c^0_{\alpha\cap\bar{\beta},\mathrm{n}} = 0$, thus ensuring that $c_{\alpha\cap\bar{\beta},\mathrm{n}}$ is continuous at t_0. For $i \in \beta$, the Taylor series further reduces to

$$\Psi_{\beta\mathrm{n}}(q(t)) = \frac{1}{2}(t-t_0)^2 a^0_{\beta\mathrm{n}} + o((t-t_0)^2).$$

Note that, since

$$\Psi_{\beta\mathrm{n}}(q(t)) \geq 0 \iff \frac{2}{(t-t_0)^2}\Psi_{\beta\mathrm{n}}(q(t)) \geq 0,$$

we require that

$$\frac{2}{(t-t_0)^2}\Psi_{\beta\mathrm{n}}(q(t)) = a^0_{\beta\mathrm{n}} + 2\frac{o((t-t_0)^2)}{(t-t_0)^2} \geq 0$$

in a right-hand neighborhood of t_0 chosen sufficiently small. Taking the limit as $t \downarrow t_0$ and substituting the result into the complementarity condition in (1), we arrive at the frictionless force-equilibrium law, given by

$$\left.\begin{aligned} &M(q^0)\dot{\nu}^0 = f(q^0,\nu^0,t_0) + W^T_{\beta\mathrm{n}}(q^0)c^0_{\beta\mathrm{n}}, \\ &c^0_{\bar{\beta}\mathrm{n}} = 0 \ \text{ and } \ 0 \leq a^0_{\beta\mathrm{n}} \perp c^0_{\beta\mathrm{n}} \geq 0. \end{aligned}\right\} \tag{5}$$

Now we have the following theorem guaranteeing the existence and uniqueness of a solution to the frictionless force-equilibrium law (5).

THEOREM 4.1. *The force-equilibrium law* (5) *has a unique solution* $\{\dot{\nu}^0, c^0_n\}$.

Proof. By (A1), $M^{-1}(q^0)$ exists, and thus solving for $\dot{\nu}^0$ and substituting the result into the expression for $a^0_{\beta\mathrm{n}}$, we have

$$\begin{aligned} a^0_{\beta\mathrm{n}} = {} & W_{\beta\mathrm{n}}(q^0)M^{-1}(q^0)f(q^0,\nu^0,t_0) + \left[(G(q^0)\nu^0)^T\nabla^{\mathbf{2}}\Psi_{i\mathrm{n}}(q^0)G(q^0)\nu^0\right]_{i\in\beta} \\ & + W_{\beta\mathrm{n}}(q^0)M^{-1}(q^0)W_{\beta\mathrm{n}}(q^0)c^0_{\beta\mathrm{n}}. \end{aligned}$$

Thus, the complementarity problem in (5) is an LCP with the defining matrix $W_{\beta\mathrm{n}}(q^0)M^{-1}(q^0)W_{\beta\mathrm{n}}(q^0)$. By (A1), $M^{-1}(q^0)$ is positive definite and, by (A3), $W_{\beta\mathrm{n}}(q^0)$ has full row rank. As a result, the defining matrix must be positive definite. Thus $c^0_{\beta\mathrm{n}}$ exists uniquely (see [8] for details). Then $\dot{\nu}^0$ follows uniquely from (5). ∎

5. Frictionless locally smooth solutions. In this section, we develop the concept of a locally smooth solution to the frictionless multi-rigid-body contact problem given by (1). After stating a precise definition, we state and prove results regarding the existence of such a solution. As we will see, these results play vital roles in section 6, where we define what is meant by a numerical solution to (1).

DEFINITION 5.1. *Given some initial time t_0 and initial data pair $\{q^0, \nu^0\}$, suppose there exist some positive scalar τ and a function triple $\{q, \nu, c_n\}$ that satisfy the following conditions:*

(C1) *$q : \mathbb{R} \to \mathbb{R}^{n_q}$ is continuously differentiable on $[t_0, t_0 + \tau)$ with $q(t_0) = q^0$;*

(C2) *$\nu : \mathbb{R} \to \mathbb{R}^{6n}$ is continuously differentiable on $[t_0, t_0 + \tau)$ with $\nu(t_0) = \nu^0$;*

(C3) *$c_n : \mathbb{R} \to \mathbb{R}^{n_c}$ is Lipschitz on $[t_0, t_0 + \tau)$; and*

(C4) *$\{q, \nu, c_n\}$ satisfies (1) on $[t_0, t_0 + \tau)$.*

Then, we say that $\{q, \nu, c_n\}$ is a locally smooth solution *to the frictionless multi-rigid-body contact problem on $[t_0, t_0 + \tau)$ with initial conditions $\{q^0, \nu^0\}$.*

As with the related field of DAEs (see [4]), it is not true in general that all possible initial data lead to locally smooth solutions. Initial data that do so are said to be consistent (borrowing terminology from the field of DAEs). Using Definition 5.1, we define consistency as follows.

DEFINITION 5.2. *We say that the pair $\{q^0, \nu^0\} \in \mathbb{R}^{n_q} \times \mathbb{R}^{6n}$ is* consistent *at time t_0 if there exists a positive scalar τ such that a locally smooth solution exists on $[t_0, t_0 + \tau)$ with initial conditions $\{q^0, \nu^0\}$.*

We have the following necessary condition regarding consistent initial conditions.

THEOREM 5.3. *If the pair $\{q^0, \nu^0\}$ is consistent at $t = t_0$, then*

$$v^0_{\alpha n} = W_{\alpha n}(q^0)\nu^0 \geq 0,$$

where the index set α is defined as

$$\alpha = \{i : \Psi_{in}(q^0) = 0\}.$$

Proof. From Definition 5.2, there exist a positive scalar τ_1 and a locally smooth solution $\{q, \nu, c_n\}$ on $[t_0, t_0 + \tau_1)$. As an immediate consequence, we have $\Psi_n(q) \geq 0$ on $[t_0, t_0 + \tau_1)$. Suppose there exists an index $i \in \alpha$ such that $v^0_{in} < 0$. Since $i \in \alpha$, we have

$$\Psi_{in}(q) = \int_{t_0}^{t} v_{in} ds \qquad \forall\, t \in [t_0, t_0 + \tau_1).$$

The functions q and ν, and as a result v_n, are continuously differentiable on $[t_0, t_0 + \tau_1)$. Thus, there exists a positive scalar $\tau_2 \leq \tau_1$ such that $v_{in} < 0$ on $[t_0, t_0 + \tau_2)$. Thus we have

$$\Psi_{in}(q) = \int_{t_0}^{t} v_{in} ds < 0 \qquad \forall\, t \in [t_0, t_0 + \tau_2),$$

contradicting the previously established nonnegativity of $\Psi_{in}(q)$ on $(t_0, t_0 + \tau_1)$. Therefore, $v_{\alpha n} \geq 0$, concluding our proof. ∎

Whether the converse to Theorem 5.3 holds remains an open question. However, we can make a physically reasonable assumption and prove a slightly weaker version. In order to do so, we must assume that after using the force-equilibrium law given by (5), the relative normal acceleration a_n and the normal contact force c_n are nondegenerate when restricted to the index set $\beta = \{i : \Psi_{in}(q^0) = v^0_{in} = 0\}$. This additional assumption gives us the following extremely useful sufficiency condition.

THEOREM 5.4. *Suppose that the following two conditions hold.*

(S1) *For* $\alpha = \{i : \Psi_{in}(q^0) = 0\}$, *we have that* $v^0_{\alpha n} = W_{\alpha n}(q^0)\nu^0 \geq 0$.

(S2) *For all* $i \in \beta = \{i \in \alpha : v^0_{in} = 0\}$, $a^0_{in} = 0$ *implies* $c^0_{in} > 0$.

Then the pair $\{q^0, \nu^0\}$ *is consistent at* $t = t_0$.

Proof. To begin, we define the function

$$b(q,\nu,t) = \left[\nu^T G(q)^T \nabla^{\mathbf{2}} \Psi_{in}(q) G(q)\nu\right]_{i\in\beta} + W_{\beta n}(q) M^{-1}(q) f(q,\nu,t)$$

and the function

$$A(q) = W_{\beta n}(q) M^{-1}(q) W^T_{\beta n}(q).$$

Defining the state vector

$$y = \begin{bmatrix} q \\ \nu \end{bmatrix} : \mathbb{R} \to \mathbb{R}^{n_q+6n},$$

consider the finite-dimensional functional linear complementarity problem (FLCP) given by

$$0 \leq b(y,t) + A(y)x \ \perp \ x \geq 0. \tag{6}$$

Since M^{-1} is uniformly positive definite on $\mathbb{R}^{n_q}$ by assumption (A1), and since W_n has full row rank uniformly on $\mathbb{R}^{n_q}$ by assumption (A4), it is an elementary result that A is uniformly positive definite on $\mathbb{R}^{n_q}$. By assumptions (A1), (A2), and (A4), A and $b(\cdot,t)$ are both Lipschitz on any closed, bounded set $\mathcal{B} \subseteq \mathbb{R}^{n_q+6n}$. Then by Theorem A.3, for every t there exists a unique function $x(\cdot,t) : \mathbb{R}^{n_q+6n} \to \mathbb{R}^{n_c}$ satisfying (6) that is itself Lipschitz on $\mathcal{B}$. By the same proof as in Theorem A.3, we can establish that $x(y,t)$ is jointly continuous in (y,t). Therefore, by Picard's existence and uniqueness theorem for ODEs (see [9], for example), there exist a positive scalar τ_1 chosen sufficiently small and a unique differentiable function $y : \mathbb{R} \to \mathbb{R}^{n_q+6n}$ on $[t_0, t_0+\tau_1)$ satisfying the ODE

$$\dot{y} = \begin{bmatrix} \begin{bmatrix} 0 & I \end{bmatrix} y \\ M^{-1}(y)\left(f(y,t) + W^T_{\beta n}(y)x(y)\right) \end{bmatrix},$$

subject to the initial condition

$$y(t_0) = y^0 \equiv \begin{bmatrix} q^0 \\ \nu^0 \end{bmatrix},$$

where, for every $t \in [t_0, t_0+\tau_1)$, the vector $y(t) \in \mathcal{B}$. Thus, x is also continuous on $[t_0, t_0+\tau_1)$. Consequently, we have that on $[t_0, t_0+\tau_1)$ there exist continuously differentiable functions $q : \mathbb{R} \to \mathbb{R}^{n_q}$ and $\nu : \mathbb{R} \to \mathbb{R}^{6n}$, as well as a continuous function $x : \mathbb{R} \to \mathbb{R}^{|\beta|}$ satisfying the differential complementarity system given by

$$\left.\begin{aligned} M(q)\dot{\nu} &= f(q,\nu,t) + W_{\beta n}(q)^T x, \\ \dot{q} &= G(q)\nu, \\ 0 \ \leq \ b(q,\nu,t) + A(q)x \ &\perp \ x \ \geq \ 0, \\ q(t_0) = q^0, \quad &\text{and} \quad \nu(t_0) = \nu^0. \end{aligned}\right\} \tag{7}$$

At this point, we remark that at $t = t_0$, (7) reduces to the force equilibrium law. Therefore we must have that $x(t_0) = c^0_{\beta \mathrm{n}}$.

By hypothesis, $\Psi_{\bar{\alpha} \mathrm{n}}(q^0) > 0$ and by (A3), Ψ_{n} is continuous on $[t_0, t_0 + \tau_1)$, so there exists a positive scalar $\tau_2 \leq \tau_1$ such that $\Psi_{\bar{\alpha} \mathrm{n}}(q) \geq 0$ on $[t_0, t_0 + \tau_2)$. The function v_{n} is continuously differentiable on $[t_0, t_0 + \tau_2)$ by (A3). Since $v^0_{\alpha \cap \bar{\beta}, \mathrm{n}} > 0$, there exists a positive scalar $\tau_3 \leq \tau_2$ such that $v_{\alpha \cap \bar{\beta}, \mathrm{n}} > 0$ on $[t_0, t_0 + \tau_3)$. Thus we have

$$\Psi_{\alpha \cap \bar{\beta}, \mathrm{n}}(q) = \int_{t_0}^{t} v_{\alpha \cap \bar{\beta}, \mathrm{n}} ds > 0$$

on $[t_0, t_0 + \tau_3)$. Since q and ν are continuously differentiable on $[t_0, t_0 + \tau_3)$, $a_{\beta \mathrm{n}} = b(q, \nu, t) + A(q)x \geq 0$ is continuous on $[t_0, t_0 + \tau_3)$. Thus, we have

$$\Psi_{\beta \mathrm{n}}(q) = \int_{t_0}^{t} \int_{t_0}^{s} a_{\beta \mathrm{n}} dr ds \geq 0$$

on $[t_0, t_0 + \tau_3)$. Combining these three results involving Ψ_{n}, we have $\Psi_{\mathrm{n}}(q) \geq 0$ on $[t_0, t_0 + \tau_3)$.

Since $c^0_{\beta \mathrm{n}}$ and $a^0_{\beta \mathrm{n}}$ are nondegenerate and $x(t_0) = c^0_{\beta \mathrm{n}}$, it must be true that $x(t_0)$ and $a^0_{\beta \mathrm{n}}$ are also nondegenerate. If $x_i(t_0) > 0$, then there exists a positive scalar $\tau_4 \leq \tau_3$ such that $x_i > 0$ on $[t_0, t_0 + \tau_4)$. Therefore $a_{i \mathrm{n}} = 0$ on $[t_0, t_0 + \tau_4)$, so that

$$\Psi_{i \mathrm{n}}(q) = \int_{t_0}^{t} \int_{t_0}^{s} a_{i \mathrm{n}} dr ds = 0$$

on $[t_0, t_0 + \tau_4)$. Similarly, if $x_i(t_0) = 0$, then $a^0_{i \mathrm{n}} > 0$ and there exists a positive scalar $\tau_5 \leq \tau_4$ such that $a_{i \mathrm{n}} > 0$ on $[t_0, t_0 + \tau_5)$. Therefore $x_i = 0$ on $[t_0, t_0 + \tau_5)$. Summarizing, we have shown that

$$0 \leq \Psi_{\beta \mathrm{n}}(q) \perp x \geq 0 \qquad \forall\, t \in [t_0, t_0 + \tau_5).$$

Then, taking $c_{\beta \mathrm{n}} = x$ and $c_{\bar{\beta} \mathrm{n}} = 0$, we must have that $\{q, \nu, c_{\mathrm{n}}\}$ is a local solution on $[t_0, t_0 + \tau_5)$. In particular, note that $\{q, \nu, c_{\mathrm{n}}\}$ is unique. Thus, the pair $\left\{q^0, \nu^0\right\}$ is consistent. □

As a result of Theorem 5.3, if the initial conditions $\left\{q^0, \nu^0\right\}$ are not consistent, then $v^0_{\alpha \mathrm{n}} \not\geq 0$. Then, by (A6), we must have that $v^0_{\alpha \mathrm{n}} \leq 0$ with $v^0_{i \mathrm{n}} < 0$ for at least one $i \in \alpha$. However, the following corollary to Theorem 5.4 shows that if this is the case, then the impact law given by (4) may be able to reinitialize the system velocities in such a way as to produce a consistent pair of initial conditions given by $\left\{q^0, \nu^{0+}\right\}$.

COROLLARY 5.5. *Suppose that an impact occurs at time t_0 and that ν^{0+} is produced from $\left\{q^0, \nu^0\right\}$ via the impact law* (4). *Further suppose that the pair $\left\{c^0_n, a^0_n\right\}$ is calculated from $\left\{q^0, \nu^{0+}\right\}$ via the force-equilibrium law* (5). *If $a^0_{in} = 0$ implies that $c^0_{in} > 0$ for all $i \in \beta = \left\{i : \Psi_{in}(q^0) = v^0_{in} = 0\right\}$, then the pair $\left\{q^0, \nu^{0+}\right\}$ is consistent.*

Proof. By (A6), either $v^0_{\alpha \mathrm{n}} \geq 0$ or $v^0_{\alpha \mathrm{n}} \leq 0$, where $\alpha = \left\{i : \Psi_{i \mathrm{n}}(q^0) = 0\right\}$. In the former case, Theorem 5.4 applies directly. Without loss of generality, we assume the latter case. From the complementarity condition in (4), we have that

$$v^{0+}_{\alpha \mathrm{n}} + \varepsilon_\alpha v^0_{\alpha \mathrm{n}} \geq 0.$$

Since $v^0_{\alpha \mathrm{n}} \leq 0$, we must have that $v^{0+}_{\alpha \mathrm{n}} \geq 0$. Thus, by Theorem 5.4, the pair $\left\{q^0, \nu^{0+}\right\}$ is consistent. □

In the next section, we will present a time-stepping method for the numerical simulation of the system trajectories. In particular, we will make use of the concept of consistent data when defining a numerical solution. As the above corollary indicates, the impact law will be used to reinitialize the system whenever a locally smooth solution fails to exist.

6. A numerical algorithm. In this section, we present a time-stepping algorithm for the numerical simulation of the frictionless multi-rigid-body contact problem given by (1). It is well known (for example, see [4]) that one-step methods are better suited than their multistep counterparts for time-stepping algorithms that encounter frequent discontinuities in the unknown function. This is because, at a bare minimum, all time-stepping methods assume the state variable to be continuous on the interval from which data is drawn. When the state estimates at the previous time-step are consistent, this continuity property holds for sufficiently small forward interval. Thus a one-step method is applicable. However, a p-step ($p > 1$) method requires that the state variables be sufficiently smooth over the interval $[t_{m-p+1}, t_{m+1}]$. Since this requirement is likely to be violated when simulating a multi-rigid-body contact problem, we restrict our attention to one-step methods. In particular, we focus on Runge–Kutta methods (see [3, 26, 4, 6]).

In particular, we focus on the trapezoidal discretization, given by

$$\left.\begin{aligned} q^{m+1} &= q^m + \frac{h}{2}(\dot{q}^m + \dot{q}^{m+1}),\\ \nu^{m+1} &= \nu^m + \frac{h}{2}(\dot{\nu}^m + \dot{\nu}^{m+1}), \end{aligned}\right\} \tag{8}$$

where m is the index of the time-step t_m, q^m is the estimate of $q(t_m)$, and ν^m is the estimate of $\nu(t_m)$. The initial estimates q^0 and ν^0 are precisely the initial conditions given to the original (continuous) frictionless multi-rigid-body contact problem (1). If there is an impact at t_0, we can use our impact law (4) to reinitialize the system velocities at t_0. This can be done "off-line." Thus, for the purposes of a time-stepping scheme, we may assume without loss of generality that no impact occurs at t_0.

Our approach to discretizing (1) is very similar to that taken in [1, 2, 31, 29]. First, we calculate $\{\hat{q}, \hat{\nu}\}$, our predicted estimates of $q(t_{m+1})$ and $\nu(t_{m+1})$, respectively, by solving the following implicit system of equations:

$$\left.\begin{aligned} M(\hat{q})\hat{\dot{\nu}} &= f(\hat{q}, \hat{\nu}, t_{m+1}) + W_{\mathrm{n}}(\hat{q})c_{\mathrm{n}}^m,\\ \hat{\nu} &= \nu^m + \frac{h}{2}(\dot{\nu}^m + \hat{\dot{\nu}}),\\ \hat{q} &= q^m + \frac{h}{2}(G(q^m)\nu^m + G(\hat{q})\hat{\nu}). \end{aligned}\right\} \tag{9}$$

Using $\{\hat{q}, \hat{\nu}\}$, we then obtain c_{n}^{m+1}, our estimate of $c_{\mathrm{n}}(t_{m+1})$, by solving the following contact LCP:

$$\left.\begin{aligned} a_{\beta\mathrm{n}}^{m+1} &= \left[\hat{\nu}^T G(\hat{q})^T \nabla^{\mathbf{2}} \Psi_{i\mathrm{n}}(\hat{q}) G(\hat{q})\hat{\nu}\right]_{i\in\beta}\\ &\quad + W_{\beta\mathrm{n}}(\hat{q})M^{-1}(\hat{q})f(\hat{q}, \hat{\nu}, t_{m+1}) + W_{\beta\mathrm{n}}(\hat{q})M^{-1}(\hat{q})W_{\beta\mathrm{n}}^T(\hat{q})c_{\beta\mathrm{n}}^{m+1},\\ c_{\bar{\beta}\mathrm{n}}^{m+1} &= 0, \quad \text{and} \quad 0 \le a_{\beta\mathrm{n}}^{m+1} \perp c_{\beta\mathrm{n}}^{m+1} \ge 0. \end{aligned}\right\} \tag{10}$$

Finally, we correct our rough estimates $\{\hat{q}, \hat{\nu}\}$ by solving the following implicit system of equations for q^{m+1} and ν^{m+1}:

$$\left.\begin{aligned} M(q^{m+1})\dot{\nu}^{m+1} &= f(q^{m+1}, \nu^{m+1}, t_{m+1}) + W_{\mathrm{n}}(q^{m+1})c_{\mathrm{n}}^{m+1}, \\ \nu^{m+1} &= \nu^m + \frac{h}{2}(\dot{\nu}^m + \dot{\nu}^{m+1}), \\ q^{m+1} &= q^m + \frac{h}{2}(G(q^m)\nu^m + G(q^{m+1})\nu^{m+1}). \end{aligned}\right\} \tag{11}$$

Before we proceed to the time-stepping algorithm, we define what is meant by a numerical solution to the frictionless multi-rigid-body contact problem as follows.

DEFINITION 6.1. *The set of tuples* $\{\{q^m, \nu^m, u^m, t_m\}\}$ *indexed by* m *is said to be a* numerical solution *to the frictionless multi-rigid-body contact problem* (1) *on* $[t_0, T]$ *if the following three conditions are satisfied.*

(N1) *The set of tuples* $\{\{q^m, \nu^m, u^m, t_m\}\}$ *satisfies* (9), (10), *and* (11).

(N2) *The set* $\mathcal{T} = \{t_m\}$ *is a countable subset of* $[t_0, T]$ *such that* $t_0, T \in \mathcal{T}$. *Also, for any* $t_m, t_n \in \mathcal{T}$, *if* $m < n$, *then* $t_m < t_n$.

(N3) *For each tuple* $\{q^m, \nu^m, u^m, t_m\}$, *the pair* $\{q^m, \nu^m\}$ *is consistent at* t_m.

In the above definition, note that $\{t_m\}$ could be countably infinite. We will discuss the consequences of an accumulation point in $\{t_m\}$ later in this section. First, we present the following time-stepping algorithm.

Step 0. Given h_0, T, ε, q^0, and ν^0, use the force-equilibrium law given by (5) to solve for the initial acceleration vector $\dot{\nu}^0$ and normal contact force vector c_{n}^0. Set $k \leftarrow 0$ and proceed to Step 1.

Step 1. Use a quadratic Taylor series approximation of Ψ_{n} around t_k to predict impact in the $(k+1)$st time-step. If an impact is predicted, determine the time t^* of earliest occurrence and set $h_{k+1} \leftarrow t^* - t_k$. Otherwise, set $h_{k+1} \leftarrow h_0$. In either case, set $t_{k+1} \leftarrow t_k + h_{k+1}$ and proceed to Step 2.

Step 2. If an impact was predicted in Step 1, augment the prediction equations given by (9) by adding the scalar equation $\Psi_{i\mathrm{n}}(\hat{q}) = 0$, where i is the index of any contact point at which an impact is about to occur. Note that we now treat the step-size h_{k+1} as an unknown. Solve the resulting augmented system for h_{k+1}, $\hat{q}$, and $\hat{\nu}$. If no impact was predicted in Step 1, solve (9) for $\hat{q}$ and $\hat{\nu}$. Proceed to Step 3.

Step 3. Solve (10) to obtain c_{n}^{k+1} and proceed to Step 4.

Step 4. Solve the correction equations given by (11) to obtain q^{k+1} and ν^{k+1}. Proceed to Step 5.

Step 5. If $\Psi_{i\mathrm{n}}(q^{k+1}) < 0$ for some $i = 1, \ldots, n_c$, then set $h_{k+1} \leftarrow h_{k+1}/2$ and go back to Step 1. Otherwise, check for impact. If an impact occurs at t_{k+1}, use the impact law (4) to reinitialize ν^{k+1}, and the force-equilibrium law (5) to solve for the acceleration vector $\dot{\nu}^{k+1}$ and contact force vector c_{n}^{k+1}. Proceed to Step 6.

Step 6. Set $k \leftarrow k + 1$. If $t_k < T$, proceed to Step 1. Otherwise, return.

A few comments regarding the above algorithm are in order.

First, we use Newton's method to solve all nonlinear systems of equations. Instead of using analytic formulae for the Jacobian matrix required by the Newton subroutine, we implement a five-point centered difference approximation. Note that, in practice, usually no more than one or two Newton iterations are needed. However, it is a

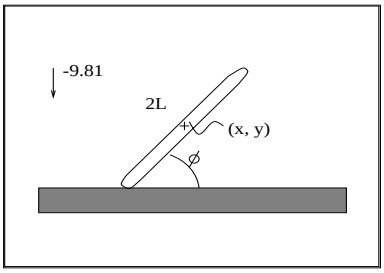
good idea to place an upper bound on the number of iterations performed per time-step. In our implementation, we use Lemke's method for the solution of all LCPs (see [8]).

Second, the set $\mathcal{T}_{\text{jump}} \cap [t_0, T]$ may contain accumulation points (which we refer to as "accumulation times"). To illustrate, let us consider the case of a vertically bouncing ball on a hard surface, as was communicated by Trinkle [32]. With each bounce, the ball loses energy, and thus the apex of its trajectory is lower and lower with each subsequent bounce. It can be shown with a bit of algebra that the intervals between jump-times for this system represent a geometrically decreasing sequence, and that the limit of the partial sums of that sequence is finite. Therefore, the ball comes to a complete stop in finite time. One possible approach to dealing with this complication is to keep a round-off threshold, as discussed in [1]. Effectively, when the maximum height of the ball above the table is below this threshold, we round the height to zero. In our implementation, we use this approach. Another possible approach suggested by Trinkle [32] is to set the coefficient of restitution to zero temporarily when the normal relative velocity before impact (denoted by v_{in}^{*-}) is below a prescribed threshold.

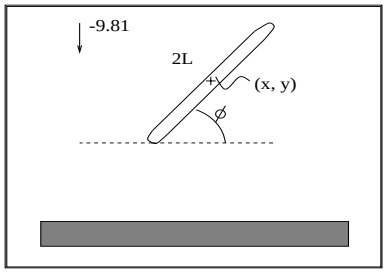
At this point, we note that the algorithm described above specifically uses the implicit trapezoidal one-step method, which has a global error of $O(h^2)$ when applied to ODEs. However, one could replace all instances of the trapezoidal method with other, higher-order (when applied to ODEs) one-step methods. However, this will result in a larger nonlinear system of equations that must be solved at every iteration. Note that any discussion of global error is in reference to the global error properties of those methods applied to ODEs, not to DCPs. In the next section, we present several examples of this algorithm.

7. Numerical results. In this section, we present six examples of rigid-body mechanical systems to which we apply the time-stepping complementarity algorithm described in section 6. The first four of these examples involve nonlinear constraints, and the last two examples have simple linear constraints. Note that, in all examples, $G(q)$ is the square identity matrix so that $\nu = \dot{q}$.

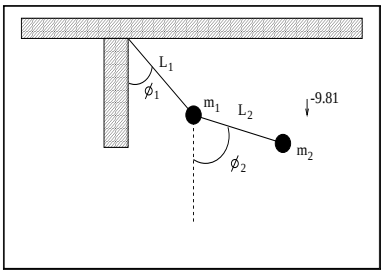
7.1. A rigid rod and an immovable table. The first two systems to which we apply our algorithm are the "sliding rod" and "falling rod" (Figure 1); for more information about the history of this class of contact problem, see [5] or [28]. Both systems have the same dynamics and gap functions, as well as the same coefficients of restitution, but differ in initial data. Both systems have three degrees of freedom and consist of a rigid rod of length $2L$ with total mass m (with uniform mass density) and an immovable horizontal surface (henceforth referred to as "the table"). In both systems, we define the angle ϕ to be the angle formed at the unique intersection of the left-most end point of the rod with a plane parallel to the horizontal surface of the table. The remaining two degrees of freedom are covered by the position of the center of mass of the rod, given by the pair (x, y). Note that since we assume the contact to be frictionless, there are no forces acting in the horizontal direction. If we define the generalized coordinate vector q as $q = [\ x \quad y \quad \phi\ ]^T$, then the gap function is given by $\Psi_{\text{n}}(q) = [\ q_2 - L\sin(q_3) \quad q_2 + L\sin(q_3)\ ]^T$, the mass matrix $M(q)$ is given by $M(q) = m\,\text{diag}[\ 1 \quad 1 \quad \frac{1}{3}L^2\ ]^T$, and the total noncontact force exerted on the system is given by $f(q, \dot{q}, t) = [\ 0 \quad -9.81m \quad 0\ ]^T$, where the gravitational acceleration constant is 9.81 meters per second per second. In the sliding rod system, the left-most end point of the rod begins in contact with the table (Figure 1(a)). In



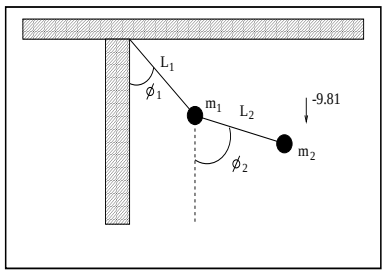
(a) Sliding rod.



(b) Falling rod.

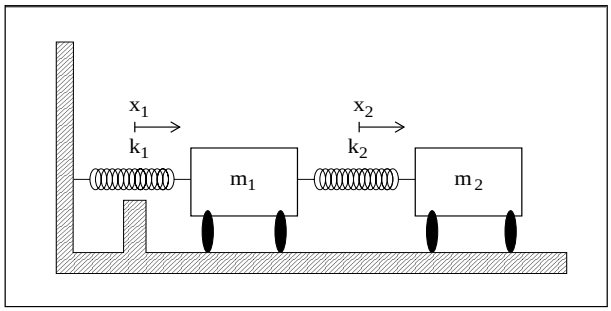


(c) Double pendulum with short wall.

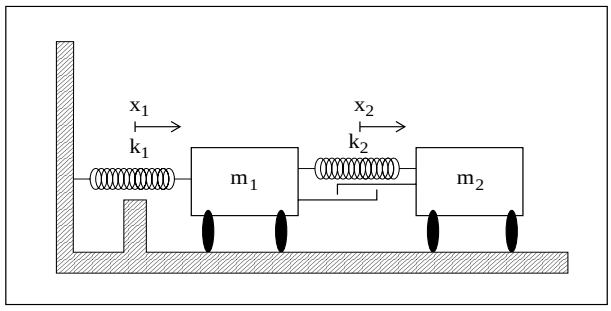


(d) Double pendulum with long wall.

FIG. 1. *Examples involving rods or pendulums.*



(a) Two carts and a stopper.



(b) Two carts, a stopper, and a hook.

FIG. 2. *Examples involving two carts.*

TABLE 1
Simulation parameters.

Simulation	Parameters
Sliding rod	$L = m = 1$
Falling rod	$L = m = 1$
Double pendulum, short wall	$L_1 = L_2 = m_1 = m_2 = 1$
Double pendulum, long wall	$L_1 = L_2 = m_1 = m_2 = 1$
Two carts & wall	$m_1 = m_2 = k_1 = k_2 = 1$
Two carts, wall, & hook	$m_1 = m_2 = k_1 = k_2 = 1$

TABLE 2
Simulation initial data.

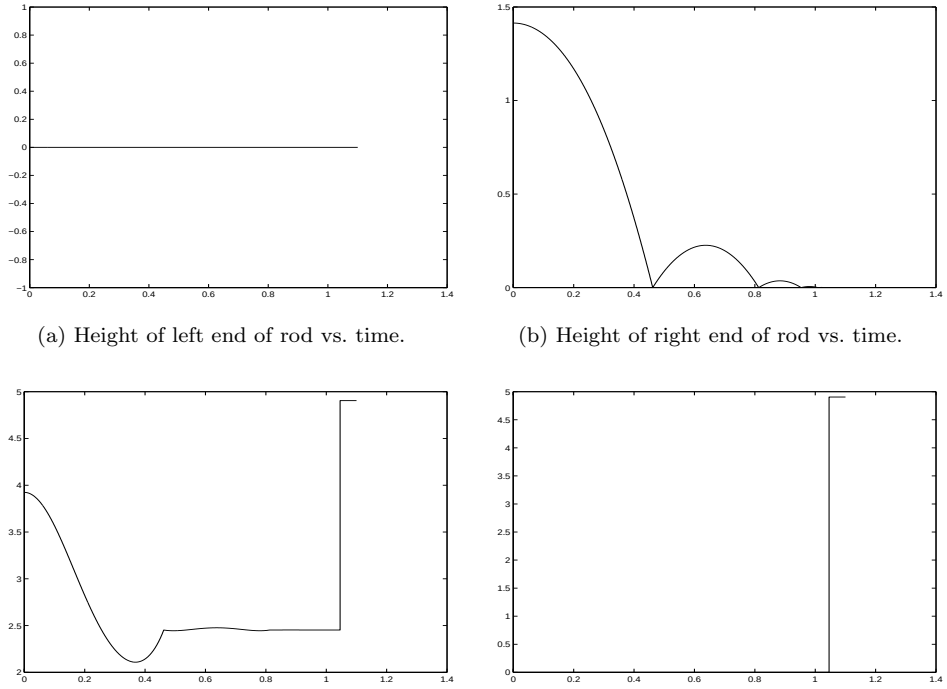
Simulation	Initial data
Sliding rod	$q^0 = \begin{bmatrix} \frac{L}{\sqrt{2}} & \frac{L}{\sqrt{2}} & \frac{\pi}{4} \end{bmatrix}^T, \quad \dot{q}^0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$
Falling rod	$q^0 = \begin{bmatrix} \frac{L}{\sqrt{2}} & \frac{1}{2} + \frac{L}{\sqrt{2}} & \frac{\pi}{4} \end{bmatrix}^T, \quad \dot{q}^0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$
Double pendulum & short wall	$q^0 = \begin{bmatrix} \frac{\pi}{3} & \frac{\pi}{5} \end{bmatrix}^T, \quad \dot{q}^0 = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$
Double pendulum & long wall	$q^0 = \begin{bmatrix} \frac{\pi}{3} & \frac{\pi}{5} \end{bmatrix}^T, \quad \dot{q}^0 = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$
Two carts & wall	$q^0 = \begin{bmatrix} 0.32024033 \\ -0.43350467 \end{bmatrix}, \quad \dot{q}^0 = \begin{bmatrix} 0.37155103 \\ -1.09145060 \end{bmatrix}$
Two carts, wall, & hook	$q^0 = \begin{bmatrix} 0.32024033 \\ -0.43350467 \end{bmatrix}, \quad \dot{q}^0 = \begin{bmatrix} 0.37155103 \\ -1.09145060 \end{bmatrix}$

TABLE 3
Coefficients of restitution used in simulations.

Simulation	Coefficients of restitution
Sliding rod	$\varepsilon = \begin{bmatrix} 0.4 & 0.4 \end{bmatrix}^T$
Falling rod	$\varepsilon = \begin{bmatrix} 0.4 & 0.4 \end{bmatrix}^T$
Double pendulum & short wall	$\varepsilon = 0.3$
Double pendulum & long wall	$\varepsilon = \begin{bmatrix} 0.1 & 0.1 \end{bmatrix}^T$
Two carts & wall	$\varepsilon = 0.3$
Two carts, wall, & hook	$\varepsilon = \begin{bmatrix} 0.3 & 0.05 \end{bmatrix}^T$

the falling rod system, the rod begins with neither end in contact with the table, but rather, is dropped from a positive height onto the table (Figure 1(b)).

7.2. A double pendulum with an immovable wall. The next two systems that we consider involve a double pendulum consisting of a ceiling (anchor), two point-masses, and two massless rods (Figure 1). The top mass (m_1) is connected via a massless rigid rod of length L_1 to the ceiling. The bottom mass (m_2) is connected via a massless rigid rod of length L_2 to the top mass. We define ϕ_1 as the angle between the vertical and the rod connecting the top mass to the ceiling, and ϕ_2 as the angle between the vertical and the rod connecting the bottom mass to the top mass. If we define the generalized coordinate vector q as $q = \begin{bmatrix} \phi_1 & \phi_2 \end{bmatrix}^T$, then the



(a) Height of left end of rod vs. time.

(b) Height of right end of rod vs. time.

(c) Nonimpulsive reaction force exerted by table at left end of rod vs. time.

(d) Nonimpulsive reaction force exerted by table at right end of rod vs. time.

FIG. 3. *Output data from simulation of sliding rod.*

mass matrix $M(q)$ is given by

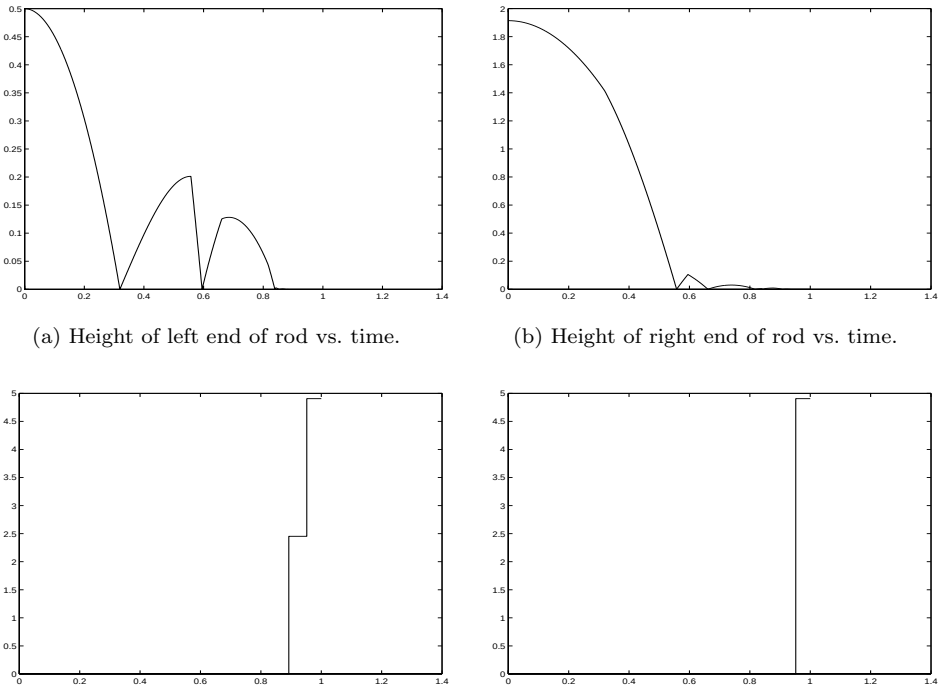
$$M(q) = \begin{bmatrix} (m_1 + m_2)L_1^2 & m_2 L_1 L_2 \cos(q_1 - q_2) \\ m_2 L_1 L_2 \cos(q_1 - q_2) & m_2 L_2^2 \end{bmatrix},$$

and the total force exerted on the system by sources other than contact is given by

$$f(q, \dot{q}, t) = \begin{bmatrix} -m_2 L_1 L_2 \dot{q}_2^2 \sin(q_1 - q_2) - 9.81(m_1 + m_2)L_1 \sin q_1 \\ m_2 L_1 L_2 \dot{q}_1^2 \sin(q_1 - q_2) - 9.81 m_2 L_2 \sin q_2 \end{bmatrix},$$

where the gravitational acceleration constant is 9.81 meters per second per second. In the first double pendulum system (Figure 1(c)), there is an immovable wall of length L_1 extending downward from the ceiling. The bottom mass may move to the left past the wall. However, the top mass is constrained so as to remain to the right of the wall. This can be expressed with the gap function given by $\Psi_{\text{n}}(q) = L_1 \sin q_1$. Note that the constraint $L_1 \sin q_1 \geq 0$ is equivalent to $q_1 \geq 0$ (for $q \in [0, \pi/2]$), but we use $L_1 \sin q_1 \geq 0$ as the constraint because the quantity $L_1 \sin q_1$ is the horizontal position of the top mass. Thus the constraint expresses the condition of being "to the right of the wall."

In the second double pendulum system (Figure 1(d)), we extend the immovable wall downward to a length of $L_1 + L_2$ from the ceiling. Both masses are thus constrained to remain to the right of the wall. This can be expressed with the gap function given by $\Psi_{\text{n}}(q) = \begin{bmatrix} L_1 \sin q_1 & L_1 \sin q_1 + L_2 \sin q_2 \end{bmatrix}^T$.



(a) Height of left end of rod vs. time.

(b) Height of right end of rod vs. time.

(c) Nonimpulsive reaction force exerted by table at left end of rod vs. time.

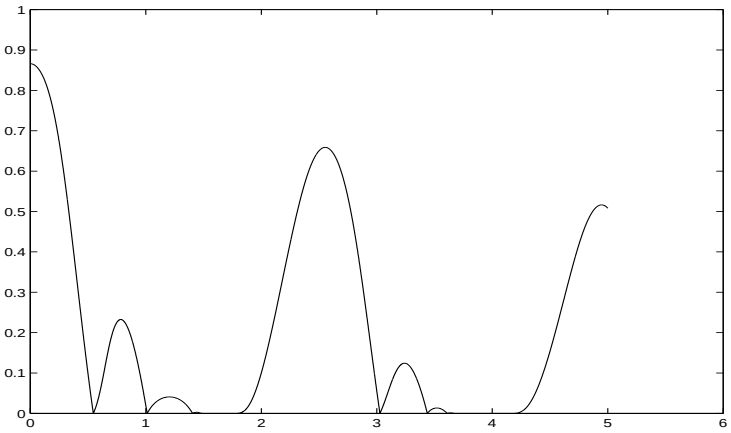
(d) Nonimpulsive reaction force exerted by table at right end of rod vs. time.

FIG. 4. *Output data from simulation of falling rod.*

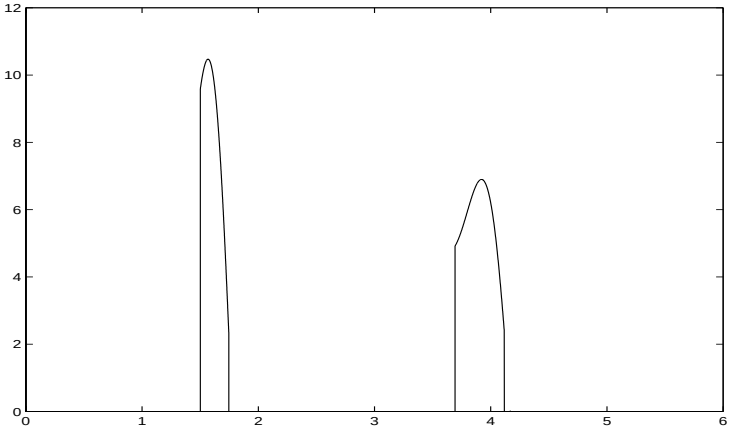
7.3. Two carts. The final two systems that we consider, borrowed from [11, 5], involve two carts (Figure 2). In both systems, the left-most cart has mass m_1 and is connected to an immovable wall via a spring with stiffness coefficient k_1. The right-most cart has mass m_2 and is connected via a spring with stiffness coefficient k_2 to the left-most cart. We define x_1 as the horizontal displacement from equilibrium for the left-most cart. Similarly, we define x_2 as the horizontal displacement from equilibrium for the right-most cart. If we define the generalized coordinate vector q as $q = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$, then the mass matrix $M(q)$ is given by $M(q) = \operatorname{diag}\begin{bmatrix} m_1 & m_2 \end{bmatrix}^T$, and the total force exerted on the system by sources other than contact is given by $f(q, \dot{q}, t) = \begin{bmatrix} -(k_1 + k_2)q_1 + k_2 q_2 & k_2(q_1 - q_2) \end{bmatrix}^T$. Both systems have an immovable wall, constraining the motion of the left-most cart. The first two-cart system has a gap function given by $\Psi_{\mathrm{n}}(q) = q_1$.

In the second two-cart system (Figure 2(b)), the carts are connected via a hook, forcing the two carts to remain within a fixed distance of each other. Note that both carts are constrained simultaneously by the hook. This can be expressed with the gap function given by $\Psi_{\mathrm{n}}(q) = \begin{bmatrix} q_1 & q_1 - q_2 \end{bmatrix}^T$.

7.4. Simulation results. In this subsection, we present the results of our numerical simulation of the six systems described in the previous section. All computations were performed on a dual-cpu Pentium II 350MHz machine with 128 MB of RAM, running Red Hat Linux 7.1, and using version 5.3 of MATLAB$^{\mathrm{TM}}$. We summarize the parameters of our algorithm and models in Tables 1, 2, and 3. In Figures 3, 4, 5, 6, 7, and 8, we plot the gap functions and the nonimpulsive parts of the reaction



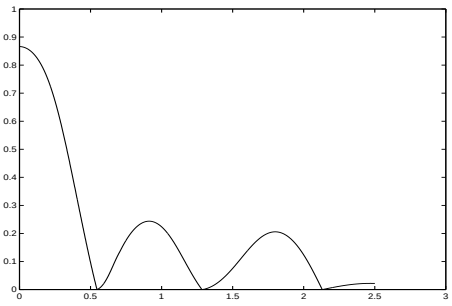
(a) Horizontal position of top pendulum vs. time.



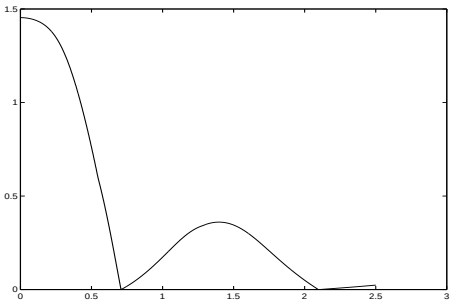
(b) Nonimpulsive reaction force exerted at wall vs. time.

FIG. 5. *Output data from simulation of double pendulum with short wall.*

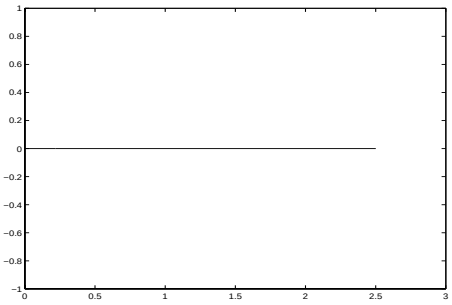
forces resulting from our simulations as functions of time. That is, when plotting the reaction forces, we omit the impulses occurring at impact times. Notice that the original complementarity relationship between the gap and the nonimpulsive reaction force is preserved. In Figure 9, we present the energy data as a function of time for both double pendulum systems. In particular, note that as is predicted by the impact model, the kinetic energies contain jump-discontinuities whenever an impact occurs, but the potential energies remain continuous over the entire time horizon. As a result, note that the total energies of both systems are piecewise constant, monotonically decreasing functions of time with jump-discontinuities whenever an impact occurs. This is consistent with the dissipative nature of our impact model. Finally, in Table 4 we summarize the number of times our algorithm needed to backtrack (Step 5) for each simulation. Notice that the number of backtracks is much larger for the system involving two carts, a wall, and a hook. This is because the coefficient of restitution for the hook is very small, and in this simulation there is a long-sustained contact involving the hook.



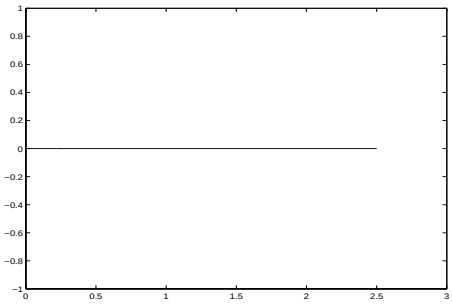
(a) Horizontal position of top pendulum vs. time.



(b) Horizontal position of bottom pendulum vs. time.



(c) Nonimpulsive reaction force exerted at wall vs. time.



(d) Nonimpulsive reaction force exerted at wall vs. time.

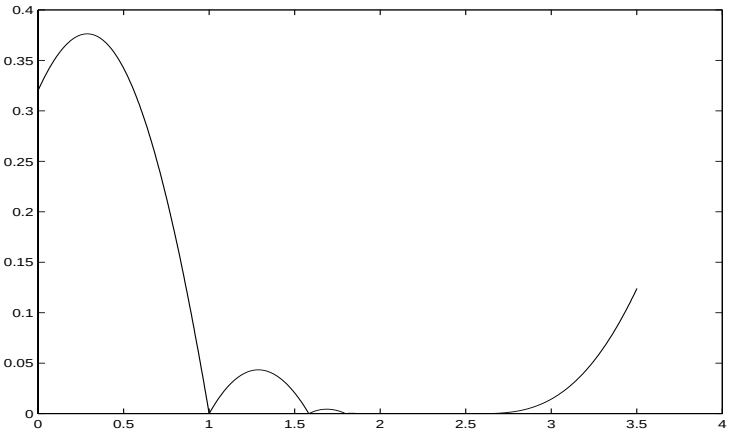
FIG. 6. *Output data from simulation of double pendulum with long wall.*

TABLE 4
Frequency of backtracking.

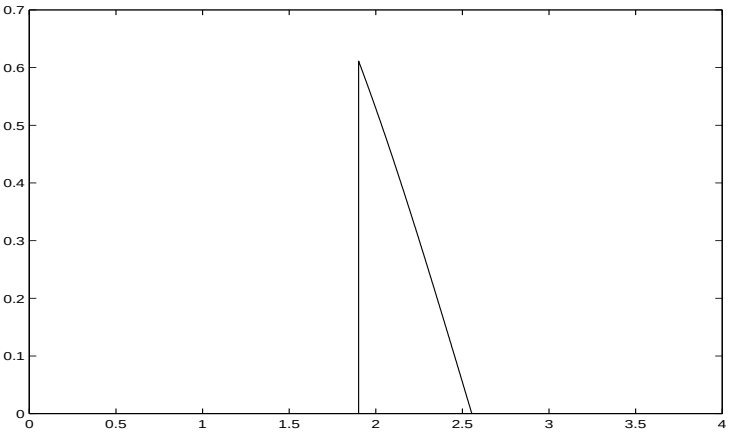
Simulation	Backtracks
Sliding rod	14
Falling rod	17
Double pendulum, short wall	0
Double pendulum, long wall	0
Two carts & wall	2
Two carts, wall, & hook	509

8. Conclusion. In this work we have presented a time-stepping numerical method for solving hybrid complementarity problems arising from frictionless rigid-body mechanical systems. The method consists of the trapezoidal discretization method for classical ODEs in addition to LCP methodology. We introduced a novel impact law in the algorithm to correctly identify the system mode after impact. The algorithm was tested on several systems of rigid bodies in contact. The numerical results obtained were consistent with expected behavior of these systems.

Note that the absence of friction greatly facilitates the proofs of Theorems 5.3 and 5.4. Indeed, in the frictionless case, the LCP (6) is defined by a positive definite matrix. When friction is present (which necessitates the inclusion of tangential forces), we cannot expect such a well-behaved LCP; in fact, the corresponding problem in the frictional case will be a nonlinear complementarity problem under the well-known



(a) Wall gap vs. time.



(b) Nonimpulsive reaction force exerted by wall vs. time.

FIG. 7. *Output data from simulation of two carts and a wall.*

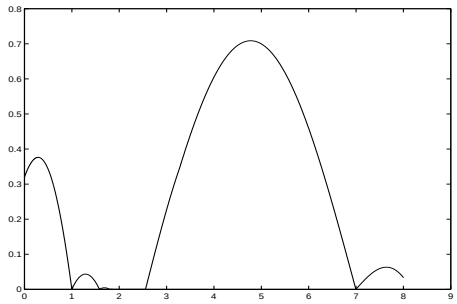
(elliptic) Coulomb friction law. Hence the Lipschitzian property of the force vector as a function of the state variable is no longer as easy to verify as with the frictionless case studied herein. In a subsequent work (see [34]), we plan to extend this methodology to the case which includes elliptic Coulomb friction.

Appendix. Functional LCPs.

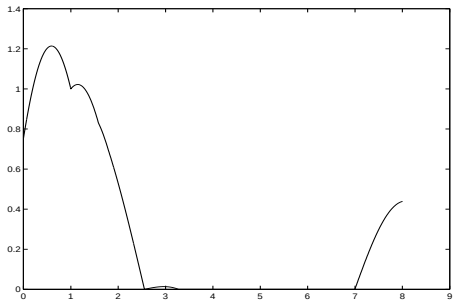
DEFINITION A.1. *Given a function* $A : \mathbb{R}^k \to \mathbb{R}^{m\times m}$ *and a function* $b : \mathbb{R}^k \to \mathbb{R}^m$, *the finite-dimensional FLCP is to find a function* $u : \mathbb{R}^k \to \mathbb{R}^m$ *such that, for all* $q \in \mathcal{D} \subseteq \mathbb{R}^k$, $0 \le w(q) = b(q) + A(q)u(q) \perp u(q) \ge 0$.

DEFINITION A.2. *A function* $A : \mathbb{R}^k \to \mathbb{R}^{m\times m}$ *is said to be uniformly positive definite on* $\mathcal{D} \subseteq \mathbb{R}^l$ *if there exists a positive constant* c *such that* $x^T A(y)x \ge c\,\|x\|^2$ *for all* $y \in \mathcal{D}$ *and* $x \in \mathbb{R}^k$.

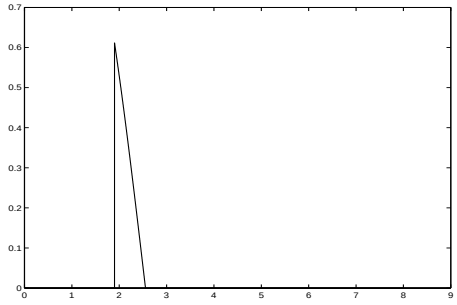
Using Definition A.2, we have the following sufficiency result for the existence and uniqueness of a solution to the finite-dimensional FLCP.



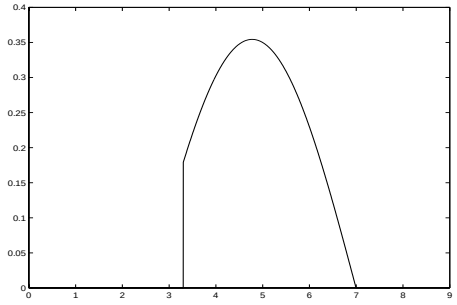
(a) Wall gap vs. time.



(b) Hook gap vs. time.



(c) Nonimpulsive reaction force exerted by wall vs. time.

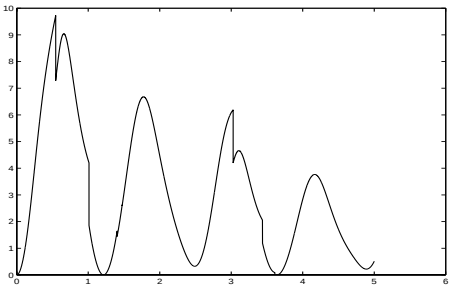


(d) Nonimpulsive reaction force exerted by hook vs. time.

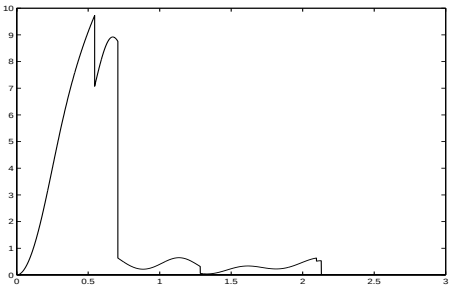
FIG. 8. *Output data from simulation of falling rod.*

THEOREM A.3. *Given any function* $A : \mathbb{R}^k \to \mathbb{R}^{m\times m}$ *that is both uniformly positive definite and Lipschitz on some bounded set* $\mathcal{B} \subseteq \mathbb{R}^k$ *and any function* $b : \mathbb{R}^k \to \mathbb{R}^m$ *that is also Lipschitz on* $\mathcal{B}$*, the finite-dimensional FLCP given by* $0 \le w(q) = b(q) + A(q)u(q) \perp u(q) \ge 0$ *for all* $q \in \mathcal{B}$ *has a unique solution* $u : \mathbb{R}^k \to \mathbb{R}^m$ *that is Lipschitz on* $\mathcal{B}$*.*

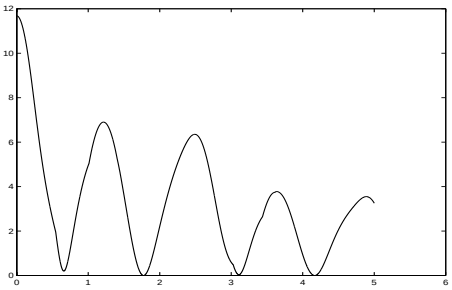
Proof. For any $q, q' \in \mathcal{B}$, consider the two LCPs given by $0 \le w(q) = b(q) + A(q)u(q) \perp u(q) \ge 0$ and $0 \le w(q') = b(q') + A(q')u(q') \perp u(q') \ge 0$. For each $q \in \mathcal{B}$, the matrix $A(q)$ is positive definite, as is the case for each $q' \in \mathcal{B}$. Therefore, by a well-known result from the theory of LCPs (see [8]), there must exist unique vectors $u(q)$ and $u(q')$ satisfying the above two LCPs for every $q, q' \in \mathcal{B}$. All that remains to be shown is the Lipschitz property. Consider $(u(q) - u(q'))^T(w(q) - w(q')) = (u(q) - u(q'))^T(b(q) - b(q') + A(q)u(q) - A(q')u(q'))$ for all $q, q' \in \mathcal{B}$. After algebraic manipulation, we have that $A(q)u(q) - A(q')u(q') = A(q)(u(q) - u(q')) + (A(q) - A(q'))u(q')$. Noting the above complementarity conditions, we have that $(u(q)-u(q'))^T(w(q)-w(q')) \le 0$, so that $(u(q)-u(q'))^T A(q)(u(q)-u(q')) \le -(u(q)-u(q'))^T(b(q) - b(q') + (A(q) - A(q'))u(q'))$. There exists some positive constant c such that $c\|u(q) - u(q')\|^2 \le -(u(q) - u(q'))^T(b(q) - b(q') + (A(q) - A(q'))u(q'))$, since A is uniformly positive definite on $\mathcal{B}$. From the Cauchy–Schwarz inequality, we have $c\|u(q) - u(q')\|^2 \le \|u(q) - u(q')\| \, \|b(q) - b(q') + (A(q) - A(q'))u(q')\|$, so that $\|u(q) - u(q')\| \le \|b(q) - b(q') + (A(q) - A(q'))u(q')\| / c$. Then, from the Lipschitz properties on A and b and the compatibility of the Euclidean matrix and vector norms (see [12]), we obtain $\|u(q) - u(q')\| \le (L_b + L_A \|u(q')\|) / c \cdot \|q - q'\|$, for pos-



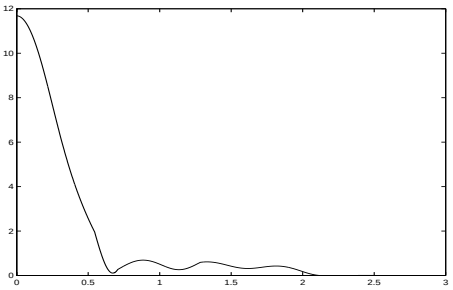
(a) Short wall: kinetic energy vs. time.



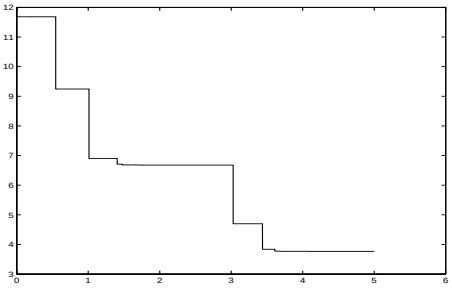
(b) Long wall: kinetic energy vs. time.



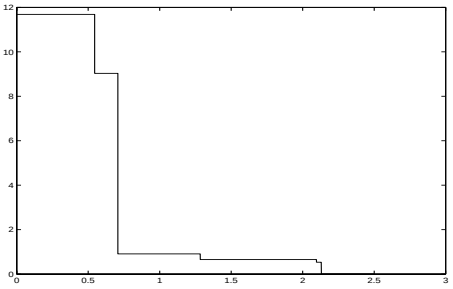
(c) Short wall: potential energy vs. time.



(d) Long wall: potential energy vs. time.



(e) Short wall: total energy vs. time.



(f) Long wall: total energy vs. time.

FIG. 9. *Energy data for both double pendulum simulations.*

itive scalars L_b and L_A. Again, from the LCP above involving q', we have that $u(q')^T A(q')u(q') = -b^T(q')u(q')$, and since A is uniformly positive definite on $\mathcal{B}$, there exists a positive scalar c' such that $c' \|u(q')\|^2 \leq -b^T(q')u(q')$ for all $q' \in \mathcal{B}$. Using the Cauchy–Schwarz inequality, $c' \|u(q')\|^2 \leq \|b(q')\| \|u(q')\|$ for all $q' \in \mathcal{B}$, so that $\|u(q')\| \leq \|b(q')\| / c'$ for all $q' \in \mathcal{B}$. Since b is Lipschitz on the bounded set $\mathcal{B}$, there exists some positive scalar M such that $\|u(q')\| \leq \frac{M}{c'}$ for all $q' \in \mathcal{B}$. Setting $L = (L_b c' + L_A M)/(cc') > 0$, we have that $\|u(q) - u(q')\| \leq L \|q - q'\|$ for all $q, q' \in \mathcal{B}$, and therefore u is Lipschitz on $\mathcal{B}$. □

Acknowledgments. The authors are grateful to Dr. Jeffrey Trinkle of Sandia National Laboratory for some insightful discussion and for his helpful comments on the subject of this paper, particularly pertaining to the impact law. They are also grateful to Dr. Maurice Heemels for sending them a copy of his Ph.D. thesis. The

referees have made some constructive comments, including the detection of a technical mistake in the original version, that helped to improve the paper; the authors thank them too.

REFERENCES

[1] M. Anitescu and F. A. Potra, *Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems*, ASME J. Nonlinear Dynamics, 14 (1997), pp. 231–247.

[2] M. Anitescu, F. A. Potra, and D. E. Stewart, *Time-stepping for three-dimensional rigid-body dynamics*, Comput. Methods Appl. Mech. Engrg., 177 (1999), pp. 183–197.

[3] U. M. Ascher and L. R. Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, SIAM, Philadelphia, PA, 1998.

[4] K. E. Brenan, S. L. Campbell, and L. R. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, Classics Appl. Math. 14, SIAM, Philadelphia, PA, 1996.

[5] B. Brogliato, *Nonsmooth Mechanics*, 2nd ed., Comm. Control Engrg. Ser. 220, Springer, Berlin, 1999.

[6] J. C. Butcher, *The Numerical Analysis of Ordinary Differential Equations: Runge-Kutta and General Linear Methods*, John Wiley and Sons, Chichester, England, 1987.

[7] P. W. Christensen, A. Klarbring, J.-S. Pang, and N. Stromberg, *Formulation and comparison of algorithms for frictional contact problems*, Internat. J. Numer. Methods Engrg., 42 (1998), pp. 145–173.

[8] R. W. Cottle, J.-S. Pang, and R. E. Stone, *The Linear Complementarity Problem*, Academic Press, Boston, 1992.

[9] J. Cronin, *Differential Equations: Introduction and Qualitative Theory*, 2nd ed., Pure Appl. Math. 180, Marcel Dekker, New York, 1994.

[10] M. Heemels, *Linear Complementary Systems, A Study in Hybrid Dynamics*, Ph.D. thesis, Technische Universiteit Eindhoven, Eindhoven, The Netherlands, 1999.

[11] W. P. M. H. Heemels, J. M. Schumacher, and S. Weiland, *Linear complementarity systems*, SIAM J. Appl. Math., 60 (2000), pp. 1234–1269.

[12] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, New York, 1985.

[13] I. Kaneko, *A parametric linear complementarity problem involving derivatives*, Math. Programming, 15 (1978), pp. 146–154.

[14] I. Kaneko, *Complete solutions for a class of elastic-plastic structures*, Comput. Methods Appl. Mech. Engrg., 21 (1980), pp. 193–209.

[15] A. Klarbring, *Contact Problems: Theory, Methods and Applications*, Lecture Notes for the CISM Course, Contact, Friction, Discrete Mechanical Structures and Mathematical Programming, Linkoping University, Sweden, manuscript, 1998.

[16] A. Klarbring and J.-S. Pang, *Existence of solutions to discrete semicoercive frictional contact problems*, SIAM J. Optim., 8 (1998), pp. 414–442.

[17] P. Lötstedt, *Mechanical systems of rigid bodies subject to unilateral constraints*, SIAM J. Appl. Math., 42 (1982), pp. 281–296.

[18] P. Lötstedt, *Time-dependent contact problems in rigid-body mechanics*, Mathematical Programming Study, 17 (1982), pp. 103–110.

[19] P. Lötstedt, *Numerical simulation of time-dependent contact and friction problems in rigid body mechanics*, SIAM J. Sci. Statist. Comput., 5 (1984), pp. 370–393.

[20] M. D. P. M. Marques, *Differential Inclusions in Nonsmooth Mechanical Problems*, Birkhäuser, Basel, 1993.

[21] J. J. Moreau, *Bounded variation in time*, in Topics in Nonsmooth Mechanics, J. J. Moreau, P. D. Panagiotopoulos, and G. Strang, eds., Birkhäuser, Basel, Boston, 1988, pp. 1–74.

[22] J. J. Moreau, *Unilateral contact and dry friction in finite freedom dynamics*, in Nonsmooth Mechanics and Applications, J. J. Moreau and P. D. Panagiotopoulos, eds., CISM Courses and Lectures 302, Springer-Verlag, Vienna, New York, 1988, pp. 1–82.

[23] P. D. Panagiotopoulos, *Inequality Problems in Mechanics and Applications*, Birkhäuser Boston, Cambridge, MA, 1985.

[24] P. D. Panagiotopoulos, *Variational principles for contact problems including impact phenomena*, in Contact Mechanics, Plenum Press, New York, 1995, pp. 431–440.

[25] F. Pfeiffer and C. Glocker, *Multibody Dynamics with Unilateral Contacts*, Wiley Ser. Nonlinear Sci., John Wiley and Sons, New York, 1996.

[26] L. F. Shampine, *Numerical Solution of Ordinary Differential Equations*, Chapman and Hall, London, 1994.
[27] D. Stewart, *High Accuracy Numerical Methods for Ordinary Differential Equations with Discontinuous Right-hand Side*, Ph.D. thesis, University of Queensland, Queensland, Australia, 1990.
[28] D. E. Stewart, *Convergence of a time-stepping scheme for rigid-body dynamics and resolution of Painlevé's problems*, Arch. Ration. Mech. Anal., 145 (1998), pp. 215–260.
[29] D. E. Stewart, *Rigid-body dynamics with friction and impact*, SIAM Rev., 42 (2000), pp. 3–39.
[30] D. E. Stewart and J. C. Trinkle, *Dynamics, friction and complementarity problems*, in Complementarity and Variational Problems: State of the Art, Proc. Appl. Math. 92, SIAM, Philadelphia, PA, 1997, pp. 425–439.
[31] D. E. Stewart and J. C. Trinkle, *An implicit time-stepping scheme for rigid-body dynamics with inelastic collisions and Coulomb friction*, Internat. J. Numer. Methods Engrg., 39 (1996), pp. 2673–2691.
[32] J. C. Trinkle, *private communication*, 1999.
[33] J. C. Trinkle, J.-S. Pang, S. Sudarsky, and G. Lo, *On dynamic multi-rigid-body contact problems with Coulomb friction*, Z. Angew. Math. Mech., 77 (1997), pp. 267–279.
[34] J. A. Tzitzouris, *Numerical Resolution of Frictional Multi-Rigid-Body Systems via Fully Implicit Time-Stepping and Nonlinear Complementarity*, Ph.D. thesis, The Johns Hopkins University, Baltimore, MD, 2001.

SIAM J. OPTIM.
Vol. 12, No. 4, pp. 861–874

SCALED STABILITY IN VARIATIONAL PROBLEMS*

A. B. LEVY†

Abstract. We introduce tools called tolerance functions for the study of a new and practical kind of stability for variational problems on normed vector spaces. Our notion of stability is distinguished by the fact that it explicitly addresses distinctions of scale. To support the stability analysis, we study the continuity and differentiability properties of tolerance functions, paying particular attention to comparisons between tolerance functions and the set-valued mappings that are used to encode variational problems. We end with a discussion of how tolerance functions can be used to analyze the convergence of numerical optimization procedures.

Key words. stability analysis, parametric optimization, numerical optimization

AMS subject classification. 90C31

PII. S1052623400376895

1. Introduction.

Previous theories of stability analysis for variational problems have largely ignored issues of scale. As an illustration, consider the minimization over $x \in \mathbb{R}$ of the parameterized function

$$f_w(x) := \begin{cases} x^4/4 - wx & \text{if } |x| < 1, \\ -3w^{\frac{4}{3}}/4 & \text{if } |x| = 1, \\ \infty & \text{otherwise.} \end{cases}$$

For each parameter value $w \in \mathbb{R}$, the optimal value is $-3/4w^{\frac{4}{3}}$, which is achieved at different points depending on the parameter: If $|w| \leq 1$, then the trio of points $\{1, w^{\frac{1}{3}}, -1\}$ all are optimal solutions, whereas if $|w| > 1$, then only the two points $\{1, -1\}$ are optimal solutions. We can encode this information compactly by defining a set-valued "solution" mapping $S : \mathbb{R} \rightrightarrows \mathbb{R}$ as follows:

$$S(w) = \begin{cases} \{1, w^{\frac{1}{3}}, -1\} & \text{if } -1 \leq w \leq 1, \\ \{1, -1\} & \text{otherwise,} \end{cases} \tag{1}$$

which is graphed in Figure 1. If the point $\bar{w} = 0$ is the base parameter and $\bar{x} = 0$

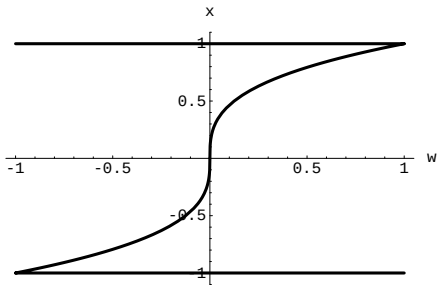


FIG. 1. *Solution mapping.*

*Received by the editors August 17, 2000; accepted for publication (in revised form) September 25, 2001; published electronically March 13, 2002.
http://www.siam.org/journals/siopt/12-4/37689.html
†Department of Mathematics, Bowdoin College, Brunswick, ME 04011 (alevy@bowdoin.edu).

is the base solution, then this problem would be characterized as unstable via all of the various current means of classifying stability. This is because the infinite slope of the graph of $w^{\frac{1}{3}}$ signals apparently significant changes in the solutions when the parameter is perturbed near $\bar{w} = 0$. However, if the scale of interest for the problem is on the order of one unit, then this problem is actually stable, since on that scale there are no significant changes in the solutions when the parameters are perturbed near $\bar{w} = 0$. In fact, if solutions are only needed within any fixed tolerance level $\epsilon > 0$, this problem is stable because changes to the solutions within this tolerance can be bounded above by a linear function of the parameter. In this paper, we introduce a new way of classifying stability where distinctions of scale are inherent.

The key to our study of "scaled stability" is a new and relatively simple concept which we call a tolerance function. In addition to revealing a new and practical concept of stability, tolerance functions convert variational problems of every kind into ones that automatically exhibit uniqueness. Tolerance functions are defined in terms of a specified *target set* $\bar{X} \subseteq \mathcal{X}$, as well as a specified *focus set* $X \subseteq \mathcal{X}$. For any normed vector spaces $\mathcal{X}$ and $\mathcal{W}$, we define the *tolerance function* $t_{\bar{X},X} : \mathcal{W} \to \mathbb{R} \cup \{\pm\infty\}$ associated with a set-valued mapping $S : \mathcal{W} \rightrightarrows \mathcal{X}$ by

$$t_{\bar{X},X}(w) := \sup_{x \in S(w) \cap X} \operatorname{dist}(x, \bar{X}),$$

where $\operatorname{dist}(x, \bar{X})$ is the distance function associated with the target set $\bar{X}$:

$$\operatorname{dist}(x, \bar{X}) := \inf_{\bar{x} \in \bar{X}} \| x - \bar{x} \| .$$

When either $S(w) \cap X$ or the target set $\bar{X}$ is empty, we adopt the convention that $t_{\bar{X},X}(w)$ is equal to negative infinity. In the special case in which the target set is a singleton $\{\bar{x}\}$, we call it the *target value* and write $t_{\bar{x},X}$ for the tolerance function. For the example (1) with target set $\bar{X} = [-\epsilon, \epsilon]$ for $\epsilon < 1$ and any focus set X containing the interval $[-1, 1]$, the tolerance function is constant $t_{[-\epsilon,\epsilon],X}(w) \equiv 1 - \epsilon$. This function is stable by any definition, and so it successfully identifies the scaled stability of this problem. On the other hand, for any focus set $X = [-c, c]$ for $c < 1$ and target value $\bar{x} = 0$, the corresponding tolerance function is

$$t_{0,[-c,c]}(w) = \begin{cases} |w^{\frac{1}{3}}| & \text{for } |w| \leq c^3, \\ c & \text{otherwise.} \end{cases}$$

The graph of this tolerance function (for $c = 0.5$) is shown in Figure 2, where it can be seen to have the same kind of stability problem at the origin as the function $w \mapsto w^{\frac{1}{3}}$. Another layer of scaling can be explored by considering the same focus set $X = [-c, c]$ but now with the target set $\bar{X} = [-\epsilon, \epsilon]$ for $0 < \epsilon < c$. The resulting tolerance function is

$$t_{[-\epsilon,\epsilon],[-c,c]}(w) = \begin{cases} 0 & \text{for } |w| \leq \epsilon^3, \\ |w^{\frac{1}{3}}| - \epsilon & \text{for } \epsilon^3 \leq |w| \leq c^3, \\ c - \epsilon & \text{otherwise,} \end{cases}$$

which is graphed (for $c = 0.5$ and $\epsilon = 0.25$) in Figure 3. There is a flat slope at the origin for any such tolerance function, which illustrates how the underlying optimization problem is inherently stable on all but the infinitesimally small scale. Through this example, we see that different scales of a problem can be studied directly

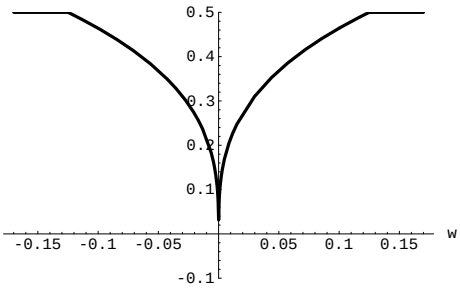


FIG. 2. *Tolerance function for* $c = 0.5$.

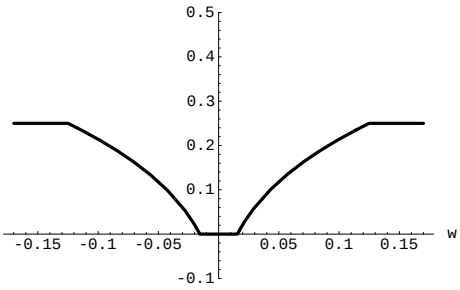


FIG. 3. *Tolerance function for* $c = 0.5$ *and* $\epsilon = 0.25$.

with tolerance functions, and that different stability information can naturally reside on different scales. We also see that tolerance functions do not necessarily retain the full variety of information about a problem; they simply measure the worst-case distance to the target set. Since the central issue in stability analysis is whether the worst-case scenario is acceptable or not, the benefits of a simplification of the variational problem come here without compromising the stability analysis.

In the next section we expand our introduction of tolerance functions and then follow that with a section describing their variational properties (continuity and differentiability). We focus particular attention on relationships between the variational properties of tolerance functions and the variational properties of the set-valued mappings S that are used to encode variational problems as above. These comparisons are important because generalized continuity and differentiability properties of set-valued mappings have until now been essentially the only means of dealing with stability issues in variational problems that do not exhibit existence and uniqueness. Through these comparisons, we show that existing stability results for variational problems, whether exhibiting existence and uniqueness or not, are essentially captured by an analysis based on tolerance functions. In tandem with this theme, we emphasize that there are important gains associated with the shift to tolerance functions. These gains include more practical standards for stability that recognize scale, as well as a simplified analysis resulting from the mathematical structure of tolerance functions. This latter issue is particularly important in dynamic optimization, where the mathematical complexities of infinite-dimensional spaces can cause difficulties for other approaches. In the final section, we illustrate how tolerance functions can be used to analyze the convergence of numerical optimization procedures.

2. Tolerance functions. In the introduction, we saw an example of a tolerance function associated with a simple finite-dimensional parametric optimization problem. One nice feature of tolerance functions is that much of their structure is similar in finite- or infinite-dimensional settings (e.g., in either case they map into the extended real numbers). As an infinite-dimensional example, we consider the following parameterized calculus of variations example, where the parameter w is a real number:

$$\min \int_0^1 \left(\dot{x}(t)^2 - w\right)^2 \mathrm{d}t \text{ over absolutely continuous } x \text{ with } x(0) = 0 \text{ and } x(1) = 1. \tag{2}$$

When $w \leq 1$, the unique optimal solution is $x(t) = t$, but when $w > 1$, there are infinitely many optimal solutions satisfying $\dot{x}(t) = \pm\sqrt{w}$ for almost every t. For the set $X := \{x : |\dot{x}(t)| \leq \sqrt{2}\}$, the intersection with $S(w)$ would be given by

$$S(w) \cap X = \begin{cases} x(t) = t & \text{if } w \leq 1, \\ \text{any feasible } x \text{ with } \dot{x}(t) = \pm\sqrt{w} \text{ for almost every } t & \text{if } 2 \geq w > 1, \\ \emptyset & \text{if } w > 2. \end{cases}$$

If the target value is $\bar{x}(t) = t$ and the norm is defined by

$$\| x \| := |x(0)| + \int_0^1 |\dot{x}(t)| \mathrm{d}t,$$

then the tolerance function for this example is given by

$$t_{\bar{x},X}(w) = \begin{cases} 0 & \text{if } w \leq 1, \\ \sqrt{w} - \frac{1}{w} & \text{if } 2 \geq w > 1, \\ -\infty & \text{if } w > 2, \end{cases}$$

which is graphed in Figure 4. Notice that the tolerance function in this case takes real values to extended real values, which is exactly the same kind of mathematical object as the tolerance function associated with the finite-dimensional example (1). This consistency across the finite-dimensional and infinite-dimensional settings is one distinctive and useful aspect of our approach using tolerance functions. We develop the theory here in the more general, infinite-dimensional setting from the beginning, instead of following the usual model of finite-dimensional results first, followed by infinite-dimensional extensions. Because of the tolerance function's consistency in both settings, we somewhat avoid the experience under the usual model where the stability analysis is much more complicated when extended to infinite dimensions.

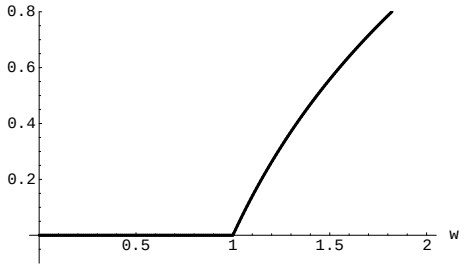


FIG. 4. *Tolerance function for an infinite-dimensional example.*

Of course for the example (2), the constraints on x encoded in the focus set X could easily be absorbed into the original problem. However, the role played in the theory by the focus set X is very different from that played by the set-valued mapping S, and thus it is useful to include both objects to yield the most flexible analysis.

Notice that uniqueness is never an issue for tolerance functions, since they can have at most one value even when the underlying variational problem does not exhibit uniqueness. Moreover, in the case in which the underlying variational problem does exhibit uniqueness (i.e., when the set $S(w) \cap X$ is a singleton), the tolerance function simply measures the distance from the single element of $S(w) \cap X$ to the target set. In general, though, the tolerance function records the greatest distance from the target set among candidates in the set $S(w) \cap X$ and ignores the rest of the structure of the set. This "worst-case" information about the variational problem is precisely the information that is most relevant for resolving questions of stability. The terminology for this new object comes from the fact that it can be used to decide whether a given tolerance for closeness to the target is achieved.

3. Variational properties of tolerance functions. Stability analyses of variational problems are most useful when they analyze stability without actually solving the problems. This is because the point of these analyses is often either to avoid having to solve the problems (e.g., because they are costly to solve), or to determine the extent to which computed solutions can be trusted. Tolerance functions automatically reduce the need for solving variational problems by focusing attention directly on information related to stability issues and by ignoring other aspects of the problems that might normally be part of the analysis. In this section, we will study the continuity and differentiability properties of tolerance functions. Moreover, we will pay particular attention to estimates for derivatives that can be computed from the original data only, since these allow a stability analysis without solving the underlying variational problems at all. A general theme of this section is that nothing important for stability analysis is lost by studying tolerance functions in place of the set-valued mappings representing the variational problems.

3.1. Continuity. There are many useful notions of continuity for nonsmooth functions, including lower semicontinuity, upper semicontinuity, calmness, and Lipschitz continuity. To support the claim that nothing important for stability analysis is lost with tolerance functions, in this section we seek relationships between the continuity properties of tolerance functions and their underlying set-valued mappings. In particular, we show that typical generalized notions of continuity for the set-valued mappings imply analogous notions of continuity for tolerance functions.

The most basic of the continuity notions for nonsmooth functions are lower and upper semicontinuity. Recall that a function $t : \mathcal{W} \to \mathbb{R} \cup \{\pm\infty\}$ is *lower semicontinuous at* $\bar{w}$ if

$$\liminf_{w_n \to \bar{w}} t(w_n) \geq t(\bar{w}),$$

is *upper semicontinuous at* $\bar{w}$ if

$$\limsup_{w_n \to \bar{w}} t(w_n) \leq t(\bar{w}),$$

and is *continuous at* $\bar{w}$ if it is both lower and upper semicontinuous at $\bar{w}$. Notice that for tolerance functions the two different notions of semicontinuity give very different information about the underlying set-valued mappings. For instance, if $S(\bar{w}) \cap X$ is

nonempty (so $t_{\bar{X},X}(\bar{w})$ does not equal negative infinity), then the lower semicontinuity of $t_{\bar{X},X}$ at $\bar{w}$ implies that $S(w) \cap X$ is nonempty for all w near $\bar{w}$. Under the same assumption that $S(\bar{w}) \cap X$ is nonempty, the upper semicontinuity at $\bar{w}$ of $t_{\bar{X},X}$ implies that whenever $S(w) \cap X$ is nonempty and w is close enough to $\bar{w}$, the worst-case distance from $\bar{X}$ of the points in $S(w) \cap X$ can be usefully estimated by the worst-case distance from $\bar{X}$ of the points in the base set $S(\bar{w}) \cap X$. In this situation, if the base set's worst-case distance is less than the tolerance level, then nearby sets also have this property.

For this paper, the appropriate generalized notions of semicontinuity for set-valued mappings are inner and outer semicontinuity. Since the set-valued mappings can be from one infinite-dimensional space to another, we will need to consider different kinds of convergence. Here and throughout the paper, an unadorned arrow "$\to$" indicates strong convergence. A set-valued mapping $S : \mathcal{W} \rightrightarrows \mathcal{X}$ is *s-s inner semicontinuous at* $\bar{w}$ if for every element $\tilde{x} \in S(\bar{w})$ and every sequence $w_n \to \bar{w}$ there exists a sequence $x_n \to \tilde{x}$, with $x_n \in S(w_n)$. On the other hand, a set-valued mapping is *s-w* outer semicontinuous at* $\bar{w}$ if for every pair of sequences $w_n \to \bar{w}$ and $x_n \to^{w^*} \tilde{x}$, with $x_n \in S(w_n)$, the limit pair satisfies $\tilde{x} \in S(\bar{w})$. Of course there are analogous notions of semicontinuity for set-valued mappings that use different combinations of convergence, but the two above are particularly appropriate for the comparison to continuity of tolerance functions.

THEOREM 3.1. *For any set-valued mapping $S : \mathcal{W} \rightrightarrows \mathcal{X}$, any target set $\bar{X} \subseteq \mathcal{X}$, and any focus set $X \subseteq \mathcal{X}$,*

(i) *if the set-valued mapping $w \mapsto S(w) \cap X$ is s-s inner semicontinuous at $\bar{w}$, then the associated tolerance function $t_{\bar{X},X}$ is lower semicontinuous at $\bar{w}$;*

(ii) *if X is bounded and the set-valued mapping $w \mapsto S(w) \cap X$ is s-w* outer semicontinuous at $\bar{w}$, then the associated tolerance function $t_{\bar{X},X}$ is upper semicontinuous at $\bar{w}$;*

(iii) *if X is bounded and the set-valued mapping $w \mapsto S(w) \cap X$ is both s-s inner semicontinuous and s-w* outer semicontinuous at $\bar{w}$, then the associated tolerance function $t_{\bar{X},X}$ is continuous at $\bar{w}$.*

Proof of (i). If the set $S(\bar{w}) \cap X$ is empty, then the tolerance function satisfies $t_{\bar{X},X}(\bar{w}) = -\infty$ and is trivially lower semicontinuous at $\bar{w}$. Otherwise, consider any point $\tilde{x} \in S(\bar{w}) \cap X$ and any sequence $w_n \to \bar{w}$ in $\mathcal{W}$. By the assumption of inner semicontinuity, there is some sequence of points $x_n \in S(w_n) \cap X$ with $x_n \to \tilde{x}$. Then by the definition of the tolerance function, we know that $t_{\bar{X},X}(w_n) \geq \operatorname{dist}(x_n, \bar{X})$, and thus the continuity of the distance function implies the estimate

$$\liminf t_{\bar{X},X}(w_n) \geq \operatorname{dist}(\tilde{x}, \bar{X}). \tag{3}$$

Since inequality (3) holds for all $\tilde{x} \in S(\bar{w}) \cap X$, we conclude from the definition of $t_{\bar{X},X}(\bar{w})$ that $t_{\bar{X},X}$ is lower semicontinuous at $\bar{w}$.

Proof of (ii). Consider any sequence $w_n \to \bar{w}$. If $S(w_n) \cap X$ is empty, then $t_{\bar{X},X}(w_n) = -\infty$, and thus we consider only the elements of the sequence for which $S(w_n) \cap X$ is not empty. For such sets and any positive number ϵ, since X is bounded we can find elements $x_n \in S(w_n) \cap X$ with $\operatorname{dist}(x_n, \bar{X}) + \epsilon > t_{\bar{X},X}(w_n)$. From this estimate we know that

$$\limsup \operatorname{dist}(x_n, \bar{X}) + \epsilon \geq \limsup t_{\bar{X},X}(w_n). \tag{4}$$

Since X is bounded, we know that every subsequence of $\{x_n\}$ has a subsubsequence that w^* converges to some element, say $\tilde{x}$. By the assumed s-w* outer semicontinuity,

we know that each limit point $\tilde{x}$ must be in the set $S(\bar{w}) \cap X$, from which we conclude that $t_{\bar{X},X}(\bar{w}) \geq \text{dist}(\tilde{x}, \bar{X})$. This estimate combined with the estimate (4) yields the desired upper semicontinuity of $t_{\bar{X},X}$, since the distance function is continuous and $\epsilon > 0$ is arbitrary.

Proof of (iii). This follows immediately from (i) and (ii). □

The implications in Theorem 3.1 can not be reversed in general, as can be seen by considering the set-valued mapping $S : \mathbb{R} \rightrightarrows \mathbb{R}$ defined by

$$S(w) = \begin{cases} \{-1, 1\} & \text{if } w \neq 0, \\ \{0, 1\} & \text{if } w = 0. \end{cases}$$

If the target value is $\bar{x} = 0$, then the tolerance function associated with any set X containing $[-1, 1]$ is constant $t_{0,X}(w) \equiv 1$ and thus trivially continuous at 0. However, for any of the same sets X, the mapping $w \mapsto S(w) \cap X$ is neither inner nor outer semicontinuous at 0.

For set-valued mappings, there are several different notions generalizing Lipschitz continuity. One of these, appropriately called "Lipschitz continuity," implies the local Lipschitz continuity of $t_{\bar{X},X}$ for any target set $\bar{X}$. A set-valued mapping S is *Lipschitz continuous on* W if there exists a constant $L \geq 0$ such that

$$S(w) \subseteq S(w') + L \parallel w - w' \parallel \mathbb{B} \quad \text{for all } w, w' \in W,$$

where $\mathbb{B}$ is the unit ball in $\mathcal{X}$.

THEOREM 3.2. *If the set-valued mapping $w \mapsto S(w) \cap X$ is Lipschitz continuous on $W \subseteq \mathcal{W}$ with modulus $L \geq 0$, then for any target set $\bar{X} \subseteq \mathcal{X}$ the associated tolerance function is Lipschitz continuous on W:*

$$|t_{\bar{X},X}(w) - t_{\bar{X},X}(w')| \leq L \parallel w - w' \parallel \quad \textit{for all } w, w' \in W.$$

Proof. The supremum over a set contained in a second set is less than the supremum over the second set, so we conclude immediately from the definitions that

$$t_{\bar{X},X}(w) \leq t_{\bar{X},X}(w') + L \parallel w - w' \parallel$$

for all w and w' in W. Since the roles of w and w' can be reversed in this inequality, the local Lipschitz continuity of $t_{\bar{X},X}$ follows. □

The Lipschitz continuity assumption on $w \mapsto S(w) \cap X$ is closely related to the well-studied notion of "pseudo-Lipschitz continuity" for set-valued mappings S, which was first identified in [1]. For the pseudo-Lipschitz continuity property, however, the set X only appears on the left-hand side of the inclusion.

A final popular notion of continuity for set-valued mappings that we wish to explore in this paper is called "calmness" (it was originally called upper-Lipschitz continuity in [6]). Recall that a set-valued mapping $S : \mathcal{W} \rightrightarrows \mathcal{X}$ is *calm at* $\bar{w}$ if the set $S(\bar{w})$ is nonempty and there exists a neighborhood W of $\bar{w}$ in $\mathcal{W}$ and a constant $L \geq 0$ such that

$$S(w) \subseteq S(\bar{w}) + L \parallel w - w' \parallel \mathbb{B} \quad \text{for all } w \in W.$$

This notion of continuity for set-valued mappings is the one among all of those mentioned so far that most closely reflects the concepts behind tolerance functions. This is because calmness can be viewed as essentially a "worst-case" continuity property,

where the only feature of the set $S(w)$ that is significant is its containment in a Lipschitz perturbation of the base set. Contrast this, for instance, to the situation with Lipschitz continuity on W, where for any $w \in W$ the set $S(w)$ plays a role on both sides of the inclusion. The analogue to calmness for single-valued functions is called "calmness from above" [7]. Recall that $t : \mathcal{W} \to \mathbb{R} \cup \{\pm\infty\}$ is *calm at $\bar{w}$ from above* if $t(\bar{w})$ is finite and there exists a neighborhood W of $\bar{w}$ in $\mathcal{W}$ and a constant $L \geq 0$ for which

$$t(w) \leq t(\bar{w}) + L \parallel w - \bar{w} \parallel \qquad \text{for all } w \in W.$$

This notion of continuity is particularly useful for stability analyses via tolerance functions, since it allows worst-case distances for nearby parameters to be estimated simply by the worst-case distance for the base parameter.

THEOREM 3.3. *If the set-valued mapping $w \mapsto S(w) \cap X$ is calm at $\bar{w} \in \mathcal{W}$, then for any target set $\bar{X} \subseteq \mathcal{X}$, the associated tolerance function $t_{\bar{X},X}$ is calm at $\bar{w}$ from above (with the same constant $L \geq 0$ and neighborhood $W \subseteq \mathcal{W}$ of $\bar{w}$).*

Proof. This follows immediately from the definitions, since the supremum over a set contained in a second set is less than the supremum over the second set. □

All of the comparisons in this section are important because generalized continuity properties of set-valued mappings have until now been essentially the only means of dealing with stability issues in variational problems that do not necessarily exhibit existence and uniqueness (see [5]). Of course, these generalized continuity properties reduce to more traditional notions of continuity when the set-valued mappings are actually single-valued, so they also cover stability issues in the traditional setting. It is clear from the implications established above that a tolerance function-based analysis classifies as stable any problem that is classified as stable under the current theory based on the variational properties of set-valued mappings. Moreover, as we discussed in the introduction and at the beginning of this section, our approach to stability analysis using tolerance functions reveals a new and practical kind of stability in many problems that are classified as unstable by the current theory. For instance, notice that for the example (1) the intersected mapping $w \mapsto S(w) \cap X$ is neither Lipschitz continuous on W nor calm at 0 for any neighborhoods W and X of 0. However, we already saw in the introduction that tolerance functions associated with this same problem were constant and thus trivially Lipschitz continuous and calm from above.

3.2. Differentiability. In contrast to the situation in the previous subsection, where many of the different notions of continuity are useful for analyzing tolerance functions, we focus on only one of the many popular notions of differentiability for nonsmooth functions that is particularly well suited for studying tolerance functions. For a function $t : \mathcal{W} \to \mathbb{R} \cup \{\pm\infty\}$ and a point $\bar{w}$, where t is finite, the upper subderivative function $d^+t(\bar{w}) : \mathcal{W} \to \mathbb{R} \cup \{\pm\infty\}$ is defined by

$$d^+t(\bar{w})(w) := \limsup_{\substack{\tau \downarrow 0 \\ w' \to w}} \frac{t(\bar{w} + \tau w') - t(\bar{w})}{\tau}.$$

One property of the upper subderivative that we will use is its positive homogeneity:

$$d^+t(\bar{w})(a\,w) = a\,d^+t(\bar{w})(w) \quad \text{for all positive scalars } a > 0,$$

which follows immediately from the definition. The upper subderivative is particularly useful for a stability analysis based on tolerance functions because of the following result.

PROPOSITION 3.1. *For any function $t : \mathcal{W} \to \mathbb{R} \cup \{\pm\infty\}$ and a point $\bar{w}$ where t is finite, the estimate holds that*

$$t(w) \leq t(\bar{w}) + d^+t(\bar{w})(w - \bar{w}) + o(\| w - \bar{w} \|).$$

Proof. By the definition of the upper subderivative, we have

$$d^+t(\bar{w}) \left(\frac{w - \bar{w}}{\| w - \bar{w} \|} \right) = \limsup_{\substack{\tau \downarrow 0 \\ w' \to \frac{w-\bar{w}}{\|w-\bar{w}\|}}} \frac{t(\bar{w} + \tau w') - t(\bar{w})}{\tau}. \tag{5}$$

Since one possible choice of sequence in the above limit superior is $w' \equiv \frac{w-\bar{w}}{\|w-\bar{w}\|}$, we conclude immediately from (5) that

$$d^+t(\bar{w}) \left(\frac{w - \bar{w}}{\| w - \bar{w} \|} \right) \geq \limsup_{\tau \downarrow 0} \frac{t\left(\bar{w} + \tau \frac{w-\bar{w}}{\|w-\bar{w}\|}\right) - t(\bar{w})}{\tau},$$

and from this (thinking of $\tau = \| w - \bar{w} \|$) we conclude that

$$d^+t(\bar{w}) \left(\frac{w - \bar{w}}{\| w - \bar{w} \|} \right) + O(\| w - \bar{w} \|) \geq \frac{t(w) - t(\bar{w})}{\| w - \bar{w} \|}.$$

Multiplying both sides by the term $\| w - \bar{w} \|$ gives the claimed estimate, since the upper subderivative is positively homogeneous. ∎

This proposition gives an upper estimate in terms only of data at the base point for the value of the function at any other parameter. For tolerance functions which already measure worst-case distances, an upper estimate like this is particularly attractive for stability analyses.

The upper subderivative is connected to the calmness property via the following result.

PROPOSITION 3.2. *If the function $t : \mathcal{W} \to \mathbb{R} \cup \{\pm\infty\}$ with $t(\bar{w})$ finite is calm at $\bar{w}$ from above, then for the same constant $L \geq 0$ we have the estimate $d^+t(\bar{w})(w) \leq L \| w \|$, with equality when $w = 0$.*

Proof. For any sequences $\tau \downarrow 0$ and $w' \to w$, the calmness assumption ensures the estimate

$$\frac{t(\bar{w} + \tau w') - t(\bar{w})}{\tau} \leq L \| w' \|.$$

From this, our estimate follows immediately, with equality when $w = 0$, since in that case the constant sequence $w' \equiv 0$ is one alternative for the limit superior in the definition of the upper subderivative. ∎

When the space $\mathcal{W}$ is finite-dimensional, the implication in Proposition 3.2 can be reversed.

PROPOSITION 3.3 (see [7], Proposition 8.32). *The function $t : \mathbb{R}^n \to \mathbb{R} \cup \{\pm\infty\}$ with $t(\bar{w})$ finite is calm at $\bar{w}$ from above if and only if $d^+t(\bar{w})(0) = 0$.*

One special case for applications that we will consider is when the target set equals the base set $S(\bar{w}) \cap X$, in which case the tolerance function satisfies $t_{S(\bar{w})\cap X, X}(\bar{w}) = 0$. When this base set is a singleton $\{\bar{x}\}$, we can estimate the upper subderivative of the tolerance function in terms of the outer graphical derivative DS of $S : \mathcal{W} \rightrightarrows \mathcal{X}$. Recall that the outer graphical derivative of S at $\bar{w}$ for $\bar{x}$ is defined by

$$DS(\bar{w}|\bar{x})(w) := \{x : \exists w' \to w,\, x' \to x, \text{ and } \tau \downarrow 0 \text{ with } \bar{x} + \tau x' \in S(\bar{w} + \tau w')\}.$$

PROPOSITION 3.4. *If there exists a set $X \subseteq \mathcal{X}$ such that $S(\bar{w}) \cap X = \{\bar{x}\}$, then the upper subderivative of the associated tolerance function $t_{\bar{x},X}$ satisfies*

$$d^+ t_{\bar{x},X}(\bar{w})(w) \geq \sup_{x \in DS(\bar{w}|\bar{x})(w)} \| x \| . \tag{6}$$

Proof. For any element $x \in DS(\bar{w}|\bar{x})(w)$, there exist sequences $w' \to w$, $x' \to x$, and $\tau \downarrow 0$ satisfying $\bar{x} + \tau x' \in S(\bar{w} + \tau w')$. From this inclusion, it is clear that the tolerance function satisfies

$$\frac{t_{\bar{x},X}(\bar{w} + \tau w')}{\tau} \geq \frac{\| \tau x' \|}{\tau} = \| x' \| . \tag{7}$$

Now $x' \to x$ and $t_{\bar{x},X}(\bar{w}) = 0$, so we can use (7) to obtain the estimate

$$d^+ t_{\bar{x},X}(\bar{w})(w) \geq \| x \| .$$

Since x is an arbitrary element of the set $DS(\bar{w}|\bar{x})(w)$, we conclude the claimed estimate for this case. □

The following corollary shows that for tolerance functions, the characterization in Proposition 3.3 can hold even when the domain space $\mathcal{W}$ is not finite-dimensional. Notice that the range space $\mathcal{X}$ has to be finite-dimensional in this case.

COROLLARY 3.1. *If there exists a set $X \subseteq \mathbb{R}^n$ such that $S(\bar{w}) \cap X = \{\bar{x}\}$, then there exists a focus set $\tilde{X} \subseteq \mathbb{R}^n$ such that the associated tolerance function $t_{\bar{x},\tilde{X}}$ is calm from above at $\bar{w}$ if and only if $d^+ t_{\bar{x},\tilde{X}}(\bar{w})(0) = 0$.*

Proof. The implication (i) ⇒ (ii) is immediate from Proposition 3.2.

To show (ii) ⇒ (i), we notice that the estimate (6) from Proposition 3.4 implies $DS(\bar{w}|\bar{x})(0) = \{0\}$. We then appeal to [4, Proposition 2.1], which gives the existence of a neighborhood $\tilde{X} \subseteq \mathbb{R}^n$ of $\bar{x}$ such that the mapping $w \mapsto S(w) \cap \tilde{X}$ is calm at $\bar{w}$. From Theorem 3.3, we then know that the tolerance function $t_{\bar{x},\tilde{X}}$ is calm from above at 0. □

Estimate (6) in Proposition 3.3 nicely parallels the formula for the tolerance function itself, as only the worst-case elements of the outer graphical derivative are important in the formula for the upper subderivative. For practical purposes, this estimate means that the upper subderivative of the tolerance function can be estimated without necessarily computing the entire outer graphical derivative $DS(\bar{w}|\bar{x})$, which is likely to be set-valued. The inequality in (6) is strict in general, as can be seen by considering, for example, the set-valued mapping

$$S(w) := \big\{(1, 1 + |w|] \cup [-|w|, |w|]\big\}$$

and its associated tolerance function

$$t_{0,\mathbb{R}}(w) = \begin{cases} 1 + |w| & \text{if } w \neq 0, \\ 0 & \text{if } w = 0. \end{cases}$$

In this case, the outer graphical derivative of S satisfies $DS(0|0)(w) = [-|w|, |w|]$, and thus the supremum on the right-hand side of (6) is $|w|$, but the upper subderivative on the left-hand side is always infinite. This difference reflects the fundamentally different treatment of scales by the objects underlying the two different sides of estimate (6). Outer graphical derivatives say something only about the set-valued mapping's behavior on an infinitesimally small scale near the base pair $(\bar{w}, \bar{x})$, whereas the fixed scale chosen for the tolerance function can change its properties dramatically. For instance,

if instead of the "global-scale" tolerance function (where $X = \mathbb{R}$) in the example above we used the tolerance function associated with any focus set $X \subseteq (-\infty, 1]$, we would have a tolerance function $t_{0,X}(w) = |w|$ whose upper subderivative at 0 agrees with the right-hand side of (6). It turns out that this agreement is not accidental: Whenever the upper subderivative on the left-hand side is finite and the range space $\mathcal{X}$ is finite-dimensional, estimate (6) becomes exact.

PROPOSITION 3.5. *If there exists a focus set $X \subseteq \mathbb{R}^n$ such that $S(\bar{w}) \cap X = \{\bar{x}\}$, then whenever the upper subderivative of the associated tolerance function is finite, $d^+ t_{\bar{x},X}(\bar{w})(w) < \infty$, it satisfies*

$$d^+ t_{\bar{x},X}(\bar{w})(w) = \sup_{x \in DS(\bar{w}|\bar{x})(w)} \| x \| . \tag{8}$$

Proof. The inequality

$$d^+ t_{\bar{x},X}(\bar{w})(w) \geq \sup_{x \in DS(\bar{w}|\bar{x})(w)} \| x \|$$

is provided by Proposition 3.4. To prove the opposite inequality, we consider any sequences $w' \to w$ and $\tau \downarrow 0$ and define

$$\beta := \limsup \frac{t_{\bar{x},X}(\bar{w} + \tau w')}{\tau}.$$

It is immediately evident that $t_{\bar{x},X}(\bar{w}) = 0$, and so $\beta \leq d^+ t_{\bar{x},X}(\bar{w})(w) < \infty$. From the definition of the tolerance function, we know that there exists a sequence $\tilde{x}' \in S(\bar{w} + \tau w') \cap X$ such that

$$\| \tilde{x}' - \bar{x} \| + \tau^2 \geq t_{\bar{x},X}(\bar{w} + \tau w') \geq \| \tilde{x}' - \bar{x} \| . \tag{9}$$

Defining $x' := (\tilde{x}' - \bar{x})/\tau$, we conclude from (9) that $\| x' \| \to \beta$. Thus we know that some subsequence of $\{x'\}$ converges to a point x with $\| x \| = \beta$. It follows also that x is an element of $DS(\bar{w}|\bar{x})(w)$, and thus we get the desired inequality. ∎

Proposition 3.2 identifies one important situation when the upper subderivative is always finite, so we have the following immediate corollary.

COROLLARY 3.2. *If there exists a focus set $X \subseteq \mathbb{R}^n$ such that $S(\bar{w}) \cap X = \{\bar{x}\}$ and the associated tolerance function $t_{\bar{x},X}$ is calm from above at $\bar{w}$, then its upper subderivative satisfies* (8).

The example prior to Proposition 3.5 suggests that a change of scale can also ensure (8). This is true, at least at the value $w = 0$, as the following proposition shows.

PROPOSITION 3.6. *If there exists a set $X \subseteq \mathbb{R}^n$ such that $S(\bar{w}) \cap X = \{\bar{x}\}$, then there exists a focus set $\tilde{X} \subseteq \mathbb{R}^n$ for which the associated tolerance function satisfies*

$$d^+ t_{\bar{x},\tilde{X}}(\bar{w})(0) = \sup_{x \in DS(\bar{w}|\bar{x})(0)} \| x \| .$$

Proof. The outer graphical derivative satisfies $a\,x \in DS(\bar{w}|\bar{x})(0)$ for any $a > 0$ whenever $x \in DS(\bar{w}|\bar{x})(0)$. It follows that the supremum on the right-hand side of (6) is either 0 or ∞, depending on whether or not 0 is the only element of the set $DS(\bar{w}|\bar{x})(0)$. If it is the latter case, then the result follows immediately from Proposition 3.4. On the other hand, if $DS(\bar{w}|\bar{w})(0) = \{0\}$, we can appeal to [4, Proposition 2.1] to deduce that there exists a neighborhood $\tilde{X} \subseteq \mathbb{R}^n$ of $\bar{x}$ such that the mapping $w \mapsto S(w) \cap \tilde{X}$ is calm at $\bar{w}$. From Theorem 3.3, we then know that the tolerance function $t_{\bar{x},\tilde{X}}$ is calm from above at 0, and the result follows from Corollary 3.2. ∎

4. Convergence analysis for numerical optimization. In this section, we consider the general constrained optimization problem

$$\text{minimize } f(x) \text{ over } x \in X, \tag{10}$$

where f is some extended real-valued function on a normed vector space $\mathcal{X}$, and X is a set in $\mathcal{X}$. Numerical procedures for solving the problem are all essentially concerned with determining a *minimizing sequence* of approximate solutions $x_n \in X$ for which $f(x_n)$ approaches the minimum value $\inf f := \inf_{x \in X} f(x)$. With this in mind, we construct the $|w|$-optimal solution sets

$$S(w) := \{x : f(x) \leq \inf f + |w|\} \tag{11}$$

and notice that a sequence $\{x_n\}$ is a minimizing sequence if and only if it satisfies $x_n \in S(w_n) \cap X$ for some sequence $w_n \to 0$. One classical notion of good behavior in this situation is called "Tykhonov well-posedness," which posits that there is a unique solution (i.e., $S(0) \cap X = \{\bar{x}\}$) and that every minimizing sequence converges to it (see, for example, [3]). Tolerance functions can be used to compactly characterize this property according to the following proposition.

PROPOSITION 4.1. *The optimization problem* (10) *is Tykhonov well-posed if and only if there exists a target value* $\bar{x}$ *for which the tolerance function* $t_{\bar{x},X}$ *associated with the* $|w|$*-optimal solution mapping* (11) *satisfies* $t_{\bar{x},X}(0) = 0$ *and is upper semicontinuous at* 0.

Proof. The condition that $t_{\bar{x},X}(0) = 0$ is clearly equivalent in this case to the there being a unique solution to problem (10).

Thus, assuming Tykhonov well-posedness, we need only show the upper semicontinuity of $t_{\bar{x},X}$ at 0. To this end, we take any sequence $w_n \to 0$ and consider the sequence of values $\{t_{\bar{x},X}(w_n)\}$. If for some fixed constant $\beta > 0$ there is a subsequence of these values having all of its elements greater than β, then there must be a corresponding sequence of points $x_n \in S(w_n) \cap X$ satisfying

$$\| x_n - \bar{x} \| > \beta. \tag{12}$$

However, since $x_n \in S(w_n) \cap X$ and $w_n \to 0$, the sequence $\{x_n\}$ is a minimizing sequence, and thus the bound (12) violates the Tykhonov well-posedness assumption. We conclude that the limit superior of the sequence $\{t_{\bar{x},X}(w_n)\}$ is less than or equal to zero, which proves the upper semicontinuity of $t_{\bar{x},X}$ at 0.

On the other hand, assuming that $t_{\bar{x},X}(0) = 0$, we know that the sets $S(w) \cap X$ all contain $\bar{x}$, so that $t_{\bar{x},X}(w) \geq 0$ for every $w \in \mathbb{R}$. This ensures the lower semicontinuity of $t_{\bar{x},X}$ at 0. Since $t_{\bar{x},X}$ is assumed to be upper semicontinuous at 0, it must then be continuous at 0, which evidently implies the convergence of any minimizing sequence to $\bar{x}$. ∎

This result immediately suggests the following generalization of Tykhonov well-posedness in the case in which the solution to (10) is not necessarily unique; namely, that $t_{S(0)\cap X,X}$ satisfies $t_{S(0)\cap X,X}(0) = 0$ and is upper semicontinuous at 0. This turns out to be equivalent to problem (10)'s being "metrically well-set" [2]: The solution set $S(0) \cap X$ is nonempty, and for every minimizing sequence $\{x_n\}$ one has $\operatorname{dist}(x_n, S(0) \cap X) \to 0$.

PROPOSITION 4.2. *The optimization problem* (10) *is metrically well-set if and only if the tolerance function* $t_{S(0)\cap X,X}$ *associated with* (11) *satisfies* $t_{S(0)\cap X,X}(0) = 0$ *and is upper semicontinuous at* 0.

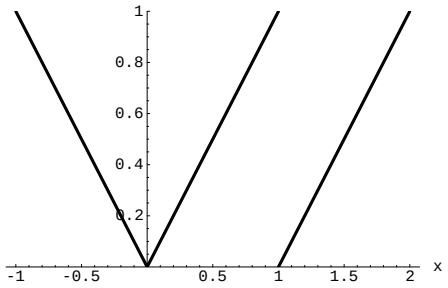


FIG. 5. *Graph of* $f(x)$.

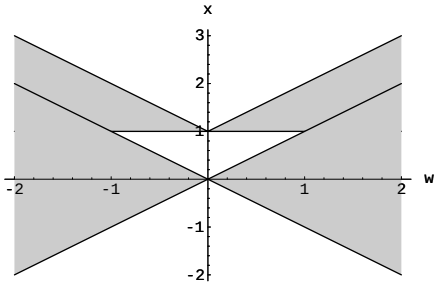


FIG. 6. *Graph of* $S(w)$.

Proof. It is clear from the definitions that $t_{S(0)\cap X,X}(0) = 0$ if and only if $S(0) \cap X \neq \emptyset$, and the rest of the proof is similar to the proof of Proposition 4.1, with $\operatorname{dist}(x_n, S(0) \cap X)$ replacing $\| x_n - \bar{x} \|$ in (12). □

Tolerance functions also very naturally suggest an entirely new concept of "scaled well-posedness." Suppose a solution to (10) were only required to a certain degree of accuracy, say within some fixed $\epsilon > 0$ of a true solution. Then we could say that the problem was *ϵ-well-posed* if, for the relaxed target set $\bar{X} = \big(S(0) \cap X\big) + \epsilon\mathbb{B}$, the tolerance function $t_{\bar{X},X}$ satisfied $t_{\bar{X},X}(0) = 0$ and was upper semicontinuous at 0. Clearly this notion of well-posedness is weaker than the other notions of well-posedness discussed above, and the weakening supports a practical consideration that practitioners might face. As an illustration of this, consider the optimization problem (10), where $X = \mathbb{R}$,

$$f(x) = \begin{cases} x - 1 & \text{if } x > 1, \\ |x| & \text{if } x \leq 1, \end{cases}$$

(see Figure 5) and its $|w|$-optimal solution sets

$$S(w) := \big\{(1, 1 + |w|] \cup [-|w|, |w|]\big\},$$

(see Figure 6) so that $S(0) \cap X = \{0\}$.

The associated tolerance function as in Proposition 4.1 is thus

$$t_{0,\mathbb{R}}(w) = \begin{cases} 0 & \text{if } w = 0, \\ 1 + |w| & \text{otherwise,} \end{cases}$$

which is clearly not upper semicontinuous at 0. It follows from Proposition 4.1 that this problem is not Tykhonov well-posed or metrically well-set (these being equivalent

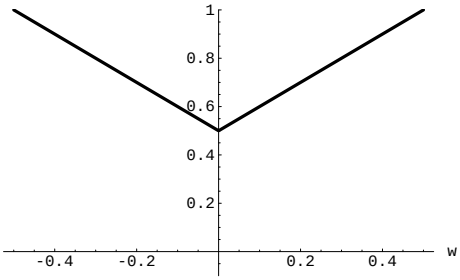


FIG. 7. *Tolerance function for* $\epsilon = 0.5$.

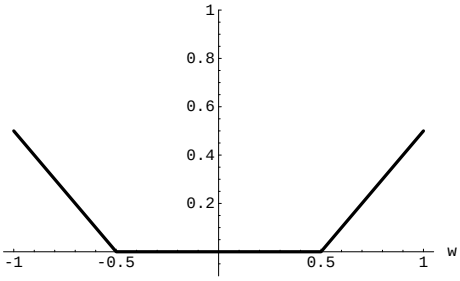


FIG. 8. *Tolerance function for* $\epsilon = 1.5$.

properties whenever there is a unique optimal solution). Moreover, for ϵ in $(0, 1)$, the tolerance function associated with the relaxed target set $\epsilon\mathbb{B}$ is

$$t_{\epsilon\mathbb{B},X}(w) = \begin{cases} 0 & \text{if } w = 0, \\ 1 - \epsilon + |w| & \text{otherwise,} \end{cases}$$

whose graph for $\epsilon = 0.5$ appears in Figure 7.

Clearly then, this problem is not ϵ-well-posed for small ϵ. However, for $\epsilon \geq 1$, the tolerance function associated with the relaxed target set is

$$t_{\epsilon\mathbb{B},X}(w) = \begin{cases} 0 & \text{if } |w| \leq \epsilon - 1, \\ 1 - \epsilon + |w| & \text{otherwise,} \end{cases}$$

which is graphed for $\epsilon = 1.5$ in Figure 8. From this tolerance function it is apparent that this problem is ϵ-well-posed for big enough ϵ.

REFERENCES

[1] J.-P. AUBIN, *Lipschitz behavior of solutions to convex minimization problems*, Math. Oper. Res., 9 (1984), pp. 87–111.
[2] E. BEDNARCZUK AND J.-P. PENOT, *Metrically well-set minimization problems*, Appl. Math. Optim., 26 (1992), pp. 273–285.
[3] A. L. DONTCHEV AND T. ZOLEZZI, *Well-Posed Optimization Problems*, Springer-Verlag, Berlin, 1993.
[4] A. J. KING AND R. T. ROCKAFELLAR, *Sensitivity analysis for nonsmooth generalized equations*, Math. Programming, 55 (1992), pp. 193–212.
[5] A. B. LEVY, *Solution sensitivity from general principles*, SIAM J. Control Optim., 40 (2001), pp. 1–38.
[6] S. M. ROBINSON, *Generalized equations and their solutions.* I. *Basic theory, Point-to-set maps and mathematical programming*, Math. Programming Stud., 10 (1979), pp. 128–141.
[7] R. T. ROCKAFELLAR AND R. J.-B. WETS, *Variational Analysis*, Springer-Verlag, Berlin, 1998.

SIAM J. OPTIM.
Vol. 12, No. 4, pp. 875–892

APPROXIMATION OF THE STABILITY NUMBER OF A GRAPH VIA COPOSITIVE PROGRAMMING*

E. DE KLERK† AND D. V. PASECHNIK†

Abstract. Lovász and Schrijver [*SIAM J. Optim.*, 1 (1991), pp. 166–190] showed how to formulate increasingly tight approximations of the stable set polytope of a graph by solving semidefinite programs (SDPs) of increasing size (lift-and-project method). In this paper we present a similar idea. We show how the stability number can be computed as the solution of a conic linear program (LP) over the cone of copositive matrices. Subsequently, we show how to approximate the copositive cone ever more closely via a hierarchy of linear or semidefinite programs of increasing size (liftings). The latter idea is based on recent work by Parrilo [*Structured Semidefinite Programs and Semi-algebraic Geometry Methods in Robustness and Optimization*, Ph.D. thesis, California Institute of Technology, Pasadena, CA, 2000]. In this way we can compute the stability number $\alpha(G)$ of any graph $G(V, E)$ after at most $\alpha(G)^2$ successive liftings for the LP-based approximations. One can compare this to the $n - \alpha(G) - 1$ bound for the LP-based lift-and-project scheme of Lovász and Schrijver. Our approach therefore requires fewer liftings for families of graphs where $\alpha(G) < O(\sqrt{n})$. We show that the first SDP-based approximation for $\alpha(G)$ in our series of increasingly tight approximations coincides with the ϑ'-function of Schrijver [*IEEE Trans. Inform. Theory*, 25 (1979), pp. 425–429]. We further show that the second approximation is tight for complements of triangle-free graphs and for odd cycles.

Key words. approximation algorithms, stability number, semidefinite programming, copositive cone, lifting

AMS subject classifications. 90C22, 68R10, 05C69, 90C25

PII. S1052623401383248

1. Introduction.

Semidefinite programming has proved to be a useful tool in formulating approximation algorithms for NP-complete problems in combinatorial optimization. The most celebrated example is the 0.878-approximation algorithm for MAX-CUT by Goemans and Williamson [8]. Their ideas have also been extended to obtain improved approximation guarantees for MAX-Bisection, MAX-3-SAT, MAX-k-CUT, and a host of other problems.

For problems which do not allow a fixed approximation guarantee, like the maximum stable set problem, semidefinite programming has also played a role. Lovász and Schrijver [16] showed how to formulate increasingly strong approximations of the maximum stable set of a graph by solving semidefinite programs (SDPs) of increasing size (liftings). They showed that their procedure is finite—the stable set polytope is obtained via a suitable projection.

In this paper we present a similar idea, but from a completely different perspective. We first show how one can compute the stability number by solving a convex conic optimization problem over the cone of copositive matrices.

Nesterov and Nemirovskii [19] showed that conic programming problems can be solved to ϵ-optimality in polynomial time if the cone in question has a *computable*[1] self-concordant barrier. As a consequence, the copositive cone does not allow a computable barrier unless $P = NP$.

*Received by the editors January 5, 2001; accepted for publication (in revised form) September 12, 2001; published electronically March 13, 2002.
http://www.siam.org/journals/siopt/12-4/38324.html
†Faculty of Information Technology and Systems, Department of Technical Mathematics and Informatics, Delft University of Technology, P.O. Box 5031, 2600 GA Delft, The Netherlands (E.deKlerk@its.tudelft.nl, D.Pasechnik@its.tudelft.nl).
[1]The gradient and Hessian of the barrier must be computable in polynomial time.

Parrilo [20] has recently suggested that the copositive cone may be approximated using linear matrix inequalities (LMIs). The approximation involves matrix variables of size $n^r \times n^r$ after r steps. We will look more closely at this procedure and will also investigate the link with a weaker linear program (LP)-based lifting scheme. Subsequently we will show that $\alpha(G)^2$ liftings are always sufficient to obtain the stability number $\alpha(G)$ of a graph $G(V,E)$ for the LP-based procedure. One can compare this to the result for the Lovász and Schrijver LP-based lift-and-project scheme, which requires $n - \alpha(G) - 1$ liftings in the worst case. For families of graphs where $\alpha(G) < O(\sqrt{n})$, our procedure therefore requires fewer liftings in the worst case.

At the first step of our SDP-based lifting scheme, we obtain the Schrijver $\vartheta'(G)$ approximation [25] to $\alpha(G)$, which is already provably stronger than the Lovász ϑ approximation [15] for certain classes of graphs. The approximation after the second lifting is tight for complements of triangle-free graphs and for odd cycles.

1.1. Preliminaries.

The maximum stable set problem. Given a graph $G(V,E)$, a subset $V' \subseteq V$ is called a stable set of G if the induced subgraph on V' contains no edges. The maximum stable set problem is to find the stable set of maximal cardinality. This problem is equivalent to finding the largest clique in the complementary graph and cannot be approximated within a factor $|V|^{\frac{1}{2}-\epsilon}$ for any $\epsilon > 0$ unless $P = NP$, or within a factor $|V|^{1-\epsilon}$ for any $\epsilon > 0$ unless $NP = ZPP$ [10]. The best known approximation guarantee for this problem is $O(|V|/(\log|V|)^2)$ [5]. For a survey of the maximum clique problem, see [3].

Conic programming. We define the following convex cones:

- The $n \times n$ symmetric matrices $\mathcal{S}_n = \left\{X \in \mathbb{R}^n \times \mathbb{R}^n,\ X = X^T\right\}$;
- The $n \times n$ symmetric positive semidefinite matrices $\mathcal{S}_n^+ = \left\{X \in \mathcal{S}_n,\ y^T X y \geq 0\ \ \forall y \in \mathbb{R}^n\right\}$;
- The $n \times n$ symmetric copositive matrices $\mathcal{C}_n = \left\{X \in \mathcal{S}_n,\ y^T X y \geq 0\ \ \forall y \in \mathbb{R}^n,\ y \geq 0\right\}$;
- The $n \times n$ symmetric completely positive matrices $\mathcal{C}_n^* = \{X = \sum_{i=1}^k y_i y_i^T,\ y_i \in \mathbb{R}^n,\ y_i \geq 0\ (i = 1, \ldots, k)\}$;
- The $n \times n$ symmetric nonnegative matrices $\mathcal{N}_n = \{X \in \mathcal{S}_n,\ X_{ij} \geq 0\ \ (i,j = 1, \ldots, n)\}$.

Recall that the completely positive cone is the dual of the copositive cone, and that the nonnegative and semidefinite cones are self-dual for the inner product $\langle X, Y\rangle := \mathrm{Tr}(XY)$, where "Tr" denotes the trace operator.

For a given cone $\mathcal{K}_n$ and its dual cone $\mathcal{K}_n^*$, we define the primal and dual pair of conic LPs:

$$(P) \qquad p^* := \inf_X \left\{\mathrm{Tr}(CX)\ :\ \mathrm{Tr}(A_i X) = b_i\ (i = 1, \ldots, m),\ X \in \mathcal{K}_n\right\},$$

$$(D) \qquad d^* := \sup_{y \in \mathbb{R}^m} \left\{b^T y\ :\ \sum_{i=1}^m y_i A_i + S = C,\ S \in \mathcal{K}_n^*\right\}.$$

If $\mathcal{K}_n = \mathcal{S}_n^+$, then we refer to semidefinite programming; if $\mathcal{K}_n = \mathcal{N}_n$, to linear programming; and if $\mathcal{K}_n = \mathcal{C}_n$, to copositive programming.

The well-known conic duality theorem (see, e.g., [24]) gives the duality relations between (P) and (D).

THEOREM 1.1 (Conic duality theorem). *If there exists an interior feasible solution $X^0 \in \operatorname{int}(\mathcal{K}_n)$ of (P) and a feasible solution of (D), then $p^* = d^*$ and the supremum in (D) is attained. Similarly, if there exist feasible y^0, S^0 for (D), where $S^0 \in \operatorname{int}(\mathcal{K}_n^*)$, and a feasible solution of (P), then $p^* = d^*$ and the infimum in (P) is attained.*

Optimization over the cones $\mathcal{S}_n^+$ and $\mathcal{N}_n$ can be done in polynomial time (to compute an ϵ-optimal solution), but copositive programming is reducible to some NP-hard problems as we will see in the next section.

2. The stability number via copositive programming. The celebrated sandwich theorem of Lovász relates three characterizing numbers of a graph $G(V, E)$: the chromatic number $\chi(\bar{G})$ of the complementary graph $\bar{G}$, the stability number $\alpha(G)$ of G, and the so-called theta number $\vartheta(G)$. The theta number can be defined as the optimal value of the following semidefinite programming relaxation of the maximum clique problem (see [15, 9]):

$$\vartheta(G) := \max \operatorname{Tr}\left(ee^T X\right) = e^T X e \tag{1}$$

subject to

$$\left.\begin{array}{rcl} X_{ij} & = & 0,\ \{i,j\} \in E\ (i \neq j) \\ \operatorname{Tr}(X) & = & 1 \\ X & \in & \mathcal{S}_n^+ \end{array}\right\}, \tag{2}$$

where e denotes the all-one vector.

The sandwich theorem states the following.

THEOREM 2.1 (Lovász's sandwich theorem). *For any graph $G = (V, E)$, one has*

$$\alpha(G) \leq \vartheta(G) \leq \chi(\bar{G}).$$

In what follows, x_S denotes the incidence vector of a stable set S of size $k = |S|$ in G, i.e.:

$$(x_S)_i = \begin{cases} 1 & \text{if } i \in S, \\ 0 & \text{otherwise.} \end{cases}$$

It is easy to check that the rank one matrix

$$X := \frac{1}{k} x_S x_S^T$$

is feasible in (2) with objective value

$$e^T X e = \frac{1}{k}\left(e^T x_S\right)^2 = \frac{k^2}{k} = k.$$

We therefore have $\alpha(G) \leq \vartheta(G)$, which proves the relevant part of the sandwich theorem.

We now show that we can actually obtain the stability number $\alpha(G)$ by replacing the semidefinite cone in (2) by the completely positive cone.

THEOREM 2.2. *Let $G(V,E)$ be given with $|E| = n$. The stability number of G is given by*

$$\alpha(G) = \max Tr\left(ee^T X\right) \tag{3}$$

subject to

$$\left.\begin{array}{rcl} X_{ij} &=& 0,\ \{i,j\} \in E\ (i \neq j) \\ Tr(X) &=& 1 \\ X &\in& \mathcal{C}_n^* \end{array}\right\}. \tag{4}$$

Proof. Consider the convex cone:

$$\mathcal{C}_G := \{X \in \mathcal{C}_n^* \ :\ X_{ij} = 0,\ \{i,j\} \in E\}.$$

The extreme rays of this cone are of the form xx^T, where $x \in \mathbb{R}^n$ is nonnegative and its support corresponds to a stable set of G. This follows from the fact that all extreme rays of $\mathcal{C}_n^*$ are of the form xx^T for nonnegative $x \in \mathbb{R}^n$. Therefore, the extreme points of the set defined by (4) are given by the intersection of the extreme rays with the hyperplane defined by $\mathrm{Tr}(X) = 1$.

Since the optimal value of problem (3) is attained at an extreme point, there is an optimal solution of the form:

$$X^* = x^* x^{*T}, \qquad x^* \in \mathbb{R}^n,\ x^* \geq 0,\ \|x^*\| = 1,$$

and where the support of x^* corresponds to a stable set, say S^*. Denoting the optimal value of problem (3) by λ, we therefore have

$$\lambda = \max_{\|x\|=1} (e^T x)^2, \quad x \geq 0, \quad \mathrm{support}(x) = \mathrm{support}(x^*).$$

The optimality conditions of this problem imply

$$x^* = \frac{1}{\sqrt{|S^*|}} x_{S^*},$$

and therefore

$$\lambda = (e^T x^*)^2 = \frac{|S^*|^2}{|S^*|} = |S^*|.$$

This shows that S^* must be the maximum stable set, and consequently $\lambda = \alpha(G)$. ∎

Note that—since $X \in \mathcal{C}_n^*$ is always nonnegative—we can simplify (3) and (4) to

$$\alpha(G) = \max\left\{\mathrm{Tr}\left(ee^T X\right)\ :\ \mathrm{Tr}\,(AX) = 0,\ \mathrm{Tr}(X) = 1,\ X \in \mathcal{C}_n^*\right\}, \tag{5}$$

where A is the adjacency matrix of G. The dual problem of (5) is given by

$$\inf_{\lambda, y \in \mathbb{R}} \left\{\lambda\ :\ Q := \lambda I + yA - ee^T \in \mathcal{C}_n\right\}. \tag{6}$$

The primal problem (5) is not strictly feasible (some entries of X must be zero), even though the dual problem (6) is strictly feasible (set $Q = (n+1)I - ee^T$). By the

conic duality theorem, we can therefore conclude only that the primal optimal set is nonempty and *not* that the dual optimal set is nonempty. We will now prove, however, that $Q = \alpha(I + A) - ee^T$ is always a dual optimal solution. This result follows from the next lemma.

LEMMA 2.3. *For a given graph $G = (V, E)$, with adjacency matrix A and stability number $\alpha(G)$, and a given parameter $\epsilon \geq 0$, the matrix*

$$Q_\epsilon^* = (1 + \epsilon)\alpha(I + A) - ee^T$$

is copositive.

Proof. Let $\epsilon \geq 0$ be given. We will show that Q_ϵ^* is copositive.

To this end, denote the standard simplex by

$$\Delta := \left\{ x \in \mathbb{R}^n \ : \ \sum_{i=1}^n x_i = 1, \ x \geq 0 \right\}$$

and note that

$$\begin{aligned} \min_{x \in \Delta} x^T Q_\epsilon^* x &= \min_{x \in \Delta} (1 + \epsilon)\alpha \left(x^T x + x^T A x \right) - x^T e e^T x \\ &= (1 + \epsilon)\alpha \min_{x \in \Delta} \left(x^T x + x^T A x \right) - 1. \end{aligned}$$

We now show that the minimum is attained at $x^* = \frac{1}{|S^*|} x_{S^*}$, where S^* denotes the maximum stable set, as before. In other words, we will show that

$$\min_{x \in \Delta} x^T Q_\epsilon^* x = \epsilon. \tag{7}$$

Let $x^* \in \Delta$ be a minimizer of $x^T Q_\epsilon^* x$ over Δ.

If the support of x^* corresponds to a stable set, then the proof is an easy consequence of the inequality:

$$\operatorname{argmax} \left\{ \|x\| \ : x \in \Delta, \ (e^T x)^2, \ x \geq 0, \ \operatorname{support}(x) = S \right\} = \frac{1}{|S|} x_S \quad \forall S \subset V,$$

which can readily be verified via the optimality conditions.

Assume therefore that the support of x^* does not correspond to a stable set, i.e., $x_i^* > 0$ and $x_j^* > 0$, where $\{i, j\} \in E$.

Now we fix all the components of x to the corresponding values of x^*, except for components i and j. Note that, defining $c_0 := \sum_{k \neq i,j} x_k^*$, one can find constants c_1, c_2, and c_3 such that

$$\begin{aligned} x^{*T} Q_\epsilon^* x^* &= \min_{x_i + x_j = 1 - c_0, x_i \geq 0, x_j \geq 0} (1 + \epsilon)\alpha \left(x_i^2 + 2 x_i x_j + x_j^2 \right) + x_i c_1 + x_j c_2 + c_3 \\ &= \min_{x_i + x_j = 1 - c_0, x_i \geq 0, x_j \geq 0} (1 + \epsilon)\alpha \left(x_i + x_j \right)^2 + x_i c_1 + x_j c_2 + c_3 \\ &= \min_{x_i + x_j = 1 - c_0, x_i \geq 0, x_j \geq 0} (1 + \epsilon)\alpha \left(1 - c_0 \right)^2 + x_i c_1 + x_j c_2 + c_3. \end{aligned}$$

The final optimization problem is simply an LP in the two variables x_i and x_j and attains it minimal value in an extremal point at which $x_i = 0$ or $x_j = 0$. We can therefore replace x^* with a vector $\bar{x}$ such that $x^{*T} Q x^* = \bar{x}^T Q \bar{x}$ and $\bar{x}_i \bar{x}_j = 0$.

By repeating this process, we obtain a minimizer of $x^T Q_\epsilon^* x$ over Δ with support corresponding to a stable set. □

The lemma shows that Q_ϵ^* is copositive and therefore ϵ-optimal in (6). For $\epsilon = 0$ we have the following result.

COROLLARY 2.4. *For any graph $G = (V, E)$ with adjacency matrix A, one has*

$$\alpha(G) = \min_{\lambda} \left\{ \lambda \ : \ \lambda(I + A) - ee^T \in \mathcal{C}_n \right\}.$$

Remark 2.1. The result of Corollary 2.4 is also a consequence of a result by Motzkin and Straus [17], who proved that

$$\frac{1}{\alpha(G)} = \min_{x \in \Delta} x^T (A + I) x,$$

where A is the adjacency matrix of G. To see the relationship between the two results, we also need the known result (see, e.g., [4]) that minimization of a quadratic function over the simplex is equivalent to a copositive programming problem:

$$\min_{x \in \Delta} x^T Q x = \min_{X \in (\mathcal{C}_n)^*} \left\{ Tr(QX) \ : \ Tr\left(ee^T X\right) = 1 \right\} = \max_{\lambda \in \mathbb{R}} \left\{ \lambda \ : \ Q - \lambda ee^T \in \mathcal{C}_n \right\}$$

for any $Q \in \mathcal{S}_n$, where the second inequality follows from the strong duality theorem.

Corollary 2.4 implies that we can simplify our conic programs even further to obtain

$$\alpha(G) = \max \left\{ \mathrm{Tr}\left(ee^T X\right) \ : \ \mathrm{Tr}\left((A + I)X\right) = 1, \ X \in \mathcal{C}_n^* \right\}, \tag{8}$$

with associated dual problem:

$$\alpha(G) = \min_{\lambda \in \mathbb{R}} \left\{ \lambda \ : \ Q := \lambda(I + A) - ee^T \in \mathcal{C}_n \right\}. \tag{9}$$

Note that both these problems are strictly feasible, and the conic duality theorem now guarantees complementary primal-dual optimal solutions.

3. Approximations of the copositive cone. The reformulation of the stable set problem as a conic copositive program makes it clear that copositive programming is not tractable (see also [23, 4]). In fact, even the problem of determining whether a matrix is not copositive is NP-complete [18].

Although we have obtained a nice convex reformulation of the stable set problem, there is no obvious way of solving this reformulation. In [4], some ideas from interior point methods for semidefinite programming are adapted for the copositive case, but convergence cannot be proved. The absence of a computable self-concordant barrier for this cone basically precludes the application of interior point methods to copositive programming.

A solution to this problem was recently proposed by Parrilo [20], who showed that one can approximate the copositive cone to any given accuracy by a sufficiently large set of linear matrix inequalities. In other words, each copositive programming problem can be approximated to any given accuracy by a sufficiently large SDP. Of course, the size of the SDP can be exponential in the size of the copositive program.

In the next subsection we will review the approach of Parrilo and subsequently work out the implications for the copositive formulation of the maximum stable set problem. We will also look at a weaker, LP-based approximation scheme.

3.1. Representations as sum-of-squares and polynomials with nonnegative coefficients. We can represent the copositivity requirement for an $(n \times n)$ symmetric matrix M as

$$P(x) := (x \circ x)^T M (x \circ x) = \sum_{i,j=1}^{n} M_{ij} x_i^2 x_j^2 \geq 0 \quad \forall x \in \mathbb{R}^n, \tag{10}$$

where "$\circ$" indicates the componentwise (Hadamard) product. We therefore wish to know whether the polynomial $P(x)$ is nonnegative for all $x \in \mathbb{R}^n$. Although one cannot answer this question in polynomial time in general, one can decide in polynomial time whether $P(x)$ can be written as a sum-of-squares. Before we give a formal exposition of the methodology, we give an example which illustrates the basic idea.

Example 3.1 (see Parrilo [20]). We show how to obtain a sum-of-squares decomposition for the polynomial $2x_1^4 + 2x_1^3x_2 - x_1^2x_2^2 + 5x_2^4$.

$$\begin{aligned}
& 2x_1^4 + 2x_1^3x_2 - x_1^2x_2^2 + 5x_2^4 \\
&= \begin{bmatrix} x_1^2 \\ x_2^2 \\ x_1x_2 \end{bmatrix}^T \begin{bmatrix} 2 & 0 & 1 \\ 0 & 5 & 0 \\ 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_1^2 \\ x_2^2 \\ x_1x_2 \end{bmatrix} \\
&= \begin{bmatrix} x_1^2 \\ x_2^2 \\ x_1x_2 \end{bmatrix}^T \begin{bmatrix} 2 & -\lambda & 1 \\ -\lambda & 5 & 0 \\ 1 & 0 & -1+2\lambda \end{bmatrix} \begin{bmatrix} x_1^2 \\ x_2^2 \\ x_1x_2 \end{bmatrix} \quad \forall \lambda \in \mathbb{R}.
\end{aligned}$$

For $\lambda = 3$ the coefficient matrix is positive semidefinite, and we obtain a sum-of-squares decomposition by taking a Choleski decomposition of the coefficient matrix.

Following the idea in the example, we represent $P(x)$ via

$$P(x) = \tilde{x}^T \tilde{M} \tilde{x}, \tag{11}$$

where $\tilde{x} = [x_1^2, \ldots, x_n^2, x_1x_2, x_1x_3, \ldots, x_{n-1}x_n]^T$ and $\tilde{M}$ is a symmetric matrix of order $n + \frac{1}{2}n(n-1)$.

Note that—as in the example—$\tilde{M}$ is not uniquely determined. The nonuniqueness follows from the identities:

$$\begin{aligned}
(x_ix_j)^2 &= (x_i^2)(x_j)^2, \\
(x_ix_j)(x_ix_k) &= (x_i^2)(x_jx_k), \\
(x_ix_j)(x_kx_l) &= (x_ix_k)(x_jx_l) = (x_ix_l)(x_jx_k).
\end{aligned}$$

It is easy to see that the possible choices for $\tilde{M}$ define an affine space.

Condition (10) will certainly hold if at least one of the following two conditions holds:

1. A representation of $P(x) = \tilde{x}^T \tilde{M} \tilde{x}$ exists with $\tilde{M}$ symmetric positive semidefinite. In this case we obtain the sum-of-squares decomposition $P(x) = \|L\tilde{x}\|^2$, where $L^TL = \tilde{M}$ denotes the Choleski factorization of $\tilde{M}$.
2. All the coefficients of $P(x)$ are nonnegative.

Note that the second condition implies the first.

Parrilo showed that $P(x)$ in (10) allows a sum-of-squares decomposition if and only if $M \in \mathcal{S}_n^+ + \mathcal{N}_n$, which is a well-known sufficient condition for copositivity. Let us define the cone $\mathcal{K}_n^0 := \mathcal{S}_n^+ + \mathcal{N}_n$. Similarly, $P(x)$ has only nonnegative coefficients if and only if $M \in \mathcal{N}_n$, which is a weaker sufficient condition for copositivity, and we define $\mathcal{C}_n^0 = \mathcal{N}_n$.

Higher-order sufficient conditions can be derived by considering the polynomial

$$P^{(r)}(x) = P(x)\left(\sum_{i=1}^n x_i^2\right)^r = \left(\sum_{i,j=1}^n M_{ij}x_i^2x_j^2\right)\left(\sum_{i=1}^n x_i^2\right)^r \tag{12}$$

and asking whether $P^{(r)}(x)$—which is a homogeneous polynomial of degree $2(r+2)$—has a sum-of-squares decomposition, or whether it has only nonnegative coefficients.

For $r = 1$, Parrilo showed that a sum-of-squares decomposition exists if and only if[2] the following system of linear matrix inequalities has a solution:

$$M - M^{(i)} \in \mathcal{S}_n^+, \quad i = 1, \ldots, n, \tag{13}$$

$$M_{ii}^{(i)} = 0, \quad i = 1, \ldots, n, \tag{14}$$

$$M_{jj}^{(i)} + 2M_{ij}^{(j)} = 0, \quad i \neq j, \tag{15}$$

$$M_{jk}^{(i)} + M_{ik}^{(j)} + M_{ij}^{(k)} \geq 0, \quad i < j < k, \tag{16}$$

where $M^{(i)}$ $(i = 1, \ldots, n)$ are symmetric matrices.

Similarly, $P^{(1)}(x)$ has only nonnegative coefficients if M satisfies the above system, but with $\mathcal{S}_n^+$ replaced by $\mathcal{N}_n$.

Note that the sets of matrices which satisfy these respective sufficient conditions for copositivity define two respective convex cones. In fact, this is generally true for all r.

DEFINITION 3.1. *Let any integer $r \geq 0$ be given. The convex cone $\mathcal{K}_n^r$ consists of the matrices $M \in \mathcal{S}_n$ for which $P^{(r)}(x)$ in* (12) *has a sum-of-squares decomposition; similarly, we define the cone $\mathcal{C}_n^r$ as the cone of matrices $M \in \mathcal{S}_n$ for which $P^{(r)}(x)$ in* (12) *has only nonnegative coefficients.*

Note that $\mathcal{C}_n^r \subset \mathcal{K}_n^r$ for all $r = 0, 1, \ldots$. (If $P(x)$ has only nonnegative coefficients, then it obviously has a sum-of-squares decomposition. The converse is not true in general.)

3.2. Upper bounds on the order of approximation. Every strictly copositive M lies in some cone $\mathcal{C}_n^r$ for r sufficiently large; this follows from the celebrated theorem of Pólya.

THEOREM 3.2 (see Pólya [21]). *Let f be a homogeneous polynomial which is positive on the simplex*

$$\Delta = \left\{ z \in \mathbb{R}^n \ : \ \sum_{i=1}^n z_i = 1, \ z \geq 0 \right\}.$$

For sufficiently large N all the coefficients of the polynomial

$$\left(\sum_{i=1}^n z_i\right)^N f(z)$$

[2] In fact, Parrilo [20] only proved the "if"-part; the proof of the converse is straightforward but tedious and can be done using the proof technique described in section 5.3 of [20].

are positive.

One can apply this theorem to the copositivity test (10) by letting $f(z) = z^T M z$ and associating $x \circ x$ with z.

In summary, we have the following theorem.

THEOREM 3.3. *Let M be strictly copositive. One has*

$$\mathcal{N}_n = \mathcal{C}_n^0 \subset \mathcal{C}_n^1 \subset \cdots \subset \mathcal{C}_n^N \ni M$$

and consequently

$$\mathcal{S}_n^+ + \mathcal{N}_n = \mathcal{K}_n^0 \subset \mathcal{K}_n^1 \subset \cdots \subset \mathcal{K}_n^N \ni M$$

for some sufficiently large N.

A tight upper bound on the size of N in Theorem 3.2 has recently been given by Powers and Reznik [22].

THEOREM 3.4 (see [22]). *Let*

$$f(z) = \sum_j \beta_j \prod_{i=1}^n z_i^{\alpha_{ij}}$$

be a homogeneous polynomial of degree d ($\sum_{i=1}^n \alpha_{ij} = d$ for all j) which is positive on the simplex Δ. The polynomial

$$\left(\sum_{i=1}^n z_i \right)^N f(z)$$

has positive coefficients if

$$N > \frac{d(d-1)L}{2\kappa} - d,$$

where

$$L = \max_j \frac{\alpha_{1j}! \alpha_{2j}! \ldots \alpha_{nj}!}{d!} |\beta_j|$$

and

$$\kappa = \min_{z \in \Delta} f(z).$$

For the problem of checking the copositivity of M we have the following.

COROLLARY 3.5. *If a symmetric $(n \times n)$ matrix M is strictly copositive, then the function*

$$P^N(z) = \left(\sum_{i,j=1}^n M_{ij} z_i z_j \right) \left(\sum_{i=1}^n z_i \right)^N$$

has only nonnegative coefficients if

$$N > L/\kappa - 2,$$

where

$$L = \max_{i,j} |M_{ij}| \tag{17}$$

and

$$\kappa = \min_{z \in \Delta} z^T M z. \tag{18}$$

Proof. The function f in Theorem 3.4 is given by $f(z) = z^T M z$ in this case. The exponents α_{ij} can now take only the values 0, 1, or 2; $d = 2$; and the coefficients β_j correspond to the entries of M. □

Note that κ is a "condition number" of M, which can be arbitrarily small, and cannot be computed in polynomial time in general unless $P = NP$.

COROLLARY 3.6. *If a strictly copositive matrix M satisfies $L/\kappa \leq r + 1$, where L and κ are respectively defined in* (17) *and* (18)*, then $M \in \mathcal{C}_n^r \subset \mathcal{K}_n^r$.*

Proof. The proof follows immediately from the definition of $\mathcal{C}_n^r$ and Corollary 3.5. □

4. Application to the maximum stable set problem. We can now define successive approximations to the stability number. In particular, we define successive LP-based approximations via

$$\zeta^{(r)}(G) = \min_{\lambda} \left\{ \lambda \ : \ Q = \lambda(I + A) - ee^T \in \mathcal{C}_n^r \right\} \tag{19}$$

for $r = 0, 1, 2, \ldots$, where we use the convention that $\zeta^{(r)}(G) = \infty$ if the problem is infeasible.

Similarly, we define successive SDP-based approximations via

$$\vartheta^{(r)}(G) = \min_{\lambda} \left\{ \lambda \ : \ Q = \lambda(I + A) - ee^T \in \mathcal{K}_n^r \right\} \tag{20}$$

for $r = 0, 1, 2, \ldots$. Note that we have merely replaced the copositive cone $\mathcal{C}_n$ in (9) by its respective approximations $\mathcal{C}_n^r$ and $\mathcal{K}_n^r$.

The minimum in (20) is always attained. The proof follows directly from the conic duality theorem if we note that $\lambda = n + 1$ always defines a matrix Q in the interior of $\mathcal{K}_n^0$ (and therefore in the interior of $\mathcal{K}_n^r \supset \mathcal{K}_n^0$ for all $r = 1, 2, \ldots$) via (20) and that

$$X^0 := \frac{1}{n^2 + n + |E|}(nI + ee^T)$$

is always strictly feasible in the associated primal problem:

$$\vartheta^{(r)}(G) = \max \left\{ \mathrm{Tr}\left(ee^T X\right) \ : \ \mathrm{Tr}\left((A + I)X\right) = 1, \ X \in (\mathcal{K}_n^r)^* \right\}.$$

The strict feasibility of X^0 follows from the fact that it is in the interior of $\mathcal{C}_n^*$: For any copositive matrix $Y \in \mathcal{C}_n$ we have

$$\mathrm{Tr}(X^0 Y) = \frac{1}{n^2 + n + |E|} \left(n\mathrm{Tr}(Y) + e^T Y e \right).$$

This expression can be zero only if Y is the zero matrix. In other words, $\mathrm{Tr}(X^0 Y) > 0$ for all nonzero $Y \in \mathcal{C}_n$, which means that X^0 is in the interior of $\mathcal{C}_n^*$. Consequently, X^0 is also in the interior of $(\mathcal{K}_n^r)^*$ for all r, since $\mathcal{C}_n^* \subset (\mathcal{K}_n^r)^*$ $(r = 0, 1, \ldots)$.

Note that

$$\alpha(G) \leq \vartheta^{(r)}(G) \leq \zeta^{(r)}(G), \quad r = 0, 1, \ldots,$$

since $\mathcal{C}_n^r \subset \mathcal{K}_n^r \subset \mathcal{C}_n$.

4.1. An upper bound for the number of liftings. We can now prove our main result.

THEOREM 4.1. *Let a graph $G(V,E)$ be given with stability number $\alpha(G)$, and let $\zeta^{(i)}$ $(i = 0, 1, 2, \dots)$ be defined as in* (19). *One has*

$$\zeta^{(0)} \geq \zeta^{(1)} \geq \cdots \geq \lfloor \zeta^{(r)} \rfloor = \alpha(G)$$

for $r \geq \alpha(G)^2$. Consequently, also $\lfloor \vartheta^{(r)} \rfloor = \alpha(G)$ for $r \geq \alpha(G)^2$.

Proof. Denote, as in the proof of Lemma 2.3,

$$Q_\epsilon^* = (1+\epsilon)\alpha(G)(I+A) - ee^T$$

for a given $\epsilon \geq 0$.

We will now prove that $Q_\epsilon^* \in \mathcal{C}_n^r$ for $r \geq \alpha(G)^2 - \alpha(G) - 2$ if

$$\epsilon := \frac{1}{\alpha(G) + 1/[\alpha(G) - 1]}. \tag{21}$$

Note that if we choose ϵ in this way, then Q_ϵ^* corresponds to a feasible solution of (19), where $\lambda = (1+\epsilon)\alpha(G) < 1 + \alpha(G)$, and we can therefore round down this value of λ to obtain $\alpha(G)$.

We proceed to bound the parameters κ and L in Corollary 3.6 for the matrix Q_ϵ^*.

- The value L is given by $L = (1+\epsilon)\alpha(G) - 1$.
- The condition number κ is given by $\kappa = \epsilon$, by (7).

Now we have

$$L/\kappa = \frac{(1+\epsilon)\alpha(G) - 1}{\epsilon} = \alpha(G)^2 + 1. \tag{22}$$

From Corollary 3.6 it now follows that $Q_\epsilon^* \in \mathcal{C}_n^r$ for $r \geq \alpha(G)^2$. ∎

Remark 4.1. If we are only interested in computing a $\zeta^{(r)} \leq (1+\epsilon)\alpha(G)$ for a given $\epsilon > 0$, then it is sufficient to choose $r = \alpha(G)/\epsilon$. To see this, note that by (22) we have

$$L/\kappa = \frac{(1+\epsilon)\alpha(G) - 1}{\epsilon} \leq \alpha(G)/\epsilon + 1 \equiv r + 1,$$

so that $Q_\epsilon^* = (1+\epsilon)\alpha(G)(I+A) - ee^T \in \mathcal{C}_n^r$ by Corollary 3.6.

Remark 4.2. The bound $\alpha(G)^2$ in Theorem 4.1 on the number of liftings can be compared to the $n - \alpha(G) - 1$ bound for the LP-based lift-and-project scheme by Lovász–Schrijver [16].[3] For families of graphs where $\alpha(G) < O(\sqrt{n})$, our LP-based lifting scheme requires fewer liftings in the worst case. This bound is satisfied, for example, by random graphs with expected edge density $\frac{1}{2}$, where one almost always has $\alpha(G) \leq 2\log_2 n$ for $n \gg 0$ (see, e.g., p. 148 of [1]).

Example 4.1. Consider the case in which $G(V,E)$ is the 5-cycle (C_5). It is well known that $\alpha(G) = 2$ and $\vartheta(G) = \vartheta'(G) = \sqrt{5}$ in this case.

[3] Lovász and Schrijver called this bound the *N-index*.

We will show that $\vartheta^{(1)}(G) = 2$; to this end, note that the matrix

$$Q = \begin{pmatrix} 1 & -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 & -1 \\ -1 & 1 & 1 & -1 & 1 \end{pmatrix} \tag{23}$$

corresponds to a feasible solution of (20) for $r = 1$, with $\lambda = 2$. The feasibility follows from the known fact that Q in (23) is in $\mathcal{K}_n^1$ (but not in $\mathcal{K}_n^0 = \mathcal{S}_n^+ + \mathcal{N}_n$); see, e.g., [20].

Example 4.2. Let $G = (V, E)$ be the complement of the graph of an icosahedron (see, e.g., [6]), where $\alpha(G) = 3$. One can solve the relevant semidefinite programming problem to obtain $\vartheta^{(1)}(G) = 1 + \sqrt{5} \approx 3.236068$.

Although $\lfloor \vartheta^{(1)}(G) \rfloor = \alpha(G)$, one has $Q := \alpha(G)(A+I) - ee^T \notin \mathcal{K}_n^1$. Thus Q gives an example of a 12×12 copositive matrix which is not in $\mathcal{K}_n^1$. This gives a partial answer to the following question posed by Parrilo [20]: "Do the copositive cone and $\mathcal{K}_n^1$ coincide for $n \times n$ matrices up to a certain size?" For this size a known lower bound is $n \geq 4$ (for $n \times n$ matrices with $n \leq 4$, $\mathcal{C}_n$ and $\mathcal{K}_n^0$ still coincide), and an upper bound is $n \leq 11$ (by this example).

4.2. Lower bounds on the number of liftings. The following theorem shows that the LP-based approximations always require at least $\alpha(G) - 1$ liftings to compute $\alpha(G)$.

THEOREM 4.2. *Let a graph $G = (V, E)$ with stability number $\alpha(G)$ be given. If $\zeta^{(r)}(G) < \infty$, then $r \geq \alpha(G) - 1$.*

Proof. Let $(1, \ldots, \alpha)$ be a maximum stable set, where $\alpha = \alpha(G)$. Then for $r = \alpha - 2$ the polynomial

$$z^T \left(t(A + I) - ee^T \right) z (e^T z)^r$$

has a monomial $Cz_1 z_2 \ldots z_\alpha$ with $C < 0$ for any value of t. This shows that problem (19) is infeasible if $r < \alpha(G) - 1$. ∎

Example 4.3. Here we show that although $\alpha(G) - 1$ liftings are necessary for computing $\alpha(G)$ via the LP-based approximations, this number of liftings is not sufficient in general. For the 4-node graph with one edge we have

$$z^T(\alpha(G)(I + A) - ee^T)z = 3z_1^2 + 3z_2^2 + 6z_1 z_2 + 3z_3^2 + 3z_4^2 - (z_1 + z_2 + z_3 + z_4)^2,$$

as $\alpha(G) = 3$ in this case, which clearly has negative coefficients. In order to get only nonnegative coefficients, we have to multiply this quadratic form by $(\sum_{i=1}^4 z_i)^6$.

5. The strength of low-order relaxations. In this section we investigate the strength of the approximations $\vartheta^{(r)}$ and $\zeta^{(r)}$ to $\alpha(G)$ for $r = 0$ and $r = 1$.

THEOREM 5.1. *If $G = (V, E)$ has $\alpha(G) = 2$, then $\zeta^{(1)}(G) \leq 3$.*

Proof. Let A be the adjacency matrix of a graph $G = (V, E)$ with $\alpha(G) = 2$, and let

$$Q(z) = z^T(3A + 3I - ee^T)z(z_1 + \cdots + z_n) := Q_+(z) - Q_-(z),$$

where

$$Q_+(z) := (3z^T(A+I)z)(z_1+\cdots+z_n), \qquad Q_-(z) = (z_1+\cdots+z_n)^3.$$

We will show that $Q(z)$ has only nonnegative coefficients, which in turn implies the theorem. The monomials of $Q_+(z)$ can be classified as follows:

$$\begin{aligned} &3z_i^3 && \forall\, i,\\ &3z_iz_j^2 && \forall\, i\neq j,\\ &6(A_{ij}+A_{ik}+A_{jk})z_iz_jz_k && \forall i<j<k. \end{aligned}$$

Note that $A_{ij}+A_{ik}+A_{jk}\geq 1$ if $i<j<k$, since $\alpha(G)=2$.

The monomials of $Q_-(z)$ are as follows:

$$\begin{aligned} &z_i^3 && \forall\, i,\\ &3z_iz_j^2 && \forall\, i\neq j,\\ &6z_iz_jz_k && \forall\, i<j<k. \end{aligned}$$

Hence $Q(z)$ has only nonnegative coefficients, as for every monomial of $Q_-(z)$ there is a monomial with the same variables in $Q_+(z)$ with a coefficient at least as large. ∎

Next we show that $\vartheta^{(0)}$ coincides with the ϑ'-function of Schrijver [25], which in turn can be seen as a strengthening of the Lovász ϑ-approximation to $\alpha(G)$.

LEMMA 5.2. *Let a graph $G=(V,E)$ be given with adjacency matrix A, and let ϑ' denote the Schrijver ϑ'-function* [25]:

$$\vartheta'(G) = \max\left\{ Tr\left(ee^TX\right) \;:\; Tr(AX)=0,\ Tr(X)=1,\ X\in(\mathcal{K}_n^0)^*\right\}.$$

Then

$$\vartheta'(G) = \vartheta^{(0)}(G).$$

Proof. Recall that

$$\vartheta^{(0)}(G) = \min_{\lambda}\left\{\lambda \;:\; \lambda(I+A)-ee^T\in\mathcal{K}_n^0\right\}, \tag{24}$$

whereas the dual formulation for $\vartheta'(G)$ is

$$\vartheta'(G) = \min_{\lambda,y}\left\{\lambda \;:\; \lambda I+yA-ee^T\in\mathcal{K}_n^0\right\}. \tag{25}$$

Further recall that $\mathcal{K}_n^0=\mathcal{S}_n^+ + \mathcal{N}_n$, and let

$$\lambda I+yA-ee^T = S+N, \quad \text{where } S\in\mathcal{S}_n^+ \text{ and } N\in\mathcal{N}_n. \tag{26}$$

Without loss of generality we assume $N_{ii}=0$ for all $i\in\{1,\ldots,n\}$, as the sum of two positive semidefinite matrices is positive semidefinite, and thus the diagonal part of N can be added to S and subtracted from N.

Assume $A_{ij}\neq 0$. Note that our choice of S and N is such that $S_{ii}=\lambda-1$. Thus, as $S_{ij}+N_{ij}=y-1$ and $S_{ii}\geq S_{ij}$,[4] one obtains $\lambda-1+N_{ij}\geq y-1$, and so

[4]Here we use the fact that $S\in\mathcal{S}_n^+$ and has a constant diagonal.

$N_{ij} \geq y - \lambda$. Hence $N + (\lambda - y)A \in \mathcal{N}_n$. Therefore $\lambda(I + A) - ee^T \in \mathcal{K}_n^0$ as long as (26) holds. Hence we can always assume that $y = \lambda$. □

Remark 5.1. As far as we know, our simplified formulation of the Schrijver ϑ'-function, namely,

$$\vartheta'(G) = \max \left\{ Tr\left(ee^T X\right) \ : \ Tr\left((A + I)X\right) = 1, \ X \in (\mathcal{K}_n^0)^* \right\},$$

is not mentioned in the literature.

Let us restate the definition of $\vartheta^{(1)}(G)$ by using (13)–(16) as follows:

$$\vartheta^{(1)}(G) := \min \beta \quad \text{subject to} \tag{27}$$

$$\beta(I + A) - ee^T - M^{(i)} \in \mathcal{S}_n^+, \quad i = 1, \ldots, n, \tag{28}$$

$$M_{ii}^{(i)} = 0, \quad i = 1, \ldots, n, \tag{29}$$

$$M_{jj}^{(i)} + 2M_{ij}^{(j)} = 0, \quad i \neq j, \tag{30}$$

$$M_{jk}^{(i)} + M_{ik}^{(j)} + M_{ij}^{(k)} \geq 0, \quad i < j < k, \tag{31}$$

where $M^{(i)}$ $(i = 1, \ldots, n)$ are symmetric matrices.

For $v \in V$, denote by $v^\perp$ the the union of the neighborhood[5] of v with v itself, and for $D \subseteq V$ denote by $G(D)$ the subgraph of G induced on D (that is, $G(D) = (D, \{(x, y) \in E \mid x, y \in D\})$). Also, $A(D)$ will denote the adjacency matrix of $G(D)$.

THEOREM 5.3. *The system of LMIs* (27)–(31) *has a feasible solution with* $\beta = 1 + \max_{k \in V}(\vartheta'(G(V - k^\perp)))$ *and* $M_{ij}^{(i)} = 0$ *for all* i, j. *Thus*

$$\vartheta^{(1)}(G) \leq 1 + \max_{k \in V}(\vartheta'(G(V - k^\perp))).$$

In particular, if $G(V - k^\perp)$ *is perfect for all* $k \in V$, *where* $k^\perp \neq V$, *then* $\vartheta^{(1)}(G) = \beta = \alpha(G)$.

Proof. Define $M = \beta(I + A) - ee^T$, and set $M_{ij}^{(i)} = 0$ for all i, j. We now apply the Schur lemma with respect to the ith row and ith column to the matrix $M - M^{(i)}$ for each $i = 1, \ldots, n$. This transforms (28) into

$$\beta I_{n-1} + \beta A(V - \{i\}) - e_{n-1}e_{n-1}^T - \Lambda^{(i)} - \frac{1}{\beta - 1} m_i m_i^T \in \mathcal{S}_n^+, \quad i = 1, \ldots, n, \tag{32}$$

where $\Lambda^{(i)}$ is obtained from $M^{(i)}$ by removing the ith row and column, and m_i is the ith row of M with the ith entry removed. In other words, $(m_i)_j = \beta - 1$ if $(i, j) \in E$, and $(m_i)_j = -1$, otherwise.

By (30), the matrix $\Lambda^{(i)}$ has zero diagonal. Thus the jth diagonal entry of the matrix on the left-hand side of (32) is zero if $(i, j) \in E$. This means that the corresponding row and column of this matrix must be zero.

Having fixed some variables as indicated, we now work out the implications from the constraint (31). There are several cases to distinguish for $\Lambda_{jk}^{(i)}$ with $j < k$ and $(i, j) \in E$, as follows:

1. $(i, k) \in E$; here $(m_i m_i^T)_{jk} = (\beta - 1)^2$.
   (a) $(j, k) \in E$; here $\Lambda_{jk}^{(i)} = 0$.
   (b) $(j, k) \notin E$; here $\Lambda_{jk}^{(i)} = -\beta$.

[5] By neighborhood of v, we mean the set of vertices adjacent to v in G.

2. $(i,k) \notin E$; here $(m_i m_i^T)_{jk} = 1-\beta$.
   (a) $(j,k) \in E$; here $\Lambda^{(i)}_{jk} = \beta$.
   (b) $(j,k) \notin E$; here $\Lambda^{(i)}_{jk} = 0$.

Note that in case 1(b), the choice $\Lambda^{(i)}_{jk} = -\beta < 0$ does not violate (31), as $\Lambda^{(j)}_{ki}$ and $\Lambda^{(k)}_{ij}$ fall under case 2(a), and thus $\Lambda^{(i)}_{jk} + \Lambda^{(j)}_{ki} + \Lambda^{(k)}_{ij} = \beta > 0$. In case 1(a), all the Λ's where the indices i, j, k appear are set to 0.

Finally, in case 2(b), one has that $\Lambda^{(i)}_{jk} = 0$, and $\Lambda^{(j)}_{ki} = 0$ together with (31) imply $\Lambda^{(k)}_{ij} \geq 0$.

For each i, denote $n_i = |V - i^\perp|$, and define $\Delta^{(i)}$ as the $n_i \times n_i$ matrix of variables that is obtained from $\Lambda^{(i)}$ after all the variables have been fixed as indicated. In other words, $\Lambda^{(i)}_{jk}$ corresponds to $\Delta^{(i)}_{jk}$ if and only if neither j nor k are adjacent to i in G.

We arrive at the following SDP:

$$\begin{gathered}
\beta^* := \min \beta \qquad \text{subject to} \\
\beta(I_{n_i} + A(V - i^\perp)) - \left(1 + \frac{1}{\beta - 1}\right) e_{n_i} e_{n_i}^T - \Delta^{(i)} \in \mathcal{S}_n^+ \qquad \forall i \in V, \\
\Delta^{(i)}_{jk} \geq 0 \qquad \forall i \in V,\ j,k \in V - i^\perp,\ (j,k) \in E, \\
\Delta^{(i)}_{jk} + \Delta^{(j)}_{ki} + \Delta^{(k)}_{ij} \geq 0 \qquad \forall i \in V,\ j,k \in V - i^\perp.
\end{gathered}$$

Note that $\beta^* \geq \vartheta^{(1)}(G)$. Multiplying both sides of all the constraints by $1 - 1/\beta$ and setting $(1 - 1/\beta)\Delta^{(i)} = \Omega^{(i)}$, one obtains

$$\beta^* = \min \beta \qquad \text{subject to} \tag{33}$$

$$(\beta - 1)(I_{n_i} + A(V - i^\perp)) - e_{n_i} e_{n_i}^T - \Omega^{(i)} \in \mathcal{S}_n^+ \qquad \forall i \in V, \tag{34}$$

$$\Omega^{(i)}_{jk} \geq 0 \qquad \forall i \in V,\ j,k \in V - i^\perp,\ (j,k) \in E, \tag{35}$$

$$\Omega^{(i)}_{jk} + \Omega^{(j)}_{ki} + \Omega^{(k)}_{ij} \geq 0 \qquad \forall i \in V,\ j,k \in V - i^\perp. \tag{36}$$

Replacing (36) by a stronger constraint $\Omega^{(i)}_{jk} \geq 0$ $(i \in V)$, we obtain n problems

$$\begin{gathered}
\beta_i^* := \min \beta_i \qquad \text{subject to} \\
(\beta_i - 1)(I_{n_i} + A(V - i^\perp)) - e_{n_i} e_{n_i}^T - \Omega \in \mathcal{S}_n^+, \\
\Omega_{jk} \geq 0 \qquad \forall j,k \in V,
\end{gathered}$$

so that

$$\max_{i \in V} \beta_i^* \geq \beta^* \geq \vartheta^{(1)}(G) \geq \alpha(G). \tag{37}$$

By the definition of $\vartheta^{(0)}$, one has $\beta_i^* - 1 = \vartheta^{(0)}(G(V - i^\perp))$, which equals $\vartheta'(G(V - i^\perp))$ by Lemma 5.2. If $G(V - i^\perp)$ is perfect for all $i \in V$, then

$$\max_{i \in V} \beta_i^* = \max_{i \in V} \vartheta^{(0)}(G(V - i^\perp)) + 1 = \max_{i \in V} \alpha(G(V - i^\perp)) + 1 = \alpha(G).$$

Thus $\vartheta^{(1)}(G) = \alpha(G)$ by (37). □

Thus, for instance, the 5-cycle example of the previous section can be generalized to all cycles.

COROLLARY 5.4. *Let $G(V,E)$ be a cycle of length n. One has $\vartheta^{(1)}(G) = \alpha(G)$. Similarly, $\alpha(G) = \vartheta^{(1)}(G)$ if G is a wheel.*

Proof. Let $G = (V,E)$ be a cycle of length n. The required result now immediately follows from Theorem 5.3 by observing that $G(V - v^{\perp})$ is an $(n-3)$-path for all $v \in V$. The proof for wheels is similar. □

Also, complements of triangle-free graphs are recognized.

COROLLARY 5.5. *If $G = (V,E)$ has stability number $\alpha(G) = 2$, then $\vartheta^{(1)}(G) = 2$.*

Proof. The proof immediately follows from Theorem 5.3 by observing that $G(V - v^{\perp})$ is a clique (or the empty graph) for all $v \in V$. □

As a consequence, the complements of cycles or wheels are also recognized. The proof proceeds in the same way as before and is therefore omitted.

COROLLARY 5.6. *Let $G(V,E)$ be the complement of a cycle or of a wheel. In both cases one has $\vartheta^{(1)}(G) = \alpha(G)$.*

Remark 5.2. It is worth mentioning that neither the upper bound $\beta = 1 + \max_{k \in V}(\vartheta'(G(V - k^{\perp})))$ on $\vartheta^{(1)}$ given in Theorem 5.3 nor the upper bound β^* used in its proof is sharp. This is demonstrated by the example of the 7-vertex graph G obtained by taking an isolated node and a pentagon and joining these six nodes with an extra node. (The result can be viewed as a pentagon "umbrella.") Then $\beta^* = 3.068$, while $\vartheta^{(1)}(G) = \alpha(G) = 3$, and the bound given by Theorem 5.3 is $\beta = 1 + \max_{k \in V}(\vartheta'(G(V - k^{\perp}))) = 1 + \sqrt{5} \approx 3.23$.

We conjecture that the result of Corollary 5.5 can be extended to include all values of α.

CONJECTURE 5.1. *If $G = (V,E)$ has stability number $\alpha(G)$, then $\vartheta^{(\alpha(G)-1)}(G) = \alpha(G)$.*

Note that we have proven the conjecture for $\alpha(G) \leq 2$.

6. Conclusions and future work. We have introduced two successive lifting procedures for computing the stability number $\alpha(G)$ of a graph. The first procedure involves generalizations of the Schrijver ϑ'-function, which in turn is a generalization of the well-known Lovász ϑ-function. These generalized ϑ-functions were denoted by $\vartheta^{(r)}$ $(r = 0, 1, \dots)$, where $\vartheta^{(0)}(G) = \vartheta'(G)$ for all $G = (V,E)$, and $\vartheta^{(0)}(G) \geq \vartheta^{(1)}(G) \geq \cdots \geq \lfloor \vartheta^{(N)} \rfloor = \alpha(G)$ for some sufficiently large N. We have also introduced related LP-based approximations to $\alpha(G)$, namely, the numbers $\zeta^{(r)} \geq \vartheta^{(r)}$, which satisfy $\lfloor \zeta^{(N)} \rfloor = \alpha(G)$ if $N \geq \alpha^2(G)$. This can be compared to the $n - \alpha(G) - 1$ bound for the LP-based lift-and-project scheme by Lovász and Schrijver [16]. For classes of graphs where $\alpha(G) < O(\sqrt{n})$, our procedure therefore requires fewer liftings in the worst case. At step r of the respective procedures, an SDP (respectively, LP) problem involving matrix variables of size $n^{r+1} \times n^{r+1}$ is solved.

The underlying idea for these approximations was to write the maximum stable set problem as a conic linear program over the cone of copositive matrices, and to subsequently perform successive approximations of this cone by using linear (matrix) inequalities. This link between copositive matrices and the maximum stable set has also allowed us to give a partial answer to a question posed by Parrilo [20] concerning a class of copositive matrices (see Example 4.2).

There have been several—seemingly different—lift-and-project strategies for approximating combinatorial optimization problems. Apart from the approach of Lovász and Schrijver [16] (see also [7, 11]) for the stable set polytope, Anjos and Wolkowicz [2] have introduced a technique of successive Lagrangian relaxations for the MAX-CUT problem, which also leads to semidefinite programming relaxations of size $(n^r \times n^r)$ after r relaxations. Most recently, Laserre [13, 14] has introduced yet another lift-and-

project approach, based on the properties of moment matrices. Laurent [11, 12] has recently shown the relationship between these approaches. In the same vein, it would be very interesting to explore possible links between the approach of Lovász and Schrijver and the lifting scheme introduced in this paper. In particular, it seems unlikely that the bound on the number of liftings ($r = \alpha(G)^2$) is tight: The Lovász–Schrijver SDP-based procedure only requires $\alpha(G)$ liftings in the worst case. We conjecture that the proof of Theorem 5.3 can be extended to show that $\alpha(G) - 1$ liftings always suffice for our SDP-based lifting scheme.

Another interesting line of research is to further investigate the theoretical properties of the $\vartheta^{(1)}$ number introduced in this paper. Actual computation of this number involves SDPs with $n^2 \times n^2$ matrices having an $n \times n$ block diagonal structure, and it can still be done for graphs of small size (say $n \leq 30$) with current interior point technology.

Acknowledgments. The authors would like to thank Immanuel Bomze, Pablo Parrilo, and Kees Roos for their comments on a draft version of this paper. They are also indebted to Alexei Pastor for correcting an earlier proof of Corollary 2.4.

REFERENCES

[1] N. Alon, J.H. Spencer, and P. Erdös, *The Probabilistic Method*, Wiley, New York, 1992.
[2] M.F. Anjos and H. Wolkowicz, *A strengthened SDP relaxation via a second lifting for the max-cut problem*, Discrete Appl. Math., to appear.
[3] I.M. Bomze, M. Budinich, P.M. Pardalos, and M. Pelillo, *The maximum clique problem*, in Handbook of Combinatorial Optimization, Suppl. Vol. A, D.-Z. Du and P.M. Pardalos, eds., Kluwer, Dordrecht, The Netherlands, 1999, pp. 1–74.
[4] I.M. Bomze, M. Dür, E. de Klerk, C. Roos, A. Quist, and T. Terlaky, *On copositive programming and standard quadratic optimization problems*, J. Global Optim., 18 (2000), pp. 301–320.
[5] R. Boppana and M.M. Halldórsson, *Approximating maximum independent sets by excluding subgraphs*, BIT, 32 (1992), pp. 180–196.
[6] H.S.M. Coxeter, *Introduction to Geometry*, Wiley, New York, 1969.
[7] M.X. Goemans and L. Tunçel, *When Does the Positive Semidefiniteness Constraint Help in Lifting Procedures*, manuscript, Department of Mathematics, MIT, Cambridge, MA, 2000.
[8] M.X. Goemans and D.P. Williamson, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, J. ACM, 42 (1995), pp. 1115–1145.
[9] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, Berlin, 1988.
[10] J. Hastad, *Clique is hard to approximate within* $|V^{1-\epsilon}|$, Acta Math., 182 (1999), pp. 105–142.
[11] M. Laurent, *Tighter linear and semidefinite relaxations for max-cut based on the Lovász–Schrijver lift-and-project procedure*, SIAM J. Optim., 12 (2001), pp. 345–375.
[12] M. Laurent, *A Comparison of the Sherali-Adams, Lovász-Schrijver and Lasserre Relaxations for* 0-1 *Programming*, Technical report PNA-R0108, CWI, Amsterdam, 2001.
[13] J.B. Lasserre, *Global optimization with polynomials and the problem of moments*, SIAM J. Optim., 11 (2001), pp. 796–817.
[14] J.B. Lasserre, *An explicit exact SDP relaxation for nonlinear* 0-1 *programs*, in Proceedings of the 8th International Integer Programming and Combinatorial Optimization Conference, Lect. Notes in Comput. Sci. 2081, Springer, Berlin, 2001, pp. 293–303.
[15] L. Lovász, *On the Shannon capacity of a graph*, IEEE Trans. Inform. Theory, 25 (1979), pp. 1–7.
[16] L. Lovász and A. Schrijver, *Cones of matrices and set-functions and* 0–1 *optimization*, SIAM J. Optim., 1 (1991), pp. 166–190.
[17] T.S. Motzkin and E.G. Straus, *Maxima for graphs and a new proof of a theorem of Turán*, Canadian J. Math., 17 (1965), pp. 533–540.
[18] K.G. Murty and S.N. Kabadi, *Some NP-complete problems in quadratic and linear programming*, Math. Programming, 39 (1987), pp. 117–129.

[19] Yu. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM Stud. Appl. Math. 13, SIAM, Philadelphia, 1994.

[20] P.A. Parrilo, *Structured Semidefinite Programs and Semi-algebraic Geometry Methods in Robustness and Optimization*, Ph.D. thesis, California Institute of Technology, Pasadena, CA, 2000. Available online at: http://www.cds.caltech.edu/∼pablo/.

[21] G. Pólya, *Über positive Darstellung von Polynomen*, Vierteljschr. Naturforsch. Ges. Zürich, 73 (1928), pp. 141–145; Collected Papers, Vol. 2, MIT Press, Cambridge, MA, London, 1974, pp. 309–313.

[22] V. Powers and B. Reznick, *A new bound for Pólya's theorem with applications to polynomials positive on polyhedra*, J. Pure Appl. Algebra, 164 (2001), pp. 221–229.

[23] A.J. Quist, E. de Klerk, C. Roos, and T. Terlaky, *Copositive relaxation for general quadratic programming*, Optim. Methods Softw., 9 (1998), pp. 185–209.

[24] J. Renegar, *A Mathematical View of Interior-Point Methods in Convex Optimization*, SIAM, Philadelphia, 2001.

[25] A. Schrijver, *A comparison of the Delsarte and Lovász bounds*, IEEE Trans. Inform. Theory, 25 (1979), pp. 425–429.

SIAM J. OPTIM.
Vol. 12, No. 4, pp. 893–912

LOCATING THE LEAST 2-NORM SOLUTION OF LINEAR PROGRAMS VIA A PATH-FOLLOWING METHOD*

YUN-BIN ZHAO† AND DUAN LI‡

Abstract. A linear program has a unique least 2-norm solution, provided that the linear program has a solution. To locate this solution, most of the existing methods were devised to solve certain equivalent perturbed quadratic programs or unconstrained minimization problems. We provide in this paper a new theory which is different from these traditional methods and is an effective numerical method for seeking the least 2-norm solution of a linear program. The essence of this method is a (interior-point-like) path-following algorithm that traces a newly introduced regularized central path that is fairly different from the central path used in interior-point methods. One distinguishing feature of our method is that it imposes no assumption on the problem. The iterates generated by this algorithm converge to the least 2-norm solution whenever the linear program is solvable; otherwise, the iterates converge to a point which gives a minimal KKT residual when the linear program is unsolvable.

Key words. linear programming, path-following algorithm, regularized central path, least 2-norm solution

AMS subject classifications. 90C05, 90C33, 90C51, 65K05

PII. S1052623401386368

1. Introduction.

Consider the linear program

$$\min\{c^T x : \ Ax \geq b, \ x \geq 0\}, \tag{1.1}$$

where $A \in R^{m\times n}$, $c \in R^n$, and $b \in R^m$. The dual problem for the above linear program can be written as

$$\max\{b^T y : \ A^T y \leq c, \ y \geq 0\}. \tag{1.2}$$

Let S_P^* and S_D^* denote the optimal solution sets (possibly empty) of the problems (1.1) and (1.2), respectively. If a linear program has an optimal solution, it is said to be solvable; otherwise, it is unsolvable. According to linear programming theory (see, for instance, Theorem 1.13 in [34]), the primal (1.1) and the dual (1.2) have optimal solutions if and only if both problems have feasible solutions. If one of problems (1.1) or (1.2) has no feasible solution, then the other one is either unbounded or has no feasible solution, and if one of problems (1.1) or (1.2) is unbounded, then the other one has no feasible solution. Therefore, we may say that the primal problem is solvable if and only if the dual is solvable. Equivalently, the primal is unsolvable if and only if the dual is unsolvable.

Throughout this paper, we denote by $\|\cdot\|_\infty$ the ∞-norm of a vector, and by $\|\cdot\|_2$ the 2-norm, i.e., Euclidean norm. The purpose of this paper is to give a new method

*Received by the editors March 14, 2001; accepted for publication (in revised form) September 25, 2001; published electronically March 13, 2002. This work was partially supported by grant CUHK4392/99E, Research Grants Council, Hong Kong.
http://www.siam.org/journals/siopt/12-4/38636.html
†Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Shatin, New Territory, Hong Kong, and Institute of Applied Mathematics, AMSS, Chinese Academy of Sciences, Beijing, China (ybzhao@se.cuhk.edu.hk).
‡Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Shatin, New Territory, Hong Kong (dli@se.cuhk.edu.hk).

for finding the least 2-norm solutions of both primal and dual linear programs, i.e., to find $x^* \in S_P^*$ and $y^* \in S_D^*$ such that

$$\|x^*\|_2 \le \|x\|_2, \ \|y^*\|_2 \le \|y\|_2 \qquad \text{for all } x \in S_P^* \text{ and } y \in S_D^*.$$

Given a certain norm, the problem of finding the least-norm solution to some optimization problems or other applied mathematical problems has been studied by many authors such as Tikhonov and Arsenin [30], Tucker [28, 29], Parsons and Tucker [21], and Wolfe [32, 33]. In particular, many authors have studied the theoretical property of the least 2-norm solution of a linear program and have tried to design numerical methods to compute this solution. See, for instance, Tikhonov and Arsenin [30], Mangasarian [13, 14, 15, 16], Mangasarian and Meyer [19], Mangasarian and De Leone [18], Lucidi [11, 12], Skarin [25], Kiwiel [6, 7], Smith and Wolkowicz [24], and Kanzow, Qi, and Qi [5]. Note that the least 2-norm solution of a linear program could be a vertex of the feasible set and also could be a relative interior point of the optimal faces. Thus, in the general case, both simplex methods and interior-point methods (see [23, 34]) may not find the least 2-norm solution of a linear program.

The first method for the least-norm solution of a linear program was the canonical Tikhonov regularization method [30]. The basic idea of this method is to solve successively the following quadratic problem in x:

$$\min\{c^T x + \mu\|x\|_2^2 : \ Ax \ge b, \ x \ge 0\}, \tag{1.3}$$

where μ is a positive parameter. For each $\mu > 0$, denote by $x(\mu)$ the solution to the above quadratic program. Tikhonov (see [30]) showed that $x(\mu)$ converges, as $\mu \to 0$, to the least 2-norm solution of (1.1). Later, Mangasarian and Meyer [19] showed that there exists a $\bar{\mu} > 0$ such that for any $\mu \in (0, \bar{\mu}]$ the perturbed quadratic program (1.3) becomes an exact problem, i.e., for any $\mu \in (0, \bar{\mu}]$ the solution $x(\mu) = \bar{x}$, where $\bar{x}$ is the least 2-norm solution of (1.1). Based on this observation, Mangasarian [14, 16] used successive overrelaxation (SOR) methods to solve the dual problem of (1.3). As pointed out by Lucidi [11], the main advantage of SOR algorithms is that they preserve the sparsity structure of the problem and thus can tackle large-scale problems. However, the main difficulty encountered by this method appears to be that of knowing such a threshold value of $\bar{\mu}$. Thus, in general, it is not clear whether a value of μ is small enough such that the solution $x(\mu)$ of (1.3) is the least 2-norm solution of (1.1). Even the condition $x(\mu^{k+1}) = x(\mu^k)$ with $\mu^{k+1} < \mu^k$ does not imply that $x(\mu^{k+1})$ is the least 2-norm solution of (1.1).

There are two classes of ways to circumvent this difficulty. The first class of approaches, including those used by Lucidi [11, 12] and Kiwiel [6], attempts to establish an effective computational criterion to check whether the current perturbed quadratic program is exact. However, Lucidi's methods [11, 12] require that the linear program (1.1) be nondegenerate (the gradients of active constraints at the least 2-norm solution $\bar{x}$ are linearly independent), whereas Kiwiel's method [6] solves the perturbed quadratic program by finite active-set methods (see, for example, Best [1] and Kiwiel [8]), which may not be effective for large-scale problems. The second class of methods was developed by Mangasarian and De Leone [18]. In their method, a decreasing sequence $\mu^k \to 0$ is stipulated, and for each μ^k an approximate solution $x(\mu^k)$ is computed by applying SOR algorithms to the dual problem of (1.3). They showed that if the residual inaccuracy of $x(\mu^k)$ falls below a certain threshold related to μ^k, the approximate sequence $\{x(\mu^k)\}$ converges to the least 2-norm solution as $k \to \infty$.

It is worth mentioning that Kiwiel [7] extended the method in [18] to piecewise linear programs, which include the linear program as a special case.

In summary, the aforementioned approaches focus on solving problem (1.3) or its dual problem. We may categorize them as (sequential) quadratic programming methods for finding the least 2-norm solution of a linear program. Of course, in addition to SOR methods, problem (1.3) or its dual problem may also be solved by other algorithms; see Lin and Pang [10] and the references therein.

Besides the methods using an equivalent perturbed quadratic program, the least 2-norm solution of a linear program can also be obtained by solving an equivalent unconstrained convex minimization problem. The first result in this aspect was due to Mangasarian [15]. In [15], Mangasarian first proved that the least 2-norm solution problem of linear programs can be transformed into an equivalent unconstrained minimization of a parameter-free convex continuously differentiable function. As a result, some unconstrained optimization methods can be used to find the least 2-norm solution of a linear program. Recently, Kanzow, Qi, and Qi [5] studied another equivalent unconstrained reformulation of the least 2-norm solution problem. Their method is based on the result of Smith and Wolkowicz [24], which is essentially related to the result of Mangasarian and Meyer (Corollary 2 in [19]). Based on their reformulation, Kanzow, Qi, and Qi [5] proposed a Newton-type method to solve their unconstrained minimization problem. However, unlike Mangasarian's reformulation, the unconstrained minimization problem in [5] contains a parameter which is required to be sufficiently large (but that is unknown in advance). Also, their convergence analysis needs certain relatively restrictive assumptions such as the strict feasibility of (1.1) and the nondegeneracy of the least 2-norm solution.

It is well known that a linear program can also be formulated as an equivalent linear complementarity problem (LCP). In fact, writing out the KKT optimality conditions of linear program (1.1), we have

$$\left\{\begin{array}{l} s + A^T y - c = 0, \\ z - Ax + b = 0, \\ (x, y, s, z) \geq 0,\ x^T s = y^T z = 0, \end{array}\right. \tag{1.4}$$

which can be written as the following monotone LCP:

$$\left[\begin{array}{c} s \\ z \end{array}\right] = \left[\begin{array}{cc} O & -A^T \\ A & O \end{array}\right] \left[\begin{array}{c} x \\ y \end{array}\right] + \left[\begin{array}{c} c \\ -b \end{array}\right] \geq 0, \quad \left[\begin{array}{c} x \\ y \end{array}\right] \geq 0, \quad \left[\begin{array}{c} s \\ z \end{array}\right]^T \left[\begin{array}{c} x \\ y \end{array}\right] = 0. \tag{1.5}$$

Thus, locating the least 2-norm solutions of primal and dual linear programs is completely equivalent to finding a least 2-norm solution of the above monotone LCP. For LCPs, the least 2-norm solution has also been extensively studied by several authors, for example, Subramanian [26], Mangasarian [17], Sznajder and Gowda [27], and Zhao and Li [35, 36, 37]. The least 2-norm solution of a complementarity problem is also related to Tikhonov regularization methods for complementarity problems (see, for instance, Isac [4], Facchinei [2], and Facchinei and Kanzow [3]). In fact, in [26, 27, 35], it is shown that the Tikhonov regularization trajectory of a monotone complementarity problem converges to the least 2-norm solution of the problem. In [35], a new homotopy continuation trajectory, later called a regularized central path in [37], is constructed for complementarity problems. It turns out that this new trajectory converges to the least 2-norm solution of a monotone complementarity problem as the parameter approaches zero.

Motivated by recent results in [35, 36, 37], the purpose of this paper is to develop a new theory and an alternative computational method for the least 2-norm solution of linear programs. The proposed method is different from most of the existing methods, which either require additional conditions besides the solvability of the problem or have to solve quadratic programs successively. The proposed algorithm in this paper does not impose any assumption on the problem. It is convergent regardless of whether the linear program is solvable or not. If problem (1.1) is solvable, then the iterates generated by the proposed algorithm converge to the least 2-norm solution. If the problem has no solution, the iterates still converge to a point which gives a minimal KKT residual of the problem (1.1). This algorithm is a kind of interior-point-like path-following algorithm (but not an interior-point algorithm), which is based on a new concept of the regularized central path $\{x(\mu) : \mu > 0\}$ of a linear program. Remarkable features of this path are that its existence and convergence for any (solvable or unsolvable) linear program can be guaranteed. These features distinguish it from the conventional central path, whose existence and boundedness require that the primal and the dual have interior points, which in turn implies that both primal and dual problems have bounded solution sets (see Theorem 5.10.1 and the corollary of Theorem 3.4.1 in [23]). When a linear program has an unbounded solution set, in which case the problem is unstable (see Robinson [22]), the interior point does not exist, and hence the central path does not exist. However, the regularized central path proposed in this paper always exists for any linear program and converges, as μ tends to zero, to the least 2-norm solution of any solvable linear program, despite the unboundedness of its solution set. This motivates us to design a new path-following method for linear programs. To our knowledge, the proposed method can be viewed as the first (interior-point-like) path-following algorithm for the least 2-norm solution of a linear program.

In the next section, we introduce the concept of a regularized central path for linear programs. In section 3, we specify a path-following algorithm. In section 4, we prove the global convergence of the algorithm. The unsolvable case is studied in section 5. Numerical results are illustrated in section 6. Conclusions are given in the last section.

Throughout the paper, we use the standard notation found in the interior-point algorithm literature. For example, all the vectors are column. For vectors u and $v \in R^n$, we also use (u, v) to denote the column vector $(u^T, v^T)^T$ if there is no confusion. The vector e denotes the vector of ones, and its dimension, unless otherwise stated, depends on the context. For a vector x, x_+ denotes the vector with components $(x_+)_i = \max\{x_i, 0\}, i = 1, \ldots, n$, and X denotes the corresponding diagonal matrix, i.e., $X = \mathrm{diag}(x)$. R^n_+ denotes the nonnegative orthant of n-dimensional Euclidean space R^n. If $x \in R^n_+$, we also write it as $x \geq 0$. In particular, $x > 0$ means that all components of x are positive.

2. Regularized central path. We begin by recalling the concept of a central path of a linear program. The linear program (1.1) can be rewritten as

$$\min\{c^T x : Ax - z = b,\ (x, z) \geq 0\}.$$

The central path is defined by a parameter $\mu > 0$, and for each $\mu > 0$ it is the solution to the following logarithmic barrier problem:

$$\begin{array}{ll} \min & c^T x - \mu \left(\displaystyle\sum_{i=1}^{n} \log x_i + \sum_{i=1}^{m} \log z_i \right) \\ \text{subject to (s.t.)} & Ax - z = b \\ & (x > 0, z > 0). \end{array}$$

The Lagrangian of the above problem is

$$L_\mu(x,y,z) = c^T x + y^T(z - Ax + b) - \mu\left(\sum_{i=1}^{n}\log x_i + \sum_{i=1}^{m}\log z_i\right), \tag{2.1}$$

where $y \in R^m$ is the Lagrange multiplier vector corresponding to the constraint $Ax - z = b$. Thus, the central path is actually defined by the stationary point of the above Lagrange function; that is,

$$0 = \nabla L_\mu(x,y,z) = \left(\frac{\partial L_\mu}{\partial x}, \frac{\partial L_\mu}{\partial y}, \frac{\partial L_\mu}{\partial z}\right) = \begin{pmatrix} c - A^T y - \mu X^{-1} e \\ z - Ax + b \\ y - \mu Z^{-1} e \end{pmatrix},$$

which, by setting $s = \mu X^{-1} e > 0$, can be written as

$$\begin{gathered} Xs = \mu e, \\ Yz = \mu e, \\ s + A^T y - c = 0, \\ z - Ax + b = 0, \\ (x,y,s,z) > 0. \end{gathered}$$

It is well known that for every $\mu > 0$ the above system has a unique solution denoted by $(x(\mu), y(\mu), s(\mu), z(\mu))$ if and only if the primal and the dual problems have interior points. If the primal and the dual have interior points, then $x(\mu)$ converges (as $\mu \to 0$) to the analytic center of the primal optimal face, and $y(\mu)$ converges to the dual optimal face (see Theorems 5.10.1 and 5.10.3 in [23] or Theorems 2.16 and 2.17 in [34]). Clearly, the analytic center is not necessarily the least 2-norm solution.

It is worth pointing out that the existence of the central path is not guaranteed for the case when the problem has an unbounded optimal solution set, i.e., when the linear program has no interior point (see, for instance, Theorem 5.10.1 and the corollary of Theorem 3.4.1 in [23]). We now construct a new smooth path that is expected to converge to the least 2-norm solution even when the problem has an unbounded solution set. We first define a perturbed Lagrange function of (2.1). Notice that in (2.1), the Lagrange multiplier y is related to the decision variable of the dual problem. In fact, let $\nabla L_\mu(x(\mu), y(\mu), z(\mu)) = 0$. If $(x(\mu), y(\mu), z(\mu)) \to (x^*, y^*, z^*)$ as $\mu \to 0$, then (x^*, y^*, z^*) satisfies the KKT system (1.4). By the theory of linear programming, y^* is an optimal solution to the dual problem. Thus, in order to obtain the least 2-norm solution of the primal and the dual linear programs, we consider the following augmented Lagrange function:

$$\begin{aligned} \mathcal{L}_{(\mu,\theta)}(x,y,z) :&= c^T x + y^T(z - Ax + b) - \mu\left(\sum_{i=1}^{n}\log x_i + \sum_{i=1}^{m}\log z_i\right) \\ &\quad + \frac{1}{2}\theta\|(x,y)\|_2^2, \end{aligned} \tag{2.2}$$

where μ and θ are two positive parameters. The penalty term $\theta\|(x,y)\|_2^2$ attached to the Lagrangian (2.1) is used to force the stationary point of the augmented Lagrange function to approach the least 2-norm solution. It will be seen from our later discussion that the above augmented form is a judicious choice for locating the least 2-norm solution and for covering the aforementioned case of an unbounded solution set. Although the parameters μ and θ can be independent, for simplicity we consider

here only the case of $\theta = \mu^p$, where $p \in (0,1)$ is a fixed constant. Thus, the above function (2.2) can be written in the following one-parameter form:

$$\Phi_\mu(x,y,z) := c^T x + y^T(z - Ax + b) - \mu\left(\sum_{i=1}^{n}\log x_i + \sum_{i=1}^{m}\log z_i\right) + \frac{1}{2}\mu^p\|(x,y)\|_2^2. \tag{2.3}$$

We are now ready to define the concept of a regularized central path. Analogous to the central path which is the stationary point of Lagrange function (2.1), the so-called *regularized central path* can be defined by the stationary point of the augmented Lagrange function (2.3), that is, $\nabla\Phi_\mu(x,y,z) = 0$. Thus, we have the following definition.

DEFINITION 2.1. *The curve* $\{(x(\mu), y(\mu), s(\mu), z(\mu)) : \mu > 0\}$ *is said to be a regularized central path if for each* $\mu > 0$, $(x(\mu), y(\mu), s(\mu), z(\mu))$ *is the solution to the following system:*

$$\begin{cases} Xs = \mu e, \\ Yz = \mu e, \\ s + A^T y - c = \mu^p x, \\ z - Ax + b = \mu^p y, \\ (x,y,s,z) > 0. \end{cases} \tag{2.4}$$

The set $\{(x(\mu), z(\mu)) : \mu > 0\}$ can be called the primal regularized central path, and $\{(y(\mu), s(\mu)) : \mu > 0\}$ the dual regularized central path. The following result states that the existence of the regularized central path can be ensured in all situations. This path converges to the unique least 2-norm solution as long as the linear program in question is solvable. Thus, the regularized central path provides us with a novel and powerful solution scheme for linear programming problems.

THEOREM 2.1. *For any linear program* (1.1)*, the following hold:*

(i) *For each* $\mu > 0$, *system* (2.4) *has a unique solution* $(x(\mu), z(\mu), y(\mu), s(\mu)) > 0$.

(ii) *For any finite number* $0 < \hat{\mu} < \infty$, *the set* $\{(x(\mu), z(\mu), y(\mu), s(\mu)) : \mu \in (0, \hat{\mu}]\}$ *is bounded if and only if the linear problem* (1.1) *is solvable.*

(iii) *Linear problem* (1.1) *is solvable if and only if* $(x(\mu), z(\mu), y(\mu), s(\mu))$ *converges, as* $\mu \to 0$, *to* (x^*, z^*, y^*, s^*), *where* x^* *and* y^* *are least* 2*-norm solutions of the primal and the dual problems, respectively.*

Proof. It is evident that system (2.4) can be written as

$$\begin{bmatrix} s \\ z \end{bmatrix} = \begin{bmatrix} \mu^p I & -A^T \\ A & \mu^p I \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} c \\ -b \end{bmatrix} > 0, \quad \begin{bmatrix} x \\ y \end{bmatrix} > 0, \quad U\begin{bmatrix} s \\ z \end{bmatrix} = \mu e, \tag{2.5}$$

where $e \in R^{m+n}$ and $U = \begin{bmatrix} X & O \\ O & Y \end{bmatrix}$. Denote

$$M = \begin{bmatrix} O & -A^T \\ A & O \end{bmatrix}, \quad u = \begin{bmatrix} x \\ y \end{bmatrix}, \quad v = \begin{bmatrix} s \\ z \end{bmatrix}, \quad q = \begin{bmatrix} c \\ -b \end{bmatrix}.$$

Then system (2.5) can be further written as

$$v = Mu + q + \mu^p u > 0, \quad u > 0, \quad Uv = \mu e.$$

Under the one-to-one transformation of $\mu = \varepsilon/(1-\varepsilon)$, where $\varepsilon \in (0,1)$, the above system is equivalent to

$$(1-\varepsilon)v = (1-\varepsilon)(Mu+q+\phi(\varepsilon)u) > 0, \quad u > 0, \quad (1-\varepsilon)Uv = \varepsilon e,$$

where $\phi(\varepsilon) = (\frac{\varepsilon}{1-\varepsilon})^p$. Define $w = (1-\varepsilon)v$. The above system can be finally written as

$$\bar{\mathcal{H}}(u,w,\varepsilon) := \begin{pmatrix} Uw - \varepsilon e \\ w - (1-\varepsilon)(Mu+q+\phi(\varepsilon)u) \end{pmatrix} = 0, \qquad (u,w) > 0. \tag{2.6}$$

Noting that $p \in (0,1)$, we have $\varepsilon/\phi(\varepsilon) \to 0$ as $\varepsilon \to 0$. Since the matrix M is a monotone matrix, it must be a P_0 matrix or a P_* matrix. By Theorem 4.2(a) or Theorem 5.2(a) in [35] (but applied to a monotone LCP), we conclude that the above system (2.6) has a unique solution $(u(\varepsilon), w(\varepsilon))$ for each given $\varepsilon > 0$. The result (i) is proved.

To see that (ii) holds, we first note that if the path $\{(x(\mu), y(\mu), s(\mu), z(\mu)) : \mu \in (0, \hat{\mu}]\}$ is bounded, taking $\mu \to 0$ in system (2.4), we see that any accumulation point of the path is a solution to KKT system (1.4), and thus it is a solution to the linear program. Conversely, assume that the linear program is solvable. This is equivalent to saying that the LCP (1.5) is solvable. Notice that (2.4) can be written as (2.5). It follows from Theorem 5.1(b) in [35] that the path $\{(x(\mu), y(\mu), s(\mu), z(\mu)) : \mu \in (0, \hat{\mu}]\}$ is bounded. Result (ii) holds. Since the solutions of LCP (1.5) are the same as the solutions of the primal and the dual programs (1.1) and (1.2), result (iii) is an immediate consequence of Theorem 5.2 in [35]. □

From the above result, we obtain the following characterization of the least 2-norm solution of a linear program.

COROLLARY 2.2. *(x^*, y^*) is the least 2-norm solution pair to the primal and the dual problems if and only if it is the unique limiting point of the regularized central path as $\mu \to 0$. Equivalently, problem (1.1) has no optimal solution, i.e., problem (1.1) is unsolvable, if and only if the regularized central path is divergent to infinity as $\mu \to 0$.*

3. Algorithm. Our algorithm can tackle both solvable and unsolvable linear programming problems. For simplicity, however, we consider first the solvable problems. The general case, including unsolvable problems, is treated in section 5.

For a fixed scalar $p \in (0,1)$, we denote $\mathcal{F}_\mu : R^{2(n+m)} \to R^{2(n+m)}$ by

$$\mathcal{F}_\mu(x,y,s,z) = \begin{pmatrix} Xs - \mu e \\ Yz - \mu e \\ s + A^T y - c - \mu^p x \\ z - Ax + b - \mu^p y \end{pmatrix}. \tag{3.1}$$

Note that the regularized central path is given by the following system:

$$\mathcal{F}_\mu(x,y,s,z) = 0, \qquad (x,y,s,z) > 0.$$

We also note that the vector (x^*, y^*, s^*, z^*) is a solution to the KKT system (1.4) if and only if it satisfies

$$\mathcal{F}_0(x^*, y^*, s^*, z^*) = 0, \qquad (x^*, y^*, s^*, z^*) \geq 0.$$

To give a path-following algorithm, we employ the following set as a neighborhood of the regularized central path:

$$\mathcal{N}_\beta(\mu) := \{(x,y,s,z) > 0 : \ \|\mathcal{F}_\mu(x,y,s,z)\|_\infty \le \beta\mu\},$$

where $\beta \in (0,1)$ is a fixed scalar. From a starting point $(x^0,y^0,s^0,z^0) > 0$, the purpose of our path-following algorithm is to generate a positive sequence (x^k,y^k,s^k,z^k) confined in the above neighborhood. This sequence converges to a solution of the problem. In each step of the algorithm, only one linear algebraic equation is solved, and the Armijo-type line search is used to determine the stepsize. While the iteration of this algorithm proceeds in the positive orthant, i.e., all the iterates maintain positivity, the iterates are not necessarily interior points of the problem. In fact, this algorithm does not require that the problem possess an interior point, and thus it does not belong to the class of central path-based interior-point algorithms.

ALGORITHM 3.1.

Step 1 (Initial step). Let $\beta \in (0,1)$ be a positive scalar. Assign scalars α_1, α_2, and σ in (0,1). Select $(x^0,y^0,s^0,z^0) > 0$ and $\mu^0 \in (0,\infty)$ such that $(x^0,y^0,s^0,z^0) \in \mathcal{N}_\beta(\mu^0)$.

Step 2 (Centering step). If $\mathcal{F}_{\mu^k}(x^k,y^k,s^k,z^k) = 0$, set

$$(x^{k+1},y^{k+1},s^{k+1},z^{k+1}) = (x^k,y^k,s^k,z^k)$$

and go to Step 3. Otherwise, let $(\Delta x^k, \Delta y^k)$ be the solution to the following equation:

$$\begin{aligned} &\begin{bmatrix} S^k + (\mu^k)^p X^k & -X^k A^T \\ Y^k A & Z^k + (\mu^k)^p Y^k \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \\ &= \begin{bmatrix} \mu^k e - X^k s^k \\ \mu^k e - Y^k z^k \end{bmatrix} - \begin{bmatrix} X^k(-s^k + (\mu^k)^p x^k - A^T y^k + c) \\ Y^k(-z^k + Ax^k + (\mu^k)^p y^k - b) \end{bmatrix}. \end{aligned} \tag{3.2}$$

Then, set

$$\begin{bmatrix} \Delta s^k \\ \Delta z^k \end{bmatrix} = \begin{bmatrix} (\mu^k)^p I & -A^T \\ A & (\mu^k)^p I \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta y^k \end{bmatrix} + \begin{bmatrix} -s^k + (\mu^k)^p x^k - A^T y^k + c \\ -z^k + Ax^k + (\mu^k)^p y^k - b \end{bmatrix}. \tag{3.3}$$

Let

$$\begin{aligned} \bar\alpha = \arg\max\{\alpha > 0 : \ & x^k + \lambda\Delta x^k > 0, \ y^k + \lambda\Delta y^k > 0, \ s^k + \lambda\Delta z^k > 0, \\ & z^k + \lambda\Delta z^k > 0 \text{ for all } \lambda \in (0,\alpha]\}. \end{aligned}$$

Let λ_k be the maximum among the values of $\bar\alpha, \alpha_1\bar\alpha, \alpha_1^2\bar\alpha, \ldots$ such that

$$\begin{aligned} &\|\mathcal{F}_{\mu^k}(x^k + \lambda_k\Delta x^k, y^k + \lambda_k\Delta y^k, s^k + \lambda_k\Delta s^k, z^k + \lambda_k\Delta z^k)\|_\infty \\ &\le (1 - \sigma\lambda_k)\|\mathcal{F}_{\mu^k}(x^k,y^k,s^k,z^k)\|_\infty. \end{aligned} \tag{3.4}$$

Set

$$(x^{k+1},y^{k+1},s^{k+1},z^{k+1}) = (x^k,y^k,s^k,z^k) + \lambda_k(\Delta x^k, \Delta y^k, \Delta s^k, \Delta z^k)$$

and go to Step 3.

Step 3 (Reduction step for μ). Let γ^k be the maximum among the values of $\alpha_2, \alpha_2^2, \ldots$ such that

$$(x^{k+1},y^{k+1},s^{k+1},z^{k+1}) \in \mathcal{N}_\beta((1-\gamma^k)\mu^k),$$

i.e.,

$$\|\mathcal{F}_{(1-\gamma^k)\mu^k}(x^{k+1}, y^{k+1}, s^{k+1}, z^{k+1})\|_\infty \le \beta(1-\gamma^k)\mu^k.$$

Set $\mu^{k+1} := (1-\gamma^k)\mu^k$ and go to Step 2.

Remark 3.1. In numerical implementation, the initial points and some stopping criterion are needed. For the above algorithm, we may use $\|\mathcal{F}_0(x^k, y^k, s^k, z^k)\|_\infty < \varepsilon$ or $\mu^k < \varepsilon$ as the stopping criterion, where ε is a termination tolerance. The initial point for the above algorithm can be constructed without any additional cost. For instance, a practical initial step proceeds as follows.

Initial step: *Let* $(x^0, y^0) = e$ $(\in R^{n+m})$. *Choose* μ^0 *such that*

$$\mu^0 > \max\left\{1, \left\| \begin{pmatrix} A^T y^0 - c \\ -Ax^0 + b \end{pmatrix} \right\|_\infty \right\}.$$

Let $(s^0, z^0) = (\mu^0)^p e$ $(\in R^{n+m})$, *and let*

$$\eta := \frac{\|\mathcal{F}_{\mu^0}(x^0, y^0, s^0, z^0)\|_\infty}{\mu^0}.$$

Then, assign $\beta \in [\eta, 1)$.

From the above choice, by (3.1) we see that

$$\|\mathcal{F}_{\mu^0}(x^0, y^0, s^0, z^0)\|_\infty = \max\left\{ \left|\mu^0 - (\mu^0)^p\right|, \left\| \begin{pmatrix} A^T y^0 - c \\ -Ax^0 + b \end{pmatrix} \right\|_\infty \right\}.$$

By the choice of μ^0, it follows that $0 < \eta < 1$. Thus, $\eta \le \beta < 1$ and $(x^0, y^0, s^0, z^0) \in \mathcal{N}_\beta(\mu^0)$.

Remark 3.2. We now point out that, at the current point $(x^k, y^k, s^k, z^k) > 0$, the vector $(\Delta x^k, \Delta y^k, \Delta s^k, \Delta z^k)$ determined by systems (3.2) and (3.3) is unique. In fact, it is easy to see that $(\Delta x^k, \Delta y^k, \Delta s^k, \Delta z^k)$ is a solution to systems (3.2) and (3.3) if and only if it is a solution to the following system:

$$\left\{ \begin{array}{rcl} S^k \Delta x + X^k \Delta s &=& \mu^k e - X^k s^k, \\ Z^k \Delta y + Y^k \Delta z &=& \mu^k e - Y^k z^k, \\ \Delta s - (\mu^k)^p \Delta x + A^T \Delta y &=& -s^k - A^T y^k + (\mu^k)^p x^k + c, \\ \Delta z - A\Delta x + (\mu^k)^p \Delta y &=& -z^k + Ax^k + (\mu^k)^p y^k - b, \end{array} \right. \tag{3.5}$$

which is a $2(m+n)$-dimensional linear system. Notice that the Jacobian matrix of $\mathcal{F}_{\mu^k}(x, y, s, z)$ at $(x^k, y^k, s^k, z^k) > 0$ is given by

$$\nabla \mathcal{F}_{\mu^k}(x^k, y^k, s^k, z^k) = \begin{bmatrix} S^k & O & X^k & O \\ O & Z^k & O & Y^k \\ -(\mu^k)^p I & A^T & I & O \\ -A & -(\mu^k)^p I & O & I \end{bmatrix}. \tag{3.6}$$

System (3.5) coincides with the following:

$$\mathcal{F}_{\mu^k}(x^k, y^k, s^k, z^k) + \nabla \mathcal{F}_{\mu^k}(x^k, y^k, s^k, z^k)(\Delta x, \Delta y, \Delta s, \Delta z) = 0. \tag{3.7}$$

Hence, the direction $(\Delta x, \Delta y, \Delta s, \Delta z)$ is actually the Newton direction determined by (3.7). Since the matrix

$$\begin{bmatrix} (\mu^k)^p I & -A^T \\ A & (\mu^k)^p I \end{bmatrix}$$

is positive semidefinite for any $\mu^k > 0$ at the positive point (x^k, y^k, s^k, z^k) at which the Jacobian matrix $\nabla \mathcal{F}_{\mu^k}(x^k, y^k, s^k, z^k)$ given by (3.6) is nonsingular. This fact follows from Lemma 5.4 in Kojima, Megiddo, and Noma [9]. Thus system (3.7) has a unique solution, and hence systems (3.2) and (3.3) have a unique solution. This can also be explained another way. In fact, at $(x^k, y^k, s^k, z^k) > 0$, it is easy to verify that the nonsingularity of the matrix $\nabla \mathcal{F}_{\mu^k}(x^k, y^k, s^k, z^k)$ implies the nonsingularity of the matrix

$$\begin{bmatrix} S^k + (\mu^k)^p X^k & -X^k A^T \\ Y^k A & Z^k + (\mu^k)^p Y^k \end{bmatrix}.$$

While system (3.2) together with (3.3) is equivalent to system (3.5) or (3.7), we choose to solve system (3.2), since it has lower dimension than (3.5).

4. Global convergence. In this section, we show that whenever the solution set is nonempty, the iterates $\{(x^k, y^k)\}$ generated by Algorithm 3.1 converge to the least 2-norm solutions of the primal and the dual linear programs. We first show that the algorithm is well defined.

LEMMA 4.1. *Algorithm* 3.1 *is well defined. The sequence* $\{\mu^k\}$ *is monotonically decreasing, and* $(x^k, y^k, s^k, z^k) \in \mathcal{N}_\beta(\mu^k)$ *for all* $k \geq 0$.

Proof. We verify that each step of the algorithm is well defined. By Remark 3.1, the first step is well defined. The starting point satisfies

$$(x^0, y^0, s^0, z^0) > 0, \qquad (x^0, y^0, s^0, z^0) \in \mathcal{N}_\beta(\mu^0).$$

By induction, we now assume that

$$(x^k, y^k, s^k, z^k) > 0, \qquad (x^k, y^k, s^k, z^k) \in \mathcal{N}_\beta(\mu^k).$$

We show that the next iterate $(x^{k+1}, y^{k+1}, s^{k+1}, z^{k+1})$ generated by the algorithm still maintains positivity and satisfies the condition $(x^{k+1}, y^{k+1}, s^{k+1}, z^{k+1}) \in \mathcal{N}_\beta(\mu^{k+1})$. By the positivity of (x^k, y^k, s^k, z^k), from Remark 3.2, the system defined by (3.2) and (3.3) has a unique solution, and the Newton direction $(\Delta x^k, \Delta y^k, \Delta s^k, \Delta z^k)$ is a descent direction of the function $\|\mathcal{F}_{\mu^k}(x, y, s, z)\|_\infty$ at the current point $(x^k, y^k, s^k, z^k) > 0$. Thus, the line search rule (3.4) is well defined, and hence Step 2 is well defined. Since

$$\|\mathcal{F}_{\mu^k}(x^k, y^k, s^k, z^k)\|_\infty \leq \beta \mu^k \text{ and } 1 - \sigma\lambda_k < 1,$$

from (3.4) we have

$$\|\mathcal{F}_{\mu^k}(x^k + \lambda_k \Delta x^k, y^k + \lambda_k \Delta y^k, s^k + \lambda_k \Delta s^k, z^k + \lambda_k \Delta z^k)\|_\infty \leq \beta \mu^k,$$

that is,

$$\|\mathcal{F}_{\mu^k}(x^{k+1}, y^{k+1}, s^{k+1}, z^{k+1})\|_\infty \leq \beta \mu^k,$$

which implies that

$$\|X^{k+1} s^{k+1} - \mu^k e\|_\infty \leq \beta \mu^k, \qquad \|Y^{k+1} z^{k+1} - \mu^k e\|_\infty \leq \beta \mu^k.$$

By the choice of $\bar{\alpha}$ and λ_k, we see that $(x^{k+1}, y^{k+1}, s^{k+1}, z^{k+1})$ is nonnegative. Combining this fact and the above inequalities, where $0 < \beta < 1$, we conclude that the next iterate $(x^{k+1}, y^{k+1}, s^{k+1}, z^{k+1})$ must be positive.

We now show that Step 3 is well defined, and hence the next iterate is contained in the set $\mathcal{N}_\beta(\mu^{k+1})$. There are two possible cases.

Case 1. $\mathcal{F}_{\mu^k}(x^k, y^k, s^k, z^k) = 0$. According to the construction of the algorithm, $(x^{k+1}, y^{k+1}, s^{k+1}, z^{k+1}) = (x^k, y^k, s^k, z^k)$. By continuity, there is a γ^k determined by Step 3 such that

$$\|\mathcal{F}_{(1-\gamma^k)\mu^k}(x^{k+1}, y^{k+1}, s^{k+1}, z^{k+1})\|_\infty \le \beta(1-\gamma^k)\mu^k.$$

Thus, by setting $\mu^{k+1} = (1-\gamma^k)\mu^k$, we have $(x^{k+1}, y^{k+1}, s^{k+1}, z^{k+1}) \in \mathcal{N}_\beta(\mu^{k+1})$.

Case 2. $\mathcal{F}_{\mu^k}(x^k, y^k, s^k, z^k) \neq 0$. In this case, the next point $(x^{k+1}, y^{k+1}, s^{k+1}, z^{k+1})$ is determined by (3.4). We now show that $(x^{k+1}, y^{k+1}, s^{k+1}, z^{k+1}) \in \mathcal{N}_\beta(\mu^{k+1})$ still holds. For any $(x, y, s, z) > 0$ and $t_2 \ge t_1 \ge 0$, it is easy to verify that

$$\|\mathcal{F}_{t_1}(x, y, s, z) - \mathcal{F}_{t_2}(x, y, s, z)\|_\infty \le t_2 - t_1 + (t_2^p - t_1^p)\|(x, y)\|_\infty.$$

Thus, by (3.4) and the above inequality, we have

$$\begin{aligned}
&\|\mathcal{F}_{(1-\gamma)\mu^k}(x^{k+1}, y^{k+1}, s^{k+1}, z^{k+1})\|_\infty \\
&\le \|\mathcal{F}_{(1-\gamma)\mu^k}(x^{k+1}, y^{k+1}, s^{k+1}, z^{k+1}) - \mathcal{F}_{\mu^k}(x^{k+1}, y^{k+1}, s^{k+1}, z^{k+1})\|_\infty \\
&\quad + \|\mathcal{F}_{\mu^k}(x^{k+1}, y^{k+1}, s^{k+1}, z^{k+1})\|_\infty \\
&\le \gamma\mu^k + (\mu^k)^p[1-(1-\gamma)^p]\|(x^{k+1}, y^{k+1})\|_\infty + (1-\sigma\lambda_k)\|\mathcal{F}_{\mu^k}(x^k, y^k, s^k, z^k)\|_\infty \\
&\le \gamma\mu^k + (\mu^k)^p[1-(1-\gamma)^p]\|(x^{k+1}, y^{k+1})\|_\infty + (1-\sigma\lambda_k)\beta\mu^k \\
&= \left[\frac{\gamma + (\mu^k)^{p-1}[1-(1-\gamma)^p]\|(x^{k+1}, y^{k+1})\|_\infty}{(1-\gamma)\beta} + \frac{(1-\sigma\lambda_k)}{1-\gamma}\right]\beta(1-\gamma)\mu^k.
\end{aligned}$$

Since $1-\sigma\lambda_k < 1$, there is a positive scalar $\hat{\gamma} > 0$ such that for all $\gamma \in (0, \hat{\gamma}]$ the term in the above bracket is less than one. Thus, for all sufficiently small $\gamma > 0$ we have

$$\|\mathcal{F}_{(1-\gamma)\mu^k}(x^{k+1}, y^{k+1}, s^{k+1}, z^{k+1})\|_\infty \le \beta(1-\gamma)\mu^k.$$

Step 3 is well defined. Of course, the sequence $\{\mu^k\}$ is monotonically decreasing since $\mu^{k+1} = (1-\gamma^k)\mu^k$. □

By Lemma 4.1, the sequence $(x^k, y^k, s^k, z^k) \in \mathcal{N}_\beta(\mu^k)$ for all k, i.e.,

$$\|\mathcal{F}_{\mu^k}(x^k, y^k, s^k, z^k)\|_\infty \le \beta\mu^k \quad \text{and} \quad (x^k, y^k, s^k, z^k) > 0. \tag{4.1}$$

We employ auxiliary sequences $(u^k, v^k, w^k, q^k) \in R^{2(n+m)}$ defined by

$$(u^k, v^k, w^k, q^k) = \frac{1}{\mu^k}\mathcal{F}_{\mu^k}(x^k, y^k, s^k, z^k). \tag{4.2}$$

Clearly, the above sequence $\{(u^k, v^k, w^k, q^k)\}$ is uniformly bounded. Indeed, combining (4.1) and (4.2) yields $\|(u^k, v^k, w^k, q^k)\|_\infty \le \beta$. Equation (4.2) can be written as

$$X^k s^k = \mu^k(e + u^k), \tag{4.3}$$

$$Y^k z^k = \mu^k(e + v^k), \tag{4.4}$$

$$s^k = -A^T y^k + c + (\mu^k)^p x^k + \mu^k w^k, \tag{4.5}$$

$$z^k = Ax^k - b + (\mu^k)^p y^k + \mu^k q^k. \tag{4.6}$$

These relations play a key role in the remainder of our analysis. We now prove the main result of this section.

THEOREM 4.2. *Assume that the solution set of linear program* (1.1) *is nonempty. The sequence* (x^k, y^k, s^k, z^k) *generated by Algorithm* 3.1 *converges to* $(\hat{x}, \hat{y}, \hat{s}, \hat{z})$*, where* $\hat{x}$ *is the least* 2*-norm solution of the primal linear program* (1.1) *and* $\hat{y}$ *is the least* 2*-norm solution of the dual problem* (1.2)*.*

Proof. We prove this result in the following three steps:

(i) If the solution set is nonempty, then the iterative sequence (x^k, y^k, s^k, z^k) generated by the algorithm is bounded.

(ii) $\mu^k \to 0$ and $\|\mathcal{F}_{\mu^k}(x^k, y^k, s^k, z^k)\|_\infty \to 0$. Thus, every accumulation point of the iterative sequence is a solution to the linear program.

(iii) The accumulation point is unique and must be the least 2-norm solution of the linear program.

We now prove (i). Let x^* be an arbitrary optimal solution of (1.1) and y^* be an arbitrary optimal solution of its dual problem (1.2). Let (s^*, z^*) be given by

$$\begin{bmatrix} s^* \\ z^* \end{bmatrix} = \begin{bmatrix} O & -A^T \\ A & O \end{bmatrix} \begin{bmatrix} x^* \\ y^* \end{bmatrix} + \begin{bmatrix} c \\ -b \end{bmatrix}. \tag{4.7}$$

Then, it is easy to see that $(x^*, y^*, s^*, z^*) \geq 0$, $(x^*)^T s^* = 0$, and $(y^*)^T z^* = 0$. That is, (x^*, y^*, s^*, z^*) satisfies the KKT system (1.4). It follows from (4.3) and (4.4) that

$$(x^k)^T s^k = \mu^k (n + e^T u^k), \qquad (y^k)^T z^k = \mu^k (m + e^T v^k). \tag{4.8}$$

Notice that for any $(x, y) \in R^{n+m}$ we have

$$\begin{bmatrix} x \\ y \end{bmatrix}^T \begin{bmatrix} (\mu^k)^p I & -A^T \\ A & (\mu^k)^p I \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = (\mu^k)^p \left\| \begin{bmatrix} x \\ y \end{bmatrix} \right\|_2^2. \tag{4.9}$$

By the positivity of (x^k, y^k, s^k, z^k), (4.5), (4.6), (4.9), (4.8), and (4.7), we have

$$\begin{aligned}
0 &\leq \begin{bmatrix} x^* \\ y^* \end{bmatrix}^T \begin{bmatrix} s^k \\ z^k \end{bmatrix} + \begin{bmatrix} s^* \\ z^* \end{bmatrix}^T \begin{bmatrix} x^k \\ y^k \end{bmatrix} \\
&= \begin{bmatrix} x^* - x^k \\ y^* - y^k \end{bmatrix}^T \begin{bmatrix} s^k \\ z^k \end{bmatrix} + \begin{bmatrix} s^* \\ z^* \end{bmatrix}^T \begin{bmatrix} x^k \\ y^k \end{bmatrix} + \begin{bmatrix} x^k \\ y^k \end{bmatrix}^T \begin{bmatrix} s^k \\ z^k \end{bmatrix} \\
&= \begin{bmatrix} x^* - x^k \\ y^* - y^k \end{bmatrix}^T \left(\begin{bmatrix} (\mu^k)^p I & -A^T \\ A & (\mu^k)^p I \end{bmatrix} \begin{bmatrix} x^k \\ y^k \end{bmatrix} + \begin{bmatrix} c \\ -b \end{bmatrix} + \mu^k \begin{bmatrix} w^k \\ q^k \end{bmatrix} \right) \\
&\quad + \begin{bmatrix} s^* \\ z^* \end{bmatrix}^T \begin{bmatrix} x^k \\ y^k \end{bmatrix} + \begin{bmatrix} x^k \\ y^k \end{bmatrix}^T \begin{bmatrix} s^k \\ z^k \end{bmatrix} \\
&= - \begin{bmatrix} x^k - x^* \\ y^k - x^* \end{bmatrix}^T \begin{bmatrix} (\mu^k)^p I & -A^T \\ A & (\mu^k)^p I \end{bmatrix} \begin{bmatrix} x^k - x^* \\ y^k - x^* \end{bmatrix} \\
&\quad - \begin{bmatrix} x^k - x^* \\ y^k - y^* \end{bmatrix}^T \left(\begin{bmatrix} (\mu^k)^p I & -A^T \\ A & (\mu^k)^p I \end{bmatrix} \begin{bmatrix} x^* \\ y^* \end{bmatrix} + \begin{bmatrix} c \\ -b \end{bmatrix} + \mu^k \begin{bmatrix} w^k \\ q^k \end{bmatrix} \right) \\
&\quad + \begin{bmatrix} s^* \\ z^* \end{bmatrix}^T \begin{bmatrix} x^k \\ y^k \end{bmatrix} + \begin{bmatrix} x^k \\ y^k \end{bmatrix}^T \begin{bmatrix} s^k \\ z^k \end{bmatrix} \\
&= -(\mu^k)^p \left\| \begin{bmatrix} x^k - x^* \\ y^k - y^* \end{bmatrix} \right\|_2^2 + \begin{bmatrix} s^* \\ z^* \end{bmatrix}^T \begin{bmatrix} x^k \\ y^k \end{bmatrix} + \begin{bmatrix} x^k \\ y^k \end{bmatrix}^T \begin{bmatrix} s^k \\ z^k \end{bmatrix}
\end{aligned}$$

$$
\begin{aligned}
&- \begin{bmatrix} x^k - x^* \\ y^k - y^* \end{bmatrix}^T \left((\mu^k)^p \begin{bmatrix} x^* \\ y^* \end{bmatrix} + \begin{bmatrix} s^* \\ z^* \end{bmatrix} + \mu^k \begin{bmatrix} w^k \\ q^k \end{bmatrix} \right) \\
&= -(\mu^k)^p \left\| \begin{bmatrix} x^k - x^* \\ y^k - y^* \end{bmatrix} \right\|_2^2 - (\mu^k)^p \begin{bmatrix} x^k - x^* \\ y^k - y^* \end{bmatrix}^T \begin{bmatrix} x^* + (\mu^k)^{1-p} w^k \\ y^* + (\mu^k)^{1-p} q^k \end{bmatrix} \\
&\quad + \begin{bmatrix} x^k \\ y^k \end{bmatrix}^T \begin{bmatrix} s^k \\ z^k \end{bmatrix} \\
&= -(\mu^k)^p \left\| \begin{bmatrix} x^k - x^* \\ y^k - y^* \end{bmatrix} \right\|_2^2 - (\mu^k)^p \begin{bmatrix} x^k - x^* \\ y^k - y^* \end{bmatrix}^T \begin{bmatrix} x^* + (\mu^k)^{1-p} w^k \\ y^* + (\mu^k)^{1-p} q^k \end{bmatrix} \\
&\quad + \mu^k (m + n + e^T u^k + e^T v^k). \qquad (4.10)
\end{aligned}
$$

Dividing both sides of the above by $(\mu^k)^p$, we have

$$
\begin{aligned}
\left\| \begin{bmatrix} x^k - x^* \\ y^k - y^* \end{bmatrix} \right\|_2^2 \le{}& \left\| \begin{bmatrix} x^k - x^* \\ y^k - y^* \end{bmatrix} \right\|_2 \left\| \begin{bmatrix} x^* + (\mu^k)^{1-p} w^k \\ y^* + (\mu^k)^{1-p} q^k \end{bmatrix} \right\|_2 \\
&+ \left(\mu^k\right)^{1-p} (m + n + e^T u^k + e^T v^k) . \qquad (4.11)
\end{aligned}
$$

Since $\mu^k \le \mu^0$ and (u^k, v^k, w^k, q^k) is uniformly bounded, the boundedness of the iterative sequence (x^k, y^k, s^k, z^k) follows from the above inequality. Part (i) is now proven.

We now prove part (ii). Since all iterates are confined in $\mathcal{N}_\beta(\mu^k)$, it implies that (4.1) holds for all k. Thus, to show $\|\mathcal{F}_{\mu^k}(x^k, y^k, s^k, z^k)\|_\infty \to 0$, it suffices to show that $\mu^k \to 0$. In fact, μ^k is monotonically decreasing since $\mu^{k+1} = (1 - \gamma^k)\mu^k$. Thus, there exists a scalar $\hat{\mu} \ge 0$ such that $\mu^k \to \hat{\mu}$. By (i), the sequence (x^k, y^k, s^k, z^k) is bounded. Without loss of generality, we may assume that $(x^k, y^k, s^k, z^k) \to (\hat{x}, \hat{y}, \hat{s}, \hat{z})$. Taking the limit in (4.1), we have that

$$
\|\mathcal{F}_{\hat{\mu}}(\hat{x}, \hat{y}, \hat{s}, \hat{z})\|_\infty \le \beta \hat{\mu}, \qquad (\hat{x}, \hat{y}, \hat{s}, \hat{z}) \ge 0. \qquad (4.12)
$$

We assume to the contrary that $\hat{\mu} \ne 0$, i.e., $\hat{\mu} > 0$. We now derive a contradiction. The fact that $\mu^{k+1} = (1 - \gamma^k)\mu^k$, combined with $\mu^k \to \hat{\mu} > 0$, implies that $\gamma^k \to 0$ as $k \to \infty$.

We deduce from (4.12) that

$$
\|\hat{X}\hat{s} - \hat{\mu}e\|_\infty \le \beta\hat{\mu}, \qquad \|\hat{Y}\hat{z} - \hat{\mu}e\|_\infty \le \beta\hat{\mu}.
$$

Since $0 < \beta < 1$ and $(\hat{x}, \hat{y}, \hat{s}, \hat{z}) \ge 0$, the above inequality implies that $(\hat{x}, \hat{y}, \hat{s}, \hat{z}) > 0$. Thus, by Remark 3.2, the Jacobian $\nabla \mathcal{F}_{\hat{\mu}}(\hat{x}, \hat{y}, \hat{s}, \hat{z})$ is nonsingular, and hence the matrix

$$
\begin{bmatrix} \hat{S} + \hat{\mu}^p \hat{X} & -\hat{X}A^T \\ \hat{Y}A & \hat{Z} + \hat{\mu}^p \hat{Y} \end{bmatrix}
$$

is nonsingular. Therefore, the following system has a unique solution, denoted by $(\Delta\hat{x}, \Delta\hat{y}, \Delta\hat{s}, \Delta\hat{z})$:

$$
\begin{bmatrix} \hat{S} + \hat{\mu}^p \hat{X} & -\hat{X}A^T \\ \hat{Y}A & \hat{Z} + \hat{\mu}^p \hat{Y} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \hat{\mu}e - \hat{X}\hat{s} \\ \hat{\mu}e - \hat{Y}\hat{z} \end{bmatrix} - \begin{bmatrix} \hat{X}(-\hat{s} + \hat{\mu}^p \hat{x} - A^T \hat{y} + c) \\ \hat{Y}(-\hat{z} + A\hat{x} + \hat{\mu}^p \hat{y} - b) \end{bmatrix},
$$

$$
\begin{bmatrix} \Delta s \\ \Delta z \end{bmatrix} = \begin{bmatrix} \hat{\mu}^p I & -A^T \\ A & \hat{\mu}^p I \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} + \begin{bmatrix} -\hat{s} + \hat{\mu}^p \hat{x} - A^T \hat{y} + c \\ -\hat{z} + A\hat{x} + \hat{\mu}^p \hat{y} - b \end{bmatrix}.
$$

By Remark 3.2, this is equivalent to

$$\mathcal{F}_{\hat{\mu}}(\hat{x},\hat{y},\hat{s},\hat{z}) + \nabla\mathcal{F}_{\hat{\mu}}(\hat{x},\hat{y},\hat{s},\hat{z})(\Delta\hat{x},\Delta\hat{y},\Delta\hat{s},\Delta\hat{z}) = 0,$$

which implies that $(\Delta\hat{x},\Delta\hat{y},\Delta\hat{s},\Delta\hat{z})$ is a Newton descent direction of $\|\mathcal{F}_{\hat{\mu}}(x,y,s,z)\|_\infty$ at $(\hat{x},\hat{y},\hat{s},\hat{z})$. Thus the linear search stepsize $\hat{\lambda}$ in (3.4) and $\hat{\gamma}$ in Step 3 of Algorithm 3.1 are both bounded below by a positive constant. By continuity, it follows that

$$(\Delta x^k,\Delta y^k,\Delta s^k,\Delta z^k,\mu^k,\lambda^k,\gamma^k) \to (\Delta\hat{x},\Delta\hat{y},\Delta\hat{s},\Delta\hat{z},\hat{\mu},\hat{\lambda},\hat{\gamma}).$$

In particular, $\gamma^k \to \hat{\gamma} > 0$, which contradicts $\gamma^k \to 0$. This contradiction shows that μ^k must converge to zero, and thus $\|\mathcal{F}_{\mu^k}(x^k,y^k,s^k,z^k)\| \to 0$ as $k \to \infty$. Therefore, for any accumulation point $(\hat{x},\hat{y},\hat{s},\hat{z})$, by continuity we have $\|\mathcal{F}_0(\hat{x},\hat{y},\hat{s},\hat{z})\| = 0$, which implies that $(\hat{x},\hat{y})$ is a solution pair to the primal and the dual linear programs.

Finally, we show that the accumulation point of the iterates is the unique least 2-norm solution. From (4.10), we have

$$\left\|\begin{bmatrix} x^k - x^* \\ y^k - y^* \end{bmatrix}\right\|_2^2 \le -\begin{bmatrix} x^k - x^* \\ y^k - y^* \end{bmatrix}^T \begin{bmatrix} x^* + (\mu^k)^{1-p}w^k \\ y^* + (\mu^k)^{1-p}q^k \end{bmatrix} + (\mu^k)^{1-p}(m+n+e^Tu^k+e^Tv^k).$$

Let $(\hat{x},\hat{y},\hat{s},\hat{z})$ be an arbitrary accumulation point of the iterates. Notice that $p \in (0,1)$, $\mu^k \to 0$, and (u^k,v^k,w^k,q^k) is bounded. Taking the limit in the above inequality, we have

$$\left\|\begin{bmatrix} \hat{x} - x^* \\ \hat{y} - y^* \end{bmatrix}\right\|_2^2 \le -\begin{bmatrix} \hat{x} - x^* \\ \hat{y} - y^* \end{bmatrix}^T \begin{bmatrix} x^* \\ y^* \end{bmatrix},$$

which can be written as

$$\left\|\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix}\right\|_2^2 \le \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix}^T \begin{bmatrix} x^* \\ y^* \end{bmatrix} \le \left\|\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix}\right\|_2 \left\|\begin{bmatrix} x^* \\ y^* \end{bmatrix}\right\|_2.$$

Since (x^*,y^*) is an arbitrary solution pair of the primal and the dual, from the above inequality we deduce that $\hat{x}$ and $\hat{y}$ are the least 2-norm solutions of the primal and the dual, respectively. (In fact, substituting (x^*,y^*) by $(x^*,\hat{y})$ and $(\hat{x},y^*)$, respectively, we see that the above inequality implies that $\|(\hat{x},\hat{y})\|_2 \le \|(x^*,\hat{y})\|_2$ and $\|(\hat{x},\hat{y})\|_2 \le \|(\hat{x},y^*)\|_2$ for all primal and dual solutions x^* and y^*. The desired result follows.) ∎

5. Possibly unsolvable linear programs. We now consider a general linear program (1.1) which is possibly unsolvable. Let $\mathcal{R} : R_+^{n+m} \to R_+$ be a measure function for solvability of the problem (1.1), that is,

$$\mathcal{R}(x,y) = \|(A^Ty - c)_+\|_1 + \|[-(Ax-b)]_+\|_1 + (c^Tx - b^Ty)_+.$$

Clearly, the value of the above function can also be viewed as a KKT residual corresponding to an approximate solution (x,y) of the linear program. Notice that $(x^*,y^*) \ge 0$ is a solution to the primal and the dual (1.1) and (1.2) if and only if $\mathcal{R}(x^*,y^*) = 0$. Thus a linear program is equivalent to the following global minimization problem:

$$\min\{\mathcal{R}(x,y) : (x,y) \ge 0\}. \tag{5.1}$$

We may refer (5.1) to the problem of minimizing the 1-norm solvability of linear program (1.1). By a basic idea of Mangasarian [16], the above problem can be reformulated as a linear programming problem. Indeed, by introducing nonnegative variables $(s, z, t) \in R_+^{n+m+1}$, problem (5.1) can be equivalently transformed into the following linear program:

$$\begin{array}{ll} \min & e^T(s,z,t) \\ \text{s.t.} & A^T y - c \le s, \ -(Ax-b) \le z, \ c^T x - b^T y \le t, \\ & (x,y,s,z,t) \ge 0, \end{array} \tag{5.2}$$

where $e \in R^{n+m+1}$. This problem is always feasible. In fact, for any fixed $(x^0, y^0) \ge 0$, the vector $(x^0, y^0, s^0, z^0, t^0) \ge 0$ is feasible, provided that $(s^0, z^0, t^0) > 0$ is sufficiently large. Since the objective function is nonnegative, the above linear program is always solvable, and hence problem (5.1) has a global optimal solution. Let

$$c' := (0, 0, e) \in R^n \times R^m \times R^{n+m+1}, \qquad b' := (-c, b, 0) \in R^n \times R^m \times R,$$

$$A' := \begin{bmatrix} O & -A^T & O & I & O \\ A & O & I & O & O \\ -c^T & b^T & O & O & 1 \end{bmatrix}_{(n+m+1)\times 2(m+n)+1},$$

and $u = (x, y, z, s, t)$. Then, (5.2) can be written as

$$\min\{(c')^T u : A'u \ge b', u \ge 0\}.$$

Replacing (c, A, b) by (c', A', b') and applying Algorithm 3.1 to the above problem, we can obtain the unique least 2-norm solution $(x^*, y^*, s^*, z^*, t^*)$ of problem (5.2). We note that for any solution $(\hat{x}, \hat{y}, \hat{s}, \hat{z}, \hat{t})$ of (5.2), the following holds:

$$\hat{z} = [-(A\hat{x} - b)]_+, \qquad \hat{s} = (A^T\hat{y} - c)_+, \qquad \hat{t} = (c^T\hat{x} - b^T\hat{y})_+.$$

Thus, for any solution $(\hat{x}, \hat{y}, \hat{s}, \hat{z}, \hat{t})$ of (5.2) we have

$$\begin{aligned} &\|(x^*, \ y^*, \ [-(Ax^* - b)]_+, \ (A^T y^* - c)_+, \ (c^T x^* - b^T y^*)_+)\|_2 \\ &\le \|(\hat{x}, \ \hat{y}, \ [-(A\hat{x} - b)]_+, \ (A^T\hat{y} - c)_+, \ (c^T\hat{x} - b^T\hat{y})_+)\|_2. \end{aligned} \tag{5.3}$$

When linear program (1.1) or (1.2) is solvable, it is easy to see that any solution $(\hat{x}, \hat{y}, \hat{s}, \hat{z}, \hat{t})$ of (5.2) must satisfy that $\hat{s} = 0$, $\hat{z} = 0$, and $\hat{t} = 0$, and that $(\hat{x}, \hat{y})$ is a solution pair of the primal (1.1) and the dual (1.2). Conversely, if (x, y) is a solution pair to the primal and the dual, then $(x, y, 0, 0, 0)$ must be an optimal solution of problem (5.2). Thus, for solvable linear program (1.1), inequality (5.3) reduces to

$$\|(x^*, y^*, 0, 0, 0)\|_2 \le \|(\hat{x}, \hat{y}, 0, 0, 0)\|_2$$

for all solutions $(\hat{x}, \hat{y}, 0, 0, 0)$ of (5.2), which implies that x^* and y^* are the least 2-norm solutions of the primal (1.1) and the dual (1.2), respectively.

In summary, when applied to the linear program (5.2), Algorithm 3.1 is convergent whether (1.1) is solvable or not. For solvable problems, the algorithm will converge to $(x^*, y^*, 0, 0, 0)$, where x^* and y^* are the least 2-norm solutions of the primal and the dual problems (1.1) and (1.2); otherwise, Algorithm 3.1 converges to a point which gives a minimal KKT residual.

6. Numerical results. While the linear system (3.2) is $(m+n)$-dimensional, we now point out that this system can be further reduced so that, at each step, only an m- or n-dimensional linear system needs to be solved. In fact, (3.2) can be written as

$$(S^k + (\mu^k)^p X^k)\Delta x - X^k A^T \Delta y = \mu^k e - X^k \left((\mu^k)^p x^k - A^T y^k + c\right), \tag{6.1}$$

$$(Z^k + (\mu^k)^p Y^k)\Delta y + Y^k A \Delta x = \mu^k e - Y^k \left(Ax^k + (\mu^k)^p y^k - b\right). \tag{6.2}$$

When $m \geq n$, eliminating Δy leads to

$$\begin{aligned} M_k \Delta x = {} & \mu^k e - X^k \left((\mu^k)^p x^k - A^T y^k + c\right) \\ & + X^k A^T (Z^k + (\mu^k)^p Y^k)^{-1} \left[\mu^k e - Y^k \left(Ax^k + (\mu^k)^p y^k - b\right)\right], \end{aligned}$$

where M_k is an $n \times n$ matrix given by

$$M_k = S^k + (\mu^k)^p X^k + X^k A^T (Z^k + (\mu^k)^p Y^k)^{-1} Y^k A.$$

Thus, we can obtain Δx by solving the above system and then set

$$\Delta y = (Z^k + (\mu^k)^p Y^k)^{-1} \left[\mu^k e - Y^k \left(Ax^k + (\mu^k)^p y^k - b\right) - Y^k A \Delta x\right].$$

If $m \leq n$, in the same way, eliminating Δx from (6.1) and (6.2) yields

$$\begin{aligned} H_k \Delta y = {} & \mu^k e - Y^k \left((\mu^k)^p y^k + Ax^k - b\right) \\ & - Y^k A (S^k + (\mu^k)^p X^k)^{-1} \left[\mu^k e - X^k \left((\mu^k)^p x^k - A^T y^k + c\right)\right], \end{aligned}$$

where H_k is an $m \times m$ matrix given by

$$H_k = Z^k + (\mu^k)^p Y^k + Y^k A (S^k + (\mu^k)^p X^k)^{-1} X^k A^T.$$

Since system (3.2) has a unique solution, it follows that both M_k and H_k are nonsingular. Thus, at each step of Algorithm 3.1, we only need to factorize a matrix of size $\min(m,n) \times \min(m,n)$.

In numerical experiments, we took common parameters and starting points for all the test problems. Parameters were set as $p = 0.99$, $\sigma = 1e-5$, $\alpha_1 = 0.9$, and $\alpha_2 = 0.85$. The starting point (x^0, y^0, s^0, z^0) was set as in Remark 3.1, where μ^0 and β were given by

$$\mu^0 = \max\left\{1, \left\| \begin{pmatrix} A^T y^0 - c \\ -Ax^0 + b \end{pmatrix} \right\|_\infty \right\} + 1, \qquad \beta = \frac{\eta + 1}{2}.$$

Before stating our numerical results on some test problems, let us first see a very simple example with multiple solutions. Consider the following problem:

$$\min\{-x_1 - 2x_2 : x_1 + 2x_2 \leq 8,\ x_2 \leq 2,\ x_1, x_2 \geq 0\}.$$

It is easy to check that the solution set is $\{(x_1^*, x_2^*) = (4 + 4t, 2 - 2t) : 0 \leq t \leq 1\}$. Under a stopping rule of $\mu^k < 10^{-12}$, the following primal and dual solutions were obtained by the proposed algorithm:

$$\begin{aligned} x^* &= (4.0000047139881082,\ 1.9999976430060873), \\ y^* &= (0.9999999999886861,\ 1.6971276334429352e-11), \end{aligned}$$

TABLE 6.1

Name	Rows	Cols	Nonz	μ^k	$\|\mathcal{F}_0\|_\infty$	Objective values	CPU (secs)
beale	3	4	9	7.5e-11	2.0e-10	-1.25	.001
padberg	4	6	22	7.3e-09	1.8e-08	3.544147e-08	.001
refinery	22	14	63	9.6e-11	1.9e-09	-5.166833e+01	.08
william1	5	12	25	9.2e-11	1.6e-10	1.158471e-09	.002
william2	9	7	18	9.9e-11	3.3e-09	2.599999e+01	.05
william3	11	12	36	9.9e-10	1.3e-07	2.591899e+04	.4
afiro	35	32	117	9.9e-09	6.0e-06	-4.647531e+02	5.916
sc50a	70	48	182	8.9e-09	3.6e-06	-6.457507e+01	8.983
sc50b	70	48	170	9.9e-09	3.9e-06	-6.999999e+01	9.863
blend	117	83	789	9.9e-09	1.0e-06	-3.081215e+01	30.516
share2b	109	79	778	9.9e-09	3.7e-06	-4.157338e+02	37.600
sc105	150	103	402	9.8e-09	8.5e-06	-5.220206e+01	84.432
sc205	296	203	800	9.9e-09	2.8e-06	-5.220206e+01	325.632
scorpion	668	358	2526	9.8e-07	5.4e-05	1.878440e+03	> 500

with ∞-norm residual $\|\mathcal{F}_0(x^k, y^k, s^k, z^k)\|_\infty = 1.1314044977885658e - 11$. The corresponding objective value of the original problem is -8.0000000000002828. We note that (4, 2) and (1, 0) are exact least 2-norm solutions of the primal and the dual problems, respectively, and the exact optimal objective value is -8. This example shows that the proposed algorithm does locate the least 2-norm solution of the problem.

We now give a set of test examples and corresponding numerical results. We used $\mu^k \leq 10^{-10}$ or 10^{-8} as the stopping criterion for most of these test problems. All tests were carried out on a DEC Alpha V 4.0 machine. Results for 14 test problems are summarized in Table 6.1 and Table 6.2. The first problem was the well-known Beale's example and the second was Padberg's example [20, p. 60]. Both problems are cycling for simplex methods. The problem "refinery" can be found in [20]. The problems "william1," "william2," and "william3" were taken from [31] ($M = 50000$ was used in the problem "william3"). All other test problems here were taken from the collection of LP Data in NETLIB. In our code, all problems were transformed into the form of (1.1). To this end, all original inequalities "$\leq$" became "$\geq$" by multiplying both sides of inequalities by -1, and all equations were written equivalently as two inequalities. This preprocessing makes no change in the number of columns and keeps the sparsity of coefficient matrix A. However, the number of rows will be increased when the problem has equation constraints. The numbers of rows and nonzero (Nonz) entries of A in Table 6.1 are those resulting from this preprocessing. Under our stopping criterion, the computational optimal objective values, the values of μ^k, the ∞-norm residual $\|\mathcal{F}_0(x^k, y^k, s^k, z^k)\|_\infty$, and CPU time are listed in Table 6.1. The computational primal and dual least 2-norm solutions for these test problems are given in Table 6.2, where only the first six components are listed due to the space limitation.

From our results, we note that Algorithm 3.1, using the initial strategy in Remark 3.1, is efficient for small-scale linear programs. However, the convergence rate of the algorithm becomes slow as the dimension of problems increases. The main reason might be that the stepsize of Armijo-type linear searches may become smaller and smaller when iterates approach the least 2-norm solution. We also note that the matrices M_k and H_k are dense in general cases. Thus, when m and n are large, at each iteration a large and dense matrix needs to be factorized, which takes a certain amount of CPU time. Thus, the current version of the algorithm is not so efficient

for solving large-scale problems. Some modified versions of the algorithm are worth studying in the future in order to improve the convergence rate. A possible method is to use a certain approximate Newton step to accelerate the iteration, as we have done for nonlinear complementarity problems in [37].

TABLE 6.2

Name	Primal and dual least 2-norm solution (x^*, y^*)
beale	$x^* = (1.00000, 0.00000, 1.00000, 0.00000)$ $y^* = (0.00000, 1.50000, 1.25000)$
padberg	$x^* = (1.29177, 0.00000, 0.64588, 0.00000, 0.64588, 0.00000)$ $y^* = (0.91360, 0.91342, 0.91360, 0.91342)$
refinery	$x^* = (15.00000, 10.00000, 3.50000, 6.25000, 8.00000, 1.55000, \ldots)$ $y^* = (0.06333, 1.62166, 0.81166, 4.99246, 4.99246, 2.51578, \ldots)$
william1	$x^* = (0.00000, 0.00000, 0.00000, 0.00000, 0.00000, 0.00000, \ldots)$ $y^* = (0.00000, 0.00000, 0.00000, 0.00000, 0.00000)$
william2	$x^* = (0.00000, 4.16154, 17.00000, 7.00000, 17.00000, 22.00000, \ldots)$ $y^* = (0.00000, 0.00000, 1.00000, 0.00000, 1.00000, 1.00000, \ldots)$
william3	$x^* = (0.00000, 0.00000, 39.00000, 87.00000, 56.00000, 0.00000, \ldots)$ $y^* = (0.00000, 26.00000, 5.00000, 111.00732, 94.00864, 97.00838, \ldots)$
afiro	$x^* = (80.00000, 25.50000, 54.50000, 84.79999, 36.85030, 0.00000, \ldots)$ $y^* = (0.65036, 0.91201, 0.34477, 0.22857, 0.91201, 0.91201, \ldots)$
sc50a	$x^* = (0.00000, 16.56869, 64.57507, 64.57507, 64.57507, 0.00000, \ldots)$ $y^* = (0.00000, 0.13869, 0.91201, 0.81390, 0.85202, 0.78381, \ldots)$
sc50b	$x^* = (29.99999, 28.00000, 42.0000, 69.99999, 69.99999, 29.99999, \ldots)$ $y^* = (0.05836, 0.91201, 0.91201, 0.82870, 0.82871, 0.82871, \ldots)$
blend	$x^* = (20.94480, 10.17092, 11.24735, 2.98109, 0.65970, 0.47592, \ldots)$ $y^* = (0.21613, 0.22386, 0.26003, 0.26003, 0.25294, 0.25983, \ldots)$
share2b	$x^* = (1.95814, 2.02325, 0.00000, 0.00000, 0.00000, 0.00000, \ldots)$ $y^* = (0.12564, 0.00000, 0.00000, 0.00000, 0.00000, 0.33250, \ldots)$
sc105	$x^* = (0.00000, 10.84845, 52.20206, 52.20206, 52.20206, 0.00000, \ldots)$ $y^* = (0.00000, 0.16419, 0.91201, 0.79709, 0.84241, 0.76248, \ldots)$
sc205	$x^* = (0.00000, 10.84845, 52.20206, 52.20206, 52.20206, 0.00000, \ldots)$ $y^* = (0.00000, 0.16136, 0.91201, 0.79872, 0.84356, 0.76481, \ldots)$
scorpion	$x^* = (0.00871, 0.00211, 0.00023, 0.00452, 1.42494, 0.00250, \ldots)$ $y^* = (0.9293, 113.1449, 115.4149, 115.4149, 0.0000, 421.2979, \ldots)$

7. Conclusions. In this paper, we have introduced a new concept of a regularized central path for linear programs, which is different from the conventional central path. The regularized central path always exists for all linear programs, even if the linear program is unsolvable. If a linear program is solvable, the regularized central path converges, as the parameter μ tends to zero, to the unique least 2-norm solution of the linear program. As a result, we propose in this paper a regularized central path-based path-following algorithm for solving linear programming problems. This is a new alternative algorithm for locating the least 2-norm solution of a linear program. When applied to the equivalent problem (5.2), the iterative sequence generated by this algorithm is always convergent, whether or not the problem is solvable. If the primal problem is solvable, the limiting point of the sequence is the least 2-norm solution; otherwise, the limiting point gives a minimal KKT residual.

It should be pointed out that most of the existing algorithms for the least-norm solution of the linear program are akin to the canonical Tikhonov regularization method. One significance of the proposed algorithm is that it introduces the framework of interior-point methods into the canonical Tikhonov regularization method. As a result, the proposed algorithm can be viewed as a new effective implementation version of the classical Tikhonov regularization method. In addition, the convergence of the

algorithm needs no assumption when applied to the reformulated problem (5.2).

From our results, some interesting problems arise: What is the rate of convergence of Algorithm 3.1? Can certain modified versions of the algorithm be superlinearly (or quadratically) convergent in the neighborhood of the least 2-norm solution of a linear program? Can the least 2-norm solution of a linear program be solved in polynomial time? We believe that these problems are worth studying in the future.

Acknowledgments. We thank two anonymous referees for their helpful comments and suggestions. We also thank Professor Z. L. Wei for helping with the code.

REFERENCES

[1] M. J. Best, *Equivalence of some quadratic programming algorithms*, Math. Programming, 30 (1984), pp. 71–87.

[2] F. Facchinei, *Structural and stability properties of P_0 nonlinear complementarity problems*, Math. Oper. Res., 23 (1998), pp. 735–749.

[3] F. Facchinei and C. Kanzow, *Beyond monotonicity in regularization methods for nonlinear complementarity problems*, SIAM J. Control Optim., 37 (1999), pp. 1150–1161.

[4] G. Isac, *Tikhonov's regularization and the complementarity problem in Hilbert space*, J. Math. Anal. Appl., 174 (1991), pp. 53–66.

[5] C. Kanzow, H. Qi, and L. Qi, *On the Minimum Norm Solution of Linear Programming*, Technical report, Department of Mathematics, Center for Optimization and Approximation, Hamburg, Germany, 2000.

[6] K. C. Kiwiel, *Finding normal solutions in piecewise linear programming*, Appl. Math. Optim., 32 (1995), pp. 235–254.

[7] K. C. Kiwiel, *Iterative schemes for the least* 2*-norm solution of piecewise linear programs*, Linear Algebra Appl., 229 (1995), pp. 1–7.

[8] K. C. Kiwiel, *A dual method for certain positive semidefinite quadratic programming problems*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 175–186.

[9] M. Kojima, N. Megiddo, and T. Noma, *Homotopy continuation methods for nonlinear complementarity problems*, Math. Oper. Res., 16 (1991), pp. 754–774.

[10] Y. Y. Lin and J. S. Pang, *Iterative methods for large convex quadratic programs: A survey*, SIAM J. Control Optim., 25 (1987), pp. 383–411.

[11] S. Lucidi, *A new result in the theory and computation of the least-norm solution of a linear program*, J. Optim. Theory Appl., 55 (1987), pp. 103–117.

[12] S. Lucidi, *A finite algorithm for the least two-norm solution of a linear program*, Optimization, 18 (1987), pp. 809–823.

[13] O. L. Mangasarian, *Uniqueness of solution in linear programming*, Linear Algebra Appl., 25 (1979), pp. 151–162.

[14] O. L. Mangasarian, *Iterative solution of linear programs*, SIAM J. Numer. Anal., 18 (1981), pp. 606–614.

[15] O. L. Mangasarian, *Least-norm linear programming solution as an unconstrained minimization problem*, J. Math. Anal. Appl., 92 (1983), pp. 240–251.

[16] O. L. Mangasarian, *Normal solutions of linear programs*, Math. Programming Study, 22 (1984), pp. 206–216.

[17] O. L. Mangasarian, *Least norm solution of monotone linear complementarity problems*, in Functional Analysis, Optimization, and Mathematical Economics, L. J. Leifman, ed., Oxford University Press, New York, 1990, pp. 217–221.

[18] O. L. Mangasarian and R. De Leone, *Error bounds for strongly convex programs and (super)linearly convergent iterative schemes for the least* 2*-norm solution of linear programs*, Appl. Math. Optim., 17 (1988), pp. 1–14.

[19] O. L. Mangasarian and R. R. Meyer, *Nonlinear perturbation of linear programs*, SIAM J. Control Optim., 17 (1979), pp. 745–752.

[20] M. Padberg, *Linear Optimization and Extensions*, Springer-Verlag, Berlin, 1995.

[21] T. D. Parsons and A. W. Tucker, *Hybrid program: Linear and least-distance*, Math. Programming, 1 (1971), pp. 153–167.

[22] S. M. Robinson, *A characterization of stability in linear programming*, Oper. Res., 25 (1977), pp. 435–447.

[23] R. Saigal, *Linear Programming: A Modern Integrated Analysis*, Kluwer Academic Publishers, Norwell, MA, 1995.

[24] P. W. Smith and H. Wolkowicz, *A nonlinear equation for linear programming*, Math. Programming, 34 (1986), pp. 235–238.
[25] V. D. Skarin, *Methods for the correction of ill-posed problems of linear and convex programming by using a sequential programming approach*, in Parametric Optimization and Ill-Posed Problems in Mathematical Optimization, Seminarberichte 81, J. J. Eremin, ed., Humboldt University, Berlin, 1986, pp. 130–144.
[26] P. K. Subramanian, *A note on least two norm solutions of monotone complementarity problems*, Appl. Math. Lett., 1 (1988), pp. 395–397.
[27] R. Sznajder and M. S. Gowda, *On the limiting behavior of the trajectory of regularized solutions of P_0 complementarity problems*, in Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods, M. Fukushima and L. Qi, eds., Kluwer Academic Publishers, 1998, pp. 371–379.
[28] A. W. Tucker, *A least-distance approach to quadratic programming*, in Mathematics of the Decision Sciences, Part I, G. B. Dantzig and A. F. Veinott, Jr., eds., Lectures in Appl. Math. 11, AMS, Providence, RI, 1968, pp. 163–176.
[29] A. W. Tucker, *Least-distance programming*, in Proceedings of the Princeton Symposium on Mathematical Programming, H. W. Kuhn, ed., Princeton University Press, Princeton, NJ, 1971, pp. 583–588.
[30] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-posed Problems*, Halsted Press, Wiley, New York, 1977.
[31] H. P. Williams, *Model Building in Mathematical Programming*, John Wiley and Sons, New York, 1995.
[32] P. Wolfe, *Algorithm for a least-distance programming problem*, Math. Prog. Study, 1 (1974), pp. 190–205.
[33] P. Wolfe, *Finding the nearest point in a polytope*, Math. Programming, 11 (1976), pp. 128–149.
[34] Y. Ye, *Interior-Point Algorithms: Theory and Analysis*, John Wiley and Sons, Chichester, UK, 1997.
[35] Y. B. Zhao and D. Li, *On a new homotopy continuation trajectory for complementarity problems*, Math. Oper. Res., 26 (2001), pp. 119–146.
[36] Y.-B. Zhao and D. Li, *Existence and limiting behavior of a non-interior-point trajectory for nonlinear complementarity problems without strict feasibility condition*, SIAM J. Control Optim., 40 (2001), pp. 898–924.
[37] Y. B. Zhao and D. Li, *A New Path-Following Algorithm for Nonlinear P_* Complementarity Problems*, Technical report, Department of SEEM, Chinese University of Hong Kong, Hong Kong, 2000.

SIAM J. OPTIM.
Vol. 12, No. 4, pp. 913–927

ON THE SENSITIVITY ANALYSIS OF HOFFMAN CONSTANTS FOR SYSTEMS OF LINEAR INEQUALITIES*

D. AZÉ† AND J.-N. CORVELLEC‡

Abstract. Relying on a general variational method developed by the authors and Lucchetti [*Nonlinear Anal.*, to appear] (the origin of which goes back to Ioffe [*Trans. Amer. Math. Soc.,* 251 (1979), pp. 61–69]), we give a formula for the best Hoffman constant $\sigma = \inf_{x \notin P_{A,b}} \frac{\|(Ax-b)^+\|_\infty}{d(x,P_{A,b})}$, where $P_{A,b} = \{x : Ax \leq b\}$ is a nonempty polyhedron in $\mathbb{R}^n$. We also sharpen some results of Luo and Tseng [*SIAM J. Matrix Anal. Appl.,* 15 (1994), pp. 636–659] by characterizing the continuity set of some Hoffman constants and by pointing out their locally Lipschitzian character. We apply these results to the study of the behavior of the solution set of a linear program $\inf_{Ax \leq b} u^T x$ with respect to (A, b, u).

Key words. linear inequalities, Hoffman's bounds, polyhedral functions

AMS subject classifications. 15A39, 49K40, 90C05, 90C31, 65F35

PII. S1052623400375853

1. Introduction.

It is well known since Hoffman [14] that, assuming that the polyhedron $P_{A,b} = \{x \in \mathbb{R}^n : \ Ax \leq b\}$ is nonempty, there exists a positive constant K such that

$$K \sup_{1 \leq j \leq m} (a_j^T x - b_j)^+ \geq d(x, P_{A,b})$$

for every $x \in \mathbb{R}^n$. Recently, some estimates for this kind of constant were given in [4, 7, 11, 16, 18, 17], and for a more general case in [5]. When dealing with the question of the dependence of $P_{A,b}$ upon the data (A, b), the crucial point is to guarantee that the constant K remains bounded under perturbations of A and b, a central question in the paper of Luo and Tseng [20]; see also [19, 9].

In this paper, we consider this problem from a variational point of view, along the lines developed in [3]. This approach naturally leads us to use a constant σ which is just the inverse of K, that is,

$$\sup_{1 \leq j \leq m} (a_j^T x - b_j)^+ \geq \sigma d(x, P_{A,b})$$

for every $x \in \mathbb{R}^n$. In turns out that the variational nature of this constant allows for a new sensitivity analysis. Our method also permits us to give a formula for the optimal constant σ.

With these preliminaries in hand, it is then easy to study the behavior of $P_{A,b}$ with respect to the pair (A, b), and that of the solution set of the associated linear programming problems. Moreover, we provide new information on the classical stability result (see, e.g., [1, 22]) by giving conditions ensuring the lower semicontinuity

*Received by the editors July 24, 2000; accepted for publication (in revised form) June 22, 2001; published electronically March 13, 2002.
http://www.siam.org/journals/siopt/12-4/37585.html
†UMR CNRS MIP, Université Paul Sabatier, 118 Route de Narbonne, 31062 Toulouse cedex, France (aze@mip.ups-tlse.fr).
‡Laboratoire MANO, Université de Perpignan, 52 Avenue de Villeneuve, 66860 Perpignan cedex, France (corvellec@univ-perp.fr).

(in fact, the local Lipschitz character) of the solution set mapping with respect to the whole data set.

Finally, let us point out that the method developed here can be used to treat the case of polyhedra involving explicit equalities and to study the stability of semidefinite convex quadratic problems; see [12].

The paper is organized as follows. In section 2, we basically recall the general method introduced in [3], and we specialize it to the convex case. This leads to the computation of the best Hoffman constant for inequality systems, which is done in section 3. In section 4, we give a necessary and sufficient condition for the local Lipschitz property of a Hoffman constant with respect to the matrix A. The Lipschitz behavior of the solution set with respect to all data is then derived in section 5.

2. A general method. In this section, we let X be a metric space endowed with the metric d, and $f : X \to \mathbb{R} \cup \{+\infty\}$ be a lower semicontinuous function. We respectively denote by $B_r(x)$ and $\bar{B}_r(x)$ the open and closed ball of radius $r > 0$ centered at $x \in X$. If $x \in X$ and $Y \subset X$, we set

$$d(x, Y) := \inf\{d(x, y) : \ y \in Y\},$$

with the convention that $d(x, \emptyset) = +\infty$ (according to the general convention $\inf \emptyset = +\infty$). For $c \in \mathbb{R}$, we let

$$[f{\leq}c] := \{x \in X : \ f(x) \leq c\}, \quad [f{>}c] := \{x \in X : \ f(x) > c\},$$

respectively, denote the closed sublevel and open upper-level set of f at level c. We further denote by

$$\mathrm{dom} f := \{x \in X : \ f(x) < +\infty\}$$

the effective domain of f and say, as usual, that f is proper if $\mathrm{dom} f \neq \emptyset$. These notations will be used throughout the paper.

We first recall the notion of strong slope introduced by DeGiorgi, Marino, and Tosques [8].

DEFINITION 2.1. *The* (strong) slope *of f at $x \in \mathrm{dom} f$ is denoted and defined by*

$$|\nabla f|(x) := \begin{cases} \limsup\limits_{y \to x} \dfrac{f(x) - f(y)}{d(x, y)} & \text{if } x \text{ is not a local minimum of } f, \\ 0 & \text{if } x \text{ is a local minimum of } f. \end{cases}$$

If $x \notin \mathrm{dom} f$, we set $|\nabla f|(x) = +\infty$.

The following notion also plays a central role in what follows.

DEFINITION 2.2. *For $c \in \mathbb{R}$, we let $\sigma_c(f)$ denote the supremum of the σ's in $[0, +\infty[$ such that*

$$f(x) - c \geq \sigma d(x, [f{\leq}c]) \quad \text{for every } x \in [f{>}c], \tag{2.1}$$

with the convention that $\sigma_c(f) = 0$ if $[f{\leq}c] = \emptyset$. The extended real number $\sigma_c(f)$ is called the condition number of f at level c.

Introducing the notation $s^+ := \max\{s, 0\}$, $s \in \mathbb{R} \cup \{+\infty\}$, it is clear that (2.1) can equivalently be written in the form:

$$(f(x) - c)^+ \geq \sigma d(x, [f{\leq}c]) \quad \text{for every } x \in X.$$

Also, it is clear that

$$\sigma_c(f) = \inf_{f(x)>c} \frac{f(x)-c}{d(x,[f\leq c])} \tag{2.2}$$

(with the convention that the right-hand term is zero if $[f\leq c]=\emptyset$). Finally, observe that

$$\sigma_c(f) = +\infty \iff \mathrm{dom} f \subset [f\leq c]\,. \tag{2.3}$$

The following proposition, relying on Ekeland's variational principle [10], states in terms of the strong slope a basic fact we need. (See the beginning of [3, section 2] for more details.)

PROPOSITION 2.3. *Let (X,d) be a complete metric space, and $f: X \to \mathbb{R}\cup\{+\infty\}$ be a (proper) lower semicontinuous function. Let $\bar{x}\in X$, $\sigma>0$, and $r>0$ be such that*

$$f(\bar{x}) < \inf_{\bar{B}_r(\bar{x})} f + \sigma r\,.$$

Then there exists $x\in B_r(\bar{x})$ such that $|\nabla f|(x)<\sigma$.

We readily see from the proposition that for any $c\in\mathbb{R}$ we have that $[f>c]\cap \mathrm{dom}|\nabla f|$ is dense in $[f>c]\cap \mathrm{dom} f$, and that

$$\inf_{[f>c]} |\nabla f| > 0 \implies [f\leq c]\neq\emptyset\,.$$

The following result, which is our main tool in this paper, sharpens [3, Theorem 3.1, Remark 3.2] when dealing with the strong slope. For completeness, we give a proof below.

THEOREM 2.4. *Let (X,d) be a complete metric space, $f: X\to\mathbb{R}\cup\{+\infty\}$ be a proper lower semicontinuous function, and $a\in\mathbb{R}$. Then*

$$\inf_{[f>a]} |\nabla f| = \inf_{c\geq a} \sigma_c(f)\,.$$

Proof. We first show that $\sigma_c(f)\geq \inf_{[f>a]}|\nabla f|$ for any $c\geq a$. We may, of course, assume that $\inf_{[f>a]}|\nabla f|>0$, in which case $[f\leq a]\neq\emptyset$ (as observed above), and we consider a positive real number $\sigma\leq \inf_{[f>a]}|\nabla f|$. Assume that there are some $c\geq a$ and $\bar{x}\in[f>c]$ with

$$f(\bar{x})-c<\sigma d(\bar{x},[f\leq c])\,.$$

Set $r:=d(\bar{x},[f\leq c])>0$, $g:=(f-c)^+\geq 0$, i.e., $g(x):=\sup\{f(x)-c,0\}$, so that

$$g(\bar{x}) < \inf_{\bar{B}_r(\bar{x})} g + \sigma r\,.$$

According to Proposition 2.3, we find $x\in B_r(\bar{x})$ with $|\nabla g|(x)<\sigma$. By definition of r, we see that $f(x)>c$, so that $|\nabla f|(x)=|\nabla g|(x)<\sigma$, which is not true. Hence,

$$f(x)-c\geq \sigma d(x,[f\leq c]) \quad \text{for all } c\geq a \text{ and all } x\in[f>c]\,,$$

showing that $\sigma_c(f)\geq\sigma$, and the conclusion.

Conversely, we may assume that $\inf_{c\geq a}\sigma_c(f)>0$ and that $[f{>}a]\cap \mathrm{dom} f\neq\emptyset$, so let $0<\sigma<\sigma_c(f)$ for every $c\geq a$ (in particular, $[f{\leq}a]\neq\emptyset$), let $x\in[f{>}a]\cap\mathrm{dom} f$, and set $c_n:=f(x)-1/n$ for $n\in\mathbb{N}$ large enough so that $c_n\geq a$ and $\sigma>1/n$. For each $n\in\mathbb{N}$, let $x_n\in[f{\leq}c_n]$ such that

$$f(x)-c_n\geq(\sigma-1/n)d(x,x_n)\,.$$

Then we have

$$0<d(x,x_n)\leq\frac{f(x)-c_n}{\sigma-1/n}\;\to\;0\quad\text{as } n\to\infty\,,$$

$$\frac{f(x)-f(x_n)}{d(x,x_n)}\geq\frac{f(x)-c_n}{d(x,x_n)}\geq\sigma-1/n\;\to\;\sigma\quad\text{as } n\to\infty\,,$$

showing that $|\nabla f|(x)\geq\sigma$, and the conclusion follows. □

We now specialize the previous result to the case of convex functions defined on Banach spaces. Let X be a Banach space, endowed with a norm $\|\cdot\|$. We denote by X^* the topological dual of X, and by d_* the metric associated with the dual norm. Recall that if $f:X\to\mathbb{R}\cup\{+\infty\}$ is a convex lower semicontinuous function, the (Fenchel) subdifferential of f at $x\in\mathrm{dom} f$ is given by

$$\partial f(x)=\{x^*\in X^*:\;f(y)-f(x)\geq\langle x^*,y-x\rangle\}\,.$$

PROPOSITION 2.5. *Let X be a Banach space and $f:X\to\mathbb{R}\cup\{+\infty\}$ be a proper, convex, and lower semicontinuous function. Then*

(a) $|\nabla f|(x)=d_*(0,\partial f(x))$ *for every $x\in X$;*

(b) *for any $a\in\mathbb{R}$, it holds that*

$$\sigma_a(f)=\inf_{c\geq a}\sigma_c(f)\,.$$

Proof. Part (a) is a well-known fact; we sketch the proof for the reader's convenience. Let $x\in\mathrm{dom} f$. The inequality $|\nabla f|(x)\leq d_*(0,\partial f(x))$ follows from the definitions. Indeed, we may assume that $|\nabla f|(x)>0$. Letting $0<\sigma<|\nabla f|(x)$, we find $y\in X$, $y\neq x$, such that

$$f(x)-f(y)\geq\sigma\|x-y\|\,,$$

so that $\|\xi\|_*\geq\sigma$ whenever $\xi\in\partial f(x)$. To prove the reverse inequality, we may assume that $d_*(0,\partial f(x))>\sigma>0$, so that x is not a minimum point of f. Using a separation argument (see, e.g., [6, Lemma]), we find $y\in X$ such that

$$f(y)-f(x)<-\sigma\|x-y\|\,,$$

due to the convexity of f, which further implies that for small $t>0$

$$\frac{f(x+t(y-x))-f(x)}{t\|x-y\|}<-\sigma\,,$$

from which we get that $|\nabla f|(x)\geq\sigma$.

For part (b), we need to show that $\sigma_a(f)\leq\sigma_c(f)$ if $c>a$. We may assume (we are getting used to it now) that $\sigma_a(f)>\sigma>0$, so that $[f{\leq}a]\neq\emptyset$. Let $c>a$; if

$[f{>}c] \cap \mathrm{dom} f = \emptyset$, then $\sigma_c(f) = +\infty$. Otherwise, let $x \in [f{>}c] \cap \mathrm{dom} f$, and let $\varepsilon > 0$ and $y \in [f{\le}a]$ be such that

$$\|x - y\| \le (1+\varepsilon) d(x, [f{\le}a])\,,$$

so that

$$\frac{f(x)-a}{\|x-y\|} \ge \frac{\sigma d(x,[f{\le}a])}{\|x-y\|} \ge \frac{\sigma}{1+\varepsilon}\,.$$

As the convex function f is finite on $[x,y]$, it is continuous on $[x,y]$; thus we find a point z in the open segment $]x,y[$ such that

$$c = f(z) \le \frac{\|z-y\|}{\|x-y\|} f(x) + \frac{\|z-x\|}{\|x-y\|} a.$$

Hence

$$\begin{aligned} f(x) - c \;&\ge\; f(x)\left(1 - \frac{\|z-y\|}{\|x-y\|}\right) - \frac{\|z-x\|}{\|x-y\|} a \\ &=\; (f(x)-a)\frac{\|x-z\|}{\|x-y\|} \ge \frac{\sigma}{1+\varepsilon}\|x-z\| \ge \frac{\sigma}{1+\varepsilon} d(x,[f{\le}c])\,, \end{aligned}$$

showing that $\sigma_c(f) \ge \sigma/(1+\varepsilon)$, whence $\sigma_c(f) \ge \sigma$ from the arbitrariness of ε, and the conclusion follows. □

Combining Theorem 2.4 and Proposition 2.5 (recall (2.2)), we thus obtain the following theorem.

THEOREM 2.6. *Let X be a Banach space, $f : X \to \mathbb{R} \cup \{+\infty\}$ be a proper, convex, and lower semicontinuous function, and $a \in \mathbb{R}$. Then*

$$\inf_{x \in [f>a]} d_*(0, \partial f(x)) = \sigma_a(f) = \inf_{x \in [f>a]} \frac{f(x)-a}{d(x,[f{\le}a])}$$

(with the convention that the right-hand member is zero if $[f{\le}a] = \emptyset$).

Remark 2.1. (i) Theorem 2.6 is sharper than some results of a similar type in Auslender, Cominetti, and Crouzeix [2, section 6], where $X = \mathbb{R}^n$ and the additional assumption that $\inf_{\mathbb{R}^n} f < a$ is made.

(ii) Let $\mathbb{R}^n$ be endowed with the Euclidean norm, let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a proper, convex, lower semicontinuous function, and assume that $[f{\le}0] \ne \emptyset$. In this specific setting, Lewis and Pang showed in [17, Theorem 1] that for any $\gamma > 0$ it holds that $\gamma f(x)^+ \ge d(x,[f{\le}0])$ for all $x \in \mathbb{R}^n$ if and only if $f'(x;d) \ge \gamma^{-1}\|d\|$ for all $x \in f^{-1}(0)$ and all $d \in N_{[f\le 0]}(x)$, where $f'(x;d)$ denotes the directional derivative of f at x in the direction d, and $N_{[f\le 0]}(x)$ denotes the normal cone to $[f{\le}0]$ at x. Using our notation, this can be rephrased as $\sigma_0(f) = \kappa$ whenever

$$\kappa := \inf_{\{x \in f^{-1}(0),\, d \in N_{[f\le 0]}(x),\, \|d\|=1\}} f'(x;d) > 0\,.$$

Thus, letting

$$\sigma := \inf_{x \in [f>0]} d(0, \partial f(x))\,,$$

we must have $\kappa = \sigma$ in this situation, according to Theorem 2.6. It is indeed easy to see, from the definitions and convex analysis, that $\sigma \geq \kappa$. On the other hand, we remark that when dealing with finite systems of linear inequalities, it is straightforwardly seen that the condition "$\sigma > 0$" holds (see Lemma 3.2 below), while the condition "$\kappa > 0$" is not so easy to establish.

(iii) We put in Definition 2.2 that $\sigma_c(f) = 0$ whenever $[f{\leq}c] = \emptyset$. But it may well happen that $\sigma_c(f) = 0$ while $[f{\leq}c] \neq \emptyset$. For example, let X be a real Hilbert space endowed with an orthonormal basis $(e_k)_{k\geq 1}$, and let $f(x) := \frac{1}{2}\sum_{k=1}^{\infty} k^{-2}x_k^2$, where $x = \sum_{k=1}^{\infty} x_k e_k$, so that $\nabla f(x) = \sum_{k=1}^{\infty} k^{-2}x_k e_k$ (the gradient of f at x). Of course, $[f{\leq}0] \neq \emptyset$, while $f(ne_n) = \frac{1}{2}$ and $\nabla f(ne_n) = e_n/n \to 0$ as $n \to \infty$, showing that $\sigma_0(f) = 0$.

3. Sharp Hoffman estimate. In the remainder of the paper, we shall assume that $\mathbb{R}^n$ is endowed with a norm $\|\cdot\|$, the dual norm of which is denoted by $\|\cdot\|_*$, with associated metric d_*.

We denote by $\mathcal{M}_{m\times n}$ the set of $m \times n$ real matrices, and if $A \in \mathcal{M}_{m\times n}$, then we let $a_1^T, \ldots, a_m^T$ denote its rows, where $a_1, \ldots, a_m \in \mathbb{R}^n$. For (a nonempty) $J \subset [1, m] := \{1, \ldots, m\}$, we set $a_J := \{a_j : \ j \in J\}$, and we let $\mathrm{co}(a_J)$ and $\mathrm{pos}(a_J)$, respectively, denote the convex hull and the closed convex cone generated by the set a_J.

Given $b \in \mathbb{R}^m$, we denote by $P_{A,b}$ the polyhedron defined by

$$P_{A,b} := \{x \in \mathbb{R}^n : \ Ax \leq b\} = [f{\leq}0]\,,$$

where

$$f(x) := \sup_{1\leq j\leq m} (a_j^T x - b_j)\,. \tag{3.1}$$

For each $x \in \mathbb{R}^n$, we have (see, e.g., [23] or [13])

$$\partial f(x) = \mathrm{co}(a_{J_{A,b}(x)})\,,$$

where

$$J_{A,b}(x) := \{j \in [1, m] : \ a_j^T x - b_j = f(x)\}\,.$$

In this section, we give a new formula for the best Hoffman constant for systems of inequalities, and we discuss its relationship with the Hoffman constant given in [4, 11, 18]. We also give another formula for our constant that will be useful, in the next section, for the study of its dependence with respect to (A, b). We start with two preliminary lemmas.

LEMMA 3.1. *Let $a_1, \ldots, a_p \in \mathbb{R}^n$ be such that $0 \notin \mathrm{co}(a_{[1,p]})$. Then there exists $J \subset [1, p]$ such that $d_*(0, \mathrm{co}(a_J)) = d_*(0, \mathrm{co}(a_{[1,p]}))$ and the vectors $(a_j)_{j\in J}$ are linearly independent.*

Proof. Let $x \in \mathrm{co}(a_{[1,p]})$ with $\|x\|_* = d_*(0, \mathrm{co}(a_{[1,p]})) \neq 0$. According to Carathéodory's theorem (see, e.g., [24, Corollary 7.1j, p. 94]), and since x belongs to the boundary of $\mathrm{co}(a_{[1,p]})$, there exists $J \subset [1, p]$ such that $\mathrm{co}(a_J)$ is a simplex of dimension at most $n - 1$ containing x—thus $\|x\|_* = d_*(0, \mathrm{co}(a_J))$. Since $0 \notin \mathrm{co}(a_J)$, the vectors $(0, a_j)_{j\in J}$ also are affinely independent, so that the vectors $(a_j)_{j\in J}$ are linearly independent. ∎

In what follows, we shall naturally assume that the matrix A is not the zero matrix; we shall denote this assumption by $A \in \mathcal{M}^*_{m\times n}$. This is equivalent to $\sigma_0(f) < +\infty$ (recall (2.3)).

LEMMA 3.2. *Let $A \in \mathcal{M}^*_{m\times n}$ and $b \in \mathbb{R}^m$. Then $P_{A,b} \neq \emptyset$ if and only if*

$$\min_{f(x)>0} d_*(0, \partial f(x)) = \min_{x\notin P_{A,b}} d_*(0, \mathrm{co}(a_{J_{A,b}(x)})) > 0\,.$$

Proof. As $\partial f(x) = \mathrm{co}(a_{J_{A,b}(x)})$ for every $x \in \mathbb{R}^n$, it follows that

$$\sigma := \inf_{f(x)>0} d_*(0, \partial f(x)) = \min_{f(x)>0} d_*(0, \partial f(x))\,,$$

so that $\inf_{\mathbb{R}^n} f > 0$ if $\sigma = 0$, yielding $P_{A,b} = [f{\leq}0] = \emptyset$. Conversely, we already observed (see the previous section) that if $\sigma > 0$, then $[f{\leq}0] = P_{A,b} \neq \emptyset$. ∎

THEOREM 3.3. *Let $A \in \mathcal{M}^*_{m\times n}$ and $b \in \mathbb{R}^m$. Assume that the polyhedron $P_{A,b}$ is nonempty and set*

$$\sigma_{A,b} := \min\{d_*(0, \mathrm{co}(a_J)) :\ J \subset J_{A,b}(x),\ x \notin P_{A,b},\ (a_j)_{j\in J} \text{ linearly independent}\} > 0\,. \tag{3.2}$$

Then

$$\sigma_{A,b} = \inf_{x\notin P_{A,b}} \frac{f(x)}{d(x, P_{A,b})} = \inf_{x\notin P_{A,b}} \frac{\sup_{1\leq j\leq m}(a_j^T x - b_j)}{d(x, P_{A,b})}\,, \tag{3.3}$$

so that $\sigma_{A,b}$ is the greatest positive constant τ such that

$$\sup_{1\leq j\leq m} (a_j^T x - b_j)^+ \geq \tau d(x, P_{A,b}) \qquad \text{for all } x \in \mathbb{R}^n\,.$$

Proof. According to Lemma 3.2 and Lemma 3.1, we have that

$$\sigma_{A,b} = \min_{x\notin P_{A,b}} d_*(0, \mathrm{co}(a_{J_{A,b}(x)})) = \min_{x\notin P_{A,b}} d_*(0, \partial f(x)) = \min_{f(x)>0} d_*(0, \partial f(x)) > 0\,, \tag{3.4}$$

and the conclusions are given by Theorem 2.6, applied to f with $a := 0$. ∎

We now establish another useful lemma.

LEMMA 3.4. *Let $c \geq 0$ and $x \in [f{>}c]$. Then there exist $y \in \mathbb{R}^n$, $J \subset J_{A,b}(y)$, and $\zeta \in \mathrm{co}\,(a_J)$ such that $f(y) = c$, the vectors $(a_j)_{j\in J}$ are linearly independent, and*

$$d_*(0, \partial f(x)) \geq \|\zeta\|_*\,.$$

Proof. Let $\xi \in \partial f(x)$ be such that $d_*(0, \partial f(x)) = \|\xi\|_*$, and let y be a projection of x on $[f{\leq}c]$. Of course, $f(y) = c$ and $x \neq y$. We can find $\hat\zeta \in N_{[f\leq c]}(y) = \mathrm{pos}\,(a_{J_{A,b}(y)})$ such that $\|\hat\zeta\|_* = 1$ and $\hat\zeta^T(x-y) = \|x-y\|$ (where $N_{[f\leq c]}(y)$ denotes the normal cone to $[f{\leq}c]$ at the point y). From Carathéodory's theorem (see, e.g., [24, Corollary 7.1i, p. 94]), there exists $J \subset J_{A,b}(y)$ such that the vectors $(a_j)_{j\in J}$ are linearly independent and $\hat\zeta \in \mathrm{pos}\,(a_J)$. It follows that there exists $\lambda > 0$ such that $\zeta := \lambda\hat\zeta \in \mathrm{co}\,(a_J) \subset \partial f(y)$ satisfies $\zeta^T(x-y) = \|\zeta\|_*\|x-y\|$, and thus

$$\|\zeta\|_*\|x-y\| \leq f(x) - f(y) \leq \xi^T(x-y) \leq \|\xi\|_*\|x-y\|\,,$$

yielding $\|\xi\|_* \geq \|\zeta\|_*$, from which our assertion follows. ∎

PROPOSITION 3.5. *Let $A \in \mathcal{M}^*_{m\times n}$ and $b \in \mathbb{R}^m$ such that $P_{A,b} \neq \emptyset$. Then, for every $\varepsilon > 0$ we have*

$$\sigma_{A,b} = \min_{\varepsilon\geq f(x)>0} d_*(0, \mathrm{co}\,(a_{J_{A,b}(x)}))\,.$$

Proof. According to Lemma 3.4, applied with $c := \varepsilon > 0$, it holds that

$$\inf_{f(x)>\varepsilon} d_*(0, \operatorname{co}(a_{J_{A,b}(x)})) \geq \inf_{\varepsilon \geq f(x)>0} d_*(0, \operatorname{co}(a_{J_{A,b}(x)}))\,,$$

and the assertion follows from (3.4). □

Bergthaller and Singer proved in [4] that the constant

$$C_{bs} := \max_{\{J \subset J_{A,b}(x):\ x \in P_{A,b},\, (a_j)_{j\in J} \text{ lin. ind.}\}} \ \max_{\{w \in \mathbb{R}^J_+:\ \| \sum_{j\in J} w_j a_j\|_* = 1\}} \sum_{j\in J} w_j$$

satisfies

$$d(x, P_{A,b}) \leq C_{bs} f(x)^+ \qquad \text{for all } x \in \mathbb{R}^n.$$

(We assume that $P_{A,b} \neq \emptyset$.) It was pointed out in [20] that the computation of this type of Hoffman constant is quite involved and that it is not suitable for the sensitivity analysis under perturbations of the matrix A. Consider the constant

$$\sigma_{bs} := \min\{d_*(0, \operatorname{co}(a_J)) :\ J \subset J_{A,b}(x),\ x \in P_{A,b},\ (a_j)_{j\in J} \text{ lin. ind.}\}$$

(to be compared with (3.2)), which has a clear geometric meaning. It turns out that $\sigma_{bs} = C_{bs}^{-1}$, the inverse of C_{bs}. If $x \in P_{A,b}$ and $J \subset J_{A,b}(x)$ is such that the vectors in a_J are linearly independent, let $w \in \mathbb{R}^J_+$ with $\| \sum_{j\in J} w_j a_j\|_* = 1$. Setting $u_j := (\sum_{i\in J} w_i)^{-1} w_j \in \mathbb{R}_+$, we have

$$\sum_{j\in J} u_j = 1 \qquad \text{and} \qquad \left\| \sum_{j\in J} u_j a_j \right\|_* = \frac{1}{\sum_{j\in J} w_j}\,,$$

showing that $\sigma_{bs} \leq C_{bs}^{-1}$. Conversely, let $u \in \mathbb{R}^J_+$ with $\sum_{j\in J} u_j = 1$ and let $z := \sum_{j\in J} u_j a_j$. Setting $w_j := \|z\|_*^{-1} u_j$, we have

$$\left\| \sum_{j\in J} w_j a_j \right\|_* = 1 \qquad \text{and} \qquad \sum_{j\in J} w_j = \frac{1}{\|z\|_*}\,,$$

showing that $C_{bs} \geq \sigma_{bs}^{-1}$.

It follows from these considerations and from Theorem 3.3 that $\sigma_{bs} \leq \sigma_{A,b}$. However, this inequality can be seen directly. Indeed, given $\bar{x}$ with $f(\bar{x}) > 0$ such that $d_*(0, \partial f(\bar{x})) = \min_{f(x)>0} d_*(0, \partial f(x))$, Lemma 3.4 tells us that there exists $y \in [f \leq 0] = P_{A,b}$, $J \subset J_{A,b}(y)$ such that the vectors $(a_j)_{j\in J}$ are linearly independent, and $\zeta \in \operatorname{co}(a_J)$ such that

$$\sigma_{A,b} = d_*(0, \partial f(\bar{x})) \geq \|\zeta\|_* \geq \sigma_{bs}\,.$$

The following example shows that it may occur that $\sigma_{bs} < \sigma_{A,b}$.

Example 3.1. Let $n = 2$, $m = 3$, $a_1 = \binom{1}{0}$, $a_2 = \binom{0}{1}$, $a_3 = \binom{\delta}{\delta}$, with $0 < \delta < 1/2$, and let $b = 0 \in \mathbb{R}^3$. The admissible J's for the constant $\sigma_{A,b}$ are $\{1\}$, $\{2\}$, and $\{1, 2\}$, yielding $\sigma_{A,0} = \frac{\sqrt{2}}{2}$, which is sharp. On the other hand, the admissible J's for the constant σ_{bs} are $\{1\}$, $\{2\}$, $\{3\}$, $\{1, 2\}$, $\{1, 3\}$, and $\{3, 2\}$, leading to $\sigma_{bs} = \delta\sqrt{2}$, which

is not optimal. It can be observed from this example that the best Hoffman constant depends on the right member b. Indeed, letting

$$b = \begin{pmatrix} 0 \\ 0 \\ -\delta \end{pmatrix},$$

the admissible J's are $\{1\}$, $\{2\}$, $\{3\}$, $\{1,3\}$, and $\{2,3\}$, so that $\sigma_{A,b} = \delta\sqrt{2}$ in that case.

4. Regularity of two Hoffman constants. In [20], Luo and Tseng gave a necessary and sufficient condition on the matrix A ensuring that a Hoffman constant is bounded away from zero near A. In this section we go one step further, by characterizing the continuity of some Hoffman constants at A, and by showing that these Hoffman constants have a Lipschitzian behavior near A whenever they are continuous at A. The proof is very simple and geometric.

We consider the set $\mathcal{M}_{m\times n}$ of $m \times n$ real matrices A as endowed with the norm

$$\|A\| := \max_{1\le j\le m} \|a_j\|_*,$$

where, as before, $a_1^T, \dots, a_m^T$ are the rows of A. For (a nonempty) $J \subset [1,m]$, we denote by A_J the matrix with rows $(a_j^T)_{j\in J}$, and by $\mathrm{rank}(a_J)$ the rank of a_J, which is the column rank of the matrix A_J. Also, we respectively write $\mathrm{bdry}(\mathrm{co}(a_J))$ and $\mathrm{int}(\mathrm{co}(a_J))$ to denote the boundary and interior of $\mathrm{co}(a_J)$.

The first Hoffman constant we shall deal with in this section assigns to the matrix $A \in \mathcal{M}^*_{m\times n}$ the positive real number

$$\sigma(A) := \min_{\{J\subset[1,m]:\ 0\notin \mathrm{co}(a_J)\}} d_*(0, \mathrm{co}(a_J))\,. \tag{4.1}$$

If $b \in \mathbb{R}^m$ is such that $P_{A,b} \neq \emptyset$, we clearly have $\sigma(A) \le \sigma_{A,b}$ (the sharp Hoffman constant defined in (3.2); recall (3.4)), so that

$$\sup_{1\le j\le m} (a_j^T x - b_j)^+ \ge \sigma(A) d(x, P_{A,b}) \qquad \text{for all } x \in \mathbb{R}^n\,,$$

according to Theorem 3.3.

THEOREM 4.1. *Let $A \in \mathcal{M}^*_{m\times n}$ and assume that*

$$0 \notin \mathrm{bdry}(\mathrm{co}(a_J)) \qquad \textit{for all } J \subset [1,m]\,. \tag{4.2}$$

Then the function $\sigma(\cdot)$ defined in (4.1) *is Lipschitz continuous near A.*

Conversely, if $0 \in \mathrm{bdry}(\mathrm{co}(a_J))$ for some $J \subset [1,m]$, then for every $b \in \mathbb{R}^m$ such that $P_{A,b} \neq \emptyset$, for every $x \in P_{A,b}$, and for every $\varepsilon > 0$ there exist $A^\varepsilon \in \mathcal{M}_{m\times n}$ and $b^\varepsilon \in \mathbb{R}^m$ such that

$$P_{A^\varepsilon,b^\varepsilon} \neq \emptyset\,, \quad \lim_{\varepsilon\to 0}(A^\varepsilon, b^\varepsilon) = (A, \hat b)\,, \quad \textit{and} \quad \lim_{\varepsilon\to 0} \sigma_{A^\varepsilon,b^\varepsilon} = 0\,,$$

where $\hat b_j = a_j^T x$ if $j \in J$, and $\hat b_j = b_j$ if $j \notin J$.

Proof. For $\tilde A \in \mathcal{M}_{m\times n}$, let

$$\mathcal{J}(\tilde A) := \{J \subset [1,m] : 0 \notin \mathrm{co}(\tilde a_J)\}\,, \quad \mathcal{J}_0(\tilde A) := \{J \subset [1,m] : 0 \in \mathrm{co}(\tilde a_J)\}\,.$$

If $J \in \mathcal{J}_0(A)$, (4.2) tells us that $0 \in \mathrm{int}(\mathrm{co}(a_J))$. Thus, there exists a neighborhood $\mathcal{N}$ of A such that $\mathcal{J} := \mathcal{J}(A) = \mathcal{J}(\tilde{A})$ and $\mathcal{J}_0(A) = \mathcal{J}_0(\tilde{A})$ whenever $\tilde{A} \in \mathcal{N}$. Since for any A^1, $A^2 \in \mathcal{N}$ and any $J \in \mathcal{J}$, we have (with obvious notations and through a straightforward computation)

$$d_*(0, \mathrm{co}(a^1_J)) \leq d_*(0, \mathrm{co}(a^2_J)) + \|A^1 - A^2\|\,,$$

we conclude that $\sigma(A^1) \leq \sigma(A^2) + \|A^1 - A^2\|$, and thus $\sigma(\cdot)$ is Lipschitz of rank 1 near A.

Conversely, assume that $0 \in \mathrm{bdry}(\mathrm{co}(a_J))$ for some $J \subset [1, m]$, so that there exists $y \in \mathbb{R}^n$ such that $y \notin \mathrm{pos}(a_J)$. We then find $z \in \mathbb{R}^n$ such that $-y^T z > 0$ and $a_j^T z \geq 0$ for $j \in J$. Let $\tilde{b} \in \mathbb{R}^m$ be such that $Az \leq \tilde{b}$, $\tilde{b}_j \geq 0$ for all j's, and $\tilde{b}_j = a_j^T z$ for $j \in J$. Given $\varepsilon > 0$, let A^ε be the $m \times n$ matrix with rows $(a_j^\varepsilon)^T$ defined by

$$a_j^\varepsilon := a_j - \varepsilon y \quad \text{for } j \in J\,, \quad \text{and} \quad a_j^\varepsilon := a_j \quad \text{for } j \notin J\,.$$

Clearly, $A^\varepsilon \to A$ as $\varepsilon \to 0$. Now, let $b \in \mathbb{R}^m$ such that $P_{A,b} \neq \emptyset$, and let $x \in P_{A,b}$. For $\varepsilon > 0$, set

$$b_j^\varepsilon := (a_j^\varepsilon)^T x + \varepsilon \tilde{b}_j \quad \text{for } j \in J\,, \quad \text{and} \quad b_j^\varepsilon := b_j + \varepsilon \tilde{b}_j \quad \text{for } j \notin J\,.$$

Then $b^\varepsilon \to \hat{b}$ as $\varepsilon \to 0$, and it is straightforward to verify that $x \in P_{A^\varepsilon, b^\varepsilon}$. Moreover, we have

$$(a_j^\varepsilon)^T (x + \varepsilon z) - b_j^\varepsilon = -\varepsilon^2 y^T z > 0 \quad \text{for } j \in J\,,$$

while

$$(a_j^\varepsilon)^T (x + \varepsilon z) - b_j^\varepsilon = a_j^T x + \varepsilon a_j^T z - b_j^\varepsilon \leq 0 \quad \text{for } j \notin J\,,$$

so that $x + \varepsilon z \notin P_{A^\varepsilon, b^\varepsilon}$ and $J_{A^\varepsilon, b^\varepsilon}(x + \varepsilon z) = J$. Keeping in mind that $0 \in \mathrm{co}\,(a_J)$, we conclude (recall (3.4)) that $\sigma_{A^\varepsilon, b^\varepsilon} \leq d_*(0, \mathrm{co}(a_J^\varepsilon)) \leq \varepsilon \|y\|_* \to 0$ as $\varepsilon \to 0$. □

Remark 4.1. Of course, assumption (4.2) can be written as

$$\text{for all } J \subset [1, m]\,, \text{ either } 0 \notin \mathrm{co}(a_J) \text{ or } 0 \in \mathrm{int}(\mathrm{co}(a_J))\,, \tag{4.3}$$

which is equivalent to

$$0 \notin \mathrm{co}(a_J) \quad \text{for all } J \subset [1, m] \text{ such that } \quad \mathrm{rank}(a_J) < n\,. \tag{4.4}$$

Indeed, it is clear that (4.3) implies (4.4); conversely, assume that (4.4) holds, and let $0 \in \mathrm{co}(a_J)$ for some $J \subset [1, m]$ with $\mathrm{rank}(a_J) = n$. We can assume that $\sum_{j \in J} t_j a_j = 0$ with $t_j > 0$, $j \in J$, and $\sum_{j \in J} t_j = 1$. As $\mathrm{vect}\,(a_J) = \mathbb{R}^n$, it follows that $\mathrm{pos}\,(a_J) = \mathbb{R}^n$, thus $0 \in \mathrm{int}(\mathrm{co}(a_J))$, so that (4.3) holds true. Now, as $0 \notin \mathrm{co}(a_J)$ if and only if there exists $x \in \mathbb{R}^n$ with $A_J x < 0$, we observe that (4.2) is equivalent to assumption (a) of [20, Theorem 2.2]:

$$\text{for all } J \subset [1, m]\,, \text{ either } A_J x < 0 \text{ is solvable or } A_J \text{ has full column rank}\,. \tag{4.5}$$

Theorem 4.1 contains the quoted result of Luo and Tseng, with a simpler proof, and an additional conclusion on the Lipschitz behavior of the Hoffman constant. We also observe that Luo and Tseng mention in their paper that condition (4.5) is difficult to verify in practice. On the contrary, our geometric assumption (4.2) has a

clear geometric interpretation and seems easier to check, at least for rather "small" matrices.

Remark 4.2. It is worth noticing that $0 \in \operatorname{int}(\operatorname{co}(a_J))$ if and only if $\operatorname{pos}(a_J) = \mathbb{R}^n$, which in turn is equivalent to the boundedness of every (nonempty) polyhedron P_{A_J,b_J}.

It is also possible to get a result of *local* type, as in [20, Theorem 2.4], involving a Hoffman constant $\tau(\cdot)$ defined in a neighborhood of A (and a different assumption than (4.2)), as we now show.

It is established in the first step of the proof of [20, Theorem 2.4] that, given $P_{A,b} \neq \emptyset$, there exists a neighborhood $\mathcal{N}$ of (A,b) such that for every $(\hat{A},\hat{b}) \in \mathcal{N}$ and every $J \subset [1,m]$ such that $P_{\hat{A},\hat{b}} \cap \hat{A}_J^{-1}(\hat{b}_J) \neq \emptyset$, we have $J \in \mathcal{I}_{A,b}$, where

$$\hat{A}_J^{-1}(\hat{b}_J) := \{x \in \mathbb{R}^n : \ \hat{a}_j^T x = \hat{b}_j \text{ for } j \in J\}$$

and

$$\mathcal{I}_{A,b} := \{J \subset [1,m] : \ P_{A,b} \cap A_J^{-1}(b_J) \neq \emptyset\} \cup \{J \subset [1,m] : \ P_{A,0} \cap A_J^{-1}(0) \neq \{0\}\}. \tag{4.6}$$

Choose $\mathcal{N}$ open and set, for $\hat{A} \in \mathcal{M}^*_{m\times n}$,

$$\tau(\hat{A}) := \min_{\{J \in \mathcal{I}_{A,b}:\ 0 \notin \operatorname{co} \hat{a}_J\}} d_*(0, \operatorname{co}(\hat{a}_J))\,. \tag{4.7}$$

Given $(\hat{A},\hat{b}) \in \mathcal{N}$ such that $P_{\hat{A},\hat{b}} \neq \emptyset$, we can find $\varepsilon > 0$ such that $\{\hat{A}\} \times \bar{B}_\varepsilon(\hat{b}) \subset \mathcal{N}$, so that $J_{\hat{A},\hat{b}}(x) \in \mathcal{I}_{A,b}$ for any x such that $0 < \hat{f}(x) \leq \varepsilon$ (where $\hat{f}(x) := \sup_{1\leq j\leq m}(\hat{a}_j^T x - \hat{b}_j)$). Since

$$\sigma_{\hat{A},\hat{b}} = \min_{\varepsilon \geq \hat{f}(x) > 0} d_* \left(0, \operatorname{co}\, \left(a_{J_{\hat{A},\hat{b}}(x)}\right)\right)\,,$$

according to Proposition 3.5, it follows that $\tau(\hat{A})$ is a Hoffman constant for all $(\hat{A},\hat{b}) \in \hat{\mathcal{N}}$ such that $P_{\hat{A},\hat{b}} \neq \emptyset$.

THEOREM 4.2. *Let $(A,b) \in \mathcal{M}^*_{m\times n} \times \mathbb{R}^m$ with $P_{A,b} \neq \emptyset$ be such that*

$$0 \notin \operatorname{bdry}(\operatorname{co}(a_J)) \qquad \text{for all } J \in \mathcal{I}_{A,b}\,, \tag{4.8}$$

where $\mathcal{I}_{A,b}$ is defined in (4.6). *Then the function $\tau(\cdot)$ defined in* (4.7) *is Lipschitz continuous near A.*

Conversely, if $0 \in \operatorname{bdry}(\operatorname{co}(a_J))$ for some $J \in \mathcal{I}_{A,b}$, then for any $\varepsilon > 0$ there exist $A^\varepsilon \in \mathcal{M}_{m\times n}$ and $b^\varepsilon \in \mathbb{R}^m$ such that

$$P_{A^\varepsilon,b^\varepsilon} \neq \emptyset\,, \quad \lim_{\varepsilon\to 0}(A^\varepsilon,b^\varepsilon) = (A,b)\,, \quad \text{and} \quad \lim_{\varepsilon\to 0} \sigma_{A^\varepsilon,b^\varepsilon} = 0\,.$$

Proof. The first part of the proof goes exactly as that of the first part of Theorem 4.1. Conversely, assume that $0 \in \operatorname{bdry}(\operatorname{co}(a_J))$ for some $J \in \mathcal{I}_{A,b}$. If $P_{A,b} \cap A_J^{-1}(b_J) \neq \emptyset$, then the conclusion follows from the converse part of Theorem 4.1 by taking $x \in P_{A,b} \cap A_J^{-1}(b_J)$, so that $\hat{b} = b$. Assuming that $P_{A,0} \cap A_J^{-1}(0) \neq \{0\}$, we find $y, z \in \mathbb{R}^n$ such that $Az \leq 0$, $A_J z = 0$, and $y^T z > 0$. Let $x \in P_{A,b}$ and $t > 0$. Setting $r_j := b_j - a_j^T x + t$ for $j \in J$, let A^ε be the $m \times n$ matrix with rows $(a_j^\varepsilon)^T$ defined by

$$a_j^\varepsilon := a_j + \varepsilon r_j y \ \text{ for } j \in J, \quad \text{and} \quad a_j^\varepsilon := a_j \ \text{ for } j \notin J\,.$$

Given $\lambda \in \mathbb{R}$, we have for $j \in J$

$$(a_j^\varepsilon)^T(x+\lambda z) = a_j^T x + \varepsilon r_j y^T x + \lambda \varepsilon r_j y^T z = b_j + t - r_j(1 - \varepsilon y^T x - \lambda \varepsilon y^T z)\,,$$

while for $j \notin J$

$$(a_j^\varepsilon)^T(x+\lambda z) = a_j^T x + \lambda a_j^T z \leq b_j + \lambda a_j^T z\,.$$

In particular, we have

$$(a_j^\varepsilon)^T x = a_j^T x + \varepsilon r_j y^T x \leq b_j + \varepsilon \|r\|_\infty |y^T x| =: b_j^\varepsilon \quad \text{for } j \in J$$

and

$$(a_j^\varepsilon)^T x \leq b_j =: b_j^\varepsilon \quad \text{for } j \notin J\,,$$

so that $b^\varepsilon \to b$ as $\varepsilon \to 0$, and $x \in P_{A^\varepsilon, b^\varepsilon}$. On the other hand, choosing $\lambda = \frac{1-\varepsilon y^T x}{\varepsilon y^T z}$ and ε small enough in order that $\lambda > 0$ and $\varepsilon \|r\|_\infty |y^T x| < t$, we get

$$(a_j^\varepsilon)^T(x+\lambda z) - b_j^\varepsilon = t - \varepsilon \|r\|_\infty |y^T x| > 0 \ \text{ for } j \in J$$

and

$$(a_j^\varepsilon)^T(x+\lambda z) - b_j^\varepsilon = \lambda a_j^T z \leq 0 \ \text{ for } j \in J\,,$$

showing that $x + \lambda z \notin P_{A^\varepsilon, b^\varepsilon}$ and $J_{A^\varepsilon, b^\varepsilon}(x+\lambda z) = J$. Relying on the fact that $0 \in \mathrm{co}(a_J)$, we conclude (recall (3.4)) that $\sigma_{A^\varepsilon, b^\varepsilon} \leq d_*(0, \mathrm{co}(a_J^\varepsilon)) \leq \varepsilon \|r\|_\infty \|y\|_* \to 0$ as $\varepsilon \to 0$. □

Remark 4.3. Theorem 4.2 extends [20, Theorem 2.4] since, arguing in a similar way as in Remark 4.1, one can see that condition (4.8) is equivalent to condition (a) in the mentioned result. Our proof of the necessary part is similar to, but still shorter and simpler than, that in [20, Theorem 2.4].

5. Lower stability in linear programming. Let us fix some notations. If Y and Z are subsets of $\mathbb{R}^n$, we let

$$e_{\mathcal{H}}(Y,Z) := \sup_{y \in Y} d(y,Z)$$

denote the Hausdorff excess of Y with respect to Z, with the convention that

$$e_{\mathcal{H}}(\emptyset, Z) = 0,$$

and

$$d_{\mathcal{H}}(Y,Z) := \max\{e_{\mathcal{H}}(Y,Z), e_{\mathcal{H}}(Z,Y)\}$$

denote the Hausdorff distance between Y and Z. All other notations used in this section have already been introduced.

We consider the linear programming problem

$$\min_{x \in P_{A,b}} u^T x\,,$$

where $u \in \mathbb{R}^n$, and its dual problem

$$\min_{y \geq 0\,,\ A^T y = -u} y^T b\,.$$

We set

$$\mu_{A,b,u} := \inf_{x \in P_{A,b}} u^T x\,,$$

and we denote by $S_{A,b,u}$ (resp., $S^*_{A,b,u}$) the solution set of the primal (resp., dual) problem. It is known (see [1, 22]) that a necessary and sufficient condition ensuring that $S_{\hat{A},\hat{b},\hat{u}} \neq \emptyset$ and $S^*_{\hat{A},\hat{b},\hat{u}} \neq \emptyset$ for all $(\hat{A},\hat{b},\hat{u})$ near (A,b,u) is

$$0 \in \operatorname{int}\left(\operatorname{co}\left(a_{[1,m]} \cup \{u\}\right)\right) \tag{5.1}$$

and

$$\text{there exists } x \in \mathbb{R}^n \text{ such that } Ax < b\,. \tag{5.2}$$

Condition (5.1) is equivalent to

$$\operatorname{rank}(A) = n\,, \quad \text{and} \quad [A^T y = u\,,\ y < 0] \text{ is solvable}\,.$$

Moreover, if these two conditions are in force, we have that $S_{A,b,u} \cup S^*_{A,b,u}$ is bounded, that

$$\lim_{(\hat{A},\hat{b},\hat{u}) \to (A,b,u)} \mu_{\hat{A},\hat{b},\hat{u}} = \mu_{A,b,u}\,,$$

and that the multifunctions $(\hat{A},\hat{b},\hat{u}) \mapsto S_{\hat{A},\hat{b},\hat{u}}$ and $(\hat{A},\hat{b},\hat{u}) \mapsto S^*_{\hat{A},\hat{b},\hat{u}}$ are *Hausdorff upper semicontinuous at* (A,b,u); that is,

$$\lim_{(\hat{A},\hat{b},\hat{u}) \to (A,b,u)} e_{\mathcal{H}}(S_{\hat{A},\hat{b},\hat{u}}, S_{A,b,u}) = 0 \quad \text{and} \quad \lim_{(\hat{A},\hat{b},\hat{u}) \to (A,b,u)} e_{\mathcal{H}}(S^*_{\hat{A},\hat{b},\hat{u}}, S^*_{A,b,u}) = 0\,,$$

(see, e.g., [22, Theorem 1]), and, in particular, $S_{\hat{A},\hat{b},\hat{u}}$ is uniformly bounded for $(\hat{A},\hat{b},\hat{u})$ close to (A,b,u).

It is of interest to know conditions ensuring that these multifunctions are also *lower semicontinuous at* (A,b,u); that is, ensuring that every solution of the unperturbed problem can be approached by a solution of the perturbed problem as $(\hat{A},\hat{b},\hat{u})$ goes to (A,b,u). This is what we do in the following result.

THEOREM 5.1. *Assume that conditions* (5.1) *and* (5.2) *hold and that*

$$0 \notin \operatorname{bdry}(\operatorname{co}(a_J \cup \{\epsilon u\})) \quad \textit{for all } J \subset [1,m] \textit{ and for } \epsilon = 0, 1\,. \tag{5.3}$$

Then there exist constants α, $\beta > 0$ *and neighborhoods* $\mathcal{N}$ *of* A, $\mathcal{B}$ *of* b, *and* $\mathcal{U}$ *of* u *such that for all* $A^i \in \mathcal{N}$, $b^i \in \mathcal{B}$, *and* $u^i \in \mathcal{U}$, $i = 1, 2$, *we have*

$$|\mu_{A^1,b^1,u^1} - \mu_{A^2,b^2,u^2}| \leq \alpha(\|A^2 - A^1\| + \|u^2 - u^1\|_* + \|b^2 - b^1\|_\infty) \tag{5.4}$$

and

$$d_{\mathcal{H}}(S_{A^1,b^1,u^1}, S_{A^2,b^2,u^2}) \leq \beta(\|A^2 - A^1\| + \|u^2 - u^1\|_* + \|b^2 - b^1\|_\infty)\,. \tag{5.5}$$

Proof. Let $r > 0$ and $\mathcal{V}$ be a neighborhood of (A,b,u) such that $\emptyset \neq S_{\hat{A},\hat{b},\hat{u}} \subset \bar{B}_r(0)$ for all $(\hat{A},\hat{b},\hat{u}) \in \mathcal{V}$. Using condition (5.3) with $\epsilon = 0$, and applying Theorem 4.1, we can assume that for some constant $\gamma > 0$ we have $\sigma(\hat{A}) \geq \gamma$ whenever $(\hat{A},\hat{b},\hat{u}) \in \mathcal{V}$.

Let (A^1,b^1,u^1), $(A^2,b^2,u^2)\in\mathcal{V}$ and $x^1\in S_{A^1,b^1,u^1}$. Then let $x^2\in P_{A^2,b^2}$ be such that

$$\gamma\|x^1-x^2\|=\gamma d(x^1,P_{A^2,b^2})\le\sup_{1\le j\le m}((a_j^2)^Tx^1-b_j^2)^+\le r\|A^1-A^2\|+\|b^1-b^2\|_\infty\,.$$

Since

$$\mu_{A^2,b^2,u^2}\le(u^2)^Tx^2=\mu_{A^1,b^1,u^1}+(u^2-u^1)^Tx^1+(u^2)^T(x^2-x^1)\,,$$

we derive

$$\mu_{A^2,b^2,u^2}-\mu_{A^1,b^1,u^1}\le r\|u^2-u^1\|_*+\gamma^{-1}\|u^2\|_*(r\|A^1-A^2\|+\|b^1-b^2\|_\infty)\,,$$

yielding (5.4) after interchanging (A^2,b^2,u^2) with (A^1,b^1,u^1).

Let us now set, for $(\hat A,\hat b,\hat u)\in\mathcal{V}$,

$$\tilde A:=\begin{pmatrix}\hat A\\ \hat u^T\end{pmatrix}\qquad\text{and}\qquad\tilde b:=\begin{pmatrix}\hat b\\ \mu_{\hat A,\hat b,\hat u}\end{pmatrix},$$

so that $S_{\hat A,\hat b,\hat u}=P_{\tilde A,\tilde b}\subset\bar B_r(0)$. Using condition (5.3) with $\epsilon\in\{0,1\}$ and applying Theorem 4.1 again, we find a neighborhood $\tilde{\mathcal{N}}$ of $(A,u^T)^T$ and a constant $\delta>0$ such that $\sigma(\tilde A)\ge\delta$ for all $\tilde A\in\tilde{\mathcal{N}}$. We may choose $\tilde{\mathcal{N}}$ and a neighborhood $\tilde{\mathcal{B}}$ of b in such a way that $\tilde{\mathcal{N}}\times\tilde{\mathcal{B}}\subset\mathcal{V}$. Let $(A^1,u^1,b^1),(A^2,u^2,b^2)\in\tilde{\mathcal{N}}\times\tilde{\mathcal{B}}$, and $x^1\in S_{A^1,b^1,u^1}$. We have

$$\begin{aligned}\delta\,d(x^1,S_{A^2,b^2,u^2})&\le\sup_{1\le j\le m}((a_j^2)^Tx^1-b_j^2)^+ +((u^2)^Tx^1-\mu_{A^2,b^2,u^2})^+\\&\le r\|A^2-A^1\|+\|b^2-b^1\|_\infty+r\|u^2-u^1\|_*+|\mu_{A^1,b^1,u^1}-\mu_{A^2,b^2,u^2}|\,,\end{aligned}$$

from which (5.5) can be deduced from (5.4), after interchanging (A^2,b^2,u^2) with (A^1,b^1,u^1). □

We finish with a simple example exhibiting a bad behavior of the multifunction $u\mapsto S_{A,b,u}$ in a case in which assumption (5.3) is not satisfied.

Example 5.1. In [21], Mangasarian and Shiau gave the following simple example. Consider in $\mathbb{R}^2$ the vectors

$$a_1=\begin{pmatrix}-1\\0\end{pmatrix},\quad a_2=\begin{pmatrix}0\\-1\end{pmatrix},\quad a_3=\begin{pmatrix}1\\1\end{pmatrix},\quad u_\delta=\begin{pmatrix}-1-\delta\\-1\end{pmatrix},$$

and the right member

$$b=\begin{pmatrix}0\\0\\1\end{pmatrix}.$$

We have $S_{A,b,u_0}=\mathrm{co}(-a_1,-a_2)$, while

$$S_{A,b,u_\delta}=\begin{cases}-a_1 & \text{if } \delta>0\,,\\ \\ -a_2 & \text{if } \delta<0\,.\end{cases}$$

Observe that assumption (5.3) is not satisfied in this case, since $0\in\mathrm{bdry}(\mathrm{co}(a_3\cup\{u_0\}))$.

Acknowledgment. We thank the anonymous referees for several comments that led us to improve the presentation of the paper and to enrich the reference list.

REFERENCES

[1] S. A. Ashmanov, *Stability conditions for linear programming problems*, Comput. Math. Math. Phys., 21 (1981), pp. 40–49.

[2] A. Auslender, R. Cominetti, and J.-P. Crouzeix, *Convex functions with unbounded level sets and applications to duality theory*, SIAM J. Optim., 3 (1993), pp. 669–687.

[3] D. Azé, J.-N. Corvellec, and R. E. Lucchetti, *Variational pairs and applications to stability in nonsmooth analysis*, Nonlinear Anal., to appear.

[4] C. Bergthaller and I. Singer, *The distance to a polyhedron*, Linear Algebra Appl., 169 (1992), pp. 111–129.

[5] J. V. Burke and P. Tseng, *A unified analysis of Hoffman's bounds via Fenchel duality*, SIAM J. Optim., 6 (1996), pp. 265–282.

[6] J.-N. Corvellec, *A note on coercivity of lower semicontinuous functions and nonsmooth critical point theory*, Serdica Math. J., 22 (1996), pp. 57–68.

[7] J.-P. Dedieu, *Approximate solutions of analytic inequality systems*, SIAM J. Optim., 11 (2000), pp. 411–425.

[8] E. De Giorgi, A. Marino, and M. Tosques, *Problemi di evoluzione in spazi metrici e curve di massima pendenza*, Atti Accad. Naz. Lincei Cl. Sci. Fis. Mat. Natur. Rend. Lincei (9) Mat. Appl., 68 (1980), pp. 180–187.

[9] S. Deng, *Perturbation analysis of a condition number for convex inequality systems and global error bounds for analytic systems*, Math. Programming, 83 (1998), pp. 265–282.

[10] I. Ekeland, *Nonconvex minimization problems*, Bull. Amer. Math. Soc. (N.S.), 1 (1979), pp. 443–474.

[11] O. Güler, A. J. Hoffman, and U. G. Rothblum, *Approximations to solutions to systems of linear inequalities*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 688–696.

[12] A. Hantoute, *General Stability in Linear and Convex Quadratic Programming*, Technical report, University Paul Sabatier, Toulouse, France, 2000.

[13] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms*, Vol. 1 and 2, Springer-Verlag, Berlin, New York, 1993.

[14] A. J. Hoffman, *On approximate solutions of systems of linear inequalities*, J. Res. Nat. Bur. Standards, 49 (1952), pp. 263–265.

[15] A. Ioffe, *Regular points of Lipschitz functions*, Trans. Amer. Math. Soc., 251 (1979), pp. 61–69.

[16] D. Klatte and G. Thiere, *A note on Lipschitz constants for solutions of linear inequalities and equations*, Linear Algebra Appl., 244 (1996), pp. 365–374.

[17] A. S. Lewis and J. S. Pang, *Error bounds for convex inequality systems*, in Generalized Convexity, Generalized Monotonicity: Recent Results, J.-P. Crouzeix, J.-E. Martinez-Legaz, and M. Volle, eds., Nonconvex Optim. Appl. 27, Kluwer Academic Publishers, 1998, pp. 75–110.

[18] W. Li, *The sharp Lipschitz constant for feasible and optimal solutions of a perturbed linear program*, Linear Algebra Appl., 187 (1993), pp. 15–40.

[19] W. Li and I. Singer, *Global error bounds for convex multifunctions and applications*, Math. Oper. Res., 23 (1998), pp. 443–462.

[20] Z.-Q. Luo and P. Tseng, *Perturbation analysis of a condition number for linear systems*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 636–660.

[21] O. L. Mangasarian and T.-H. Shiau, *Lipschitz continuity of solutions of linear inequalities, programs and complementarity problems*, SIAM J. Control. Optim., 25 (1987), pp. 583–595.

[22] S. M. Robinson, *A characterization of stability in linear programming*, Oper. Res., 25 (1977), pp. 435–447.

[23] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1967.

[24] A. Schrijver, *Theory of Linear and Integer Programming*, Wiley, New York, 1986.

SIAM J. OPTIM.
Vol. 12, No. 4, pp. 928–948

A COMPLEMENTARY PIVOTING APPROACH TO THE MAXIMUM WEIGHT CLIQUE PROBLEM*

ALESSIO MASSARO†, MARCELLO PELILLO‡, AND IMMANUEL M. BOMZE§

Abstract. Given an undirected graph with positive weights on the vertices, the maximum weight clique problem (MWCP) is to find a subset of mutually adjacent vertices (i.e., a clique) having largest total weight. The problem is known to be NP-hard, even to approximate. Motivated by a recent quadratic programming formulation, which generalizes an earlier remarkable result of Motzkin and Straus, in this paper we propose a new framework for the MWCP based on the corresponding linear complementarity problem (LCP). We show that, generically, all stationary points of the MWCP quadratic program exhibit strict complementarity. Despite this regularity result, however, the LCP turns out to be inherently degenerate, and we find that Lemke's well-known pivoting method, equipped with standard degeneracy resolution strategies, yields unsatisfactory experimental results. We exploit the degeneracy inherent in the problem to develop a variant of Lemke's algorithm which incorporates a new and effective "look-ahead" pivot rule. The resulting algorithm is tested extensively on various instances of random as well as DIMACS benchmark graphs, and the results obtained show the effectiveness of our method.

Key words. maximum weight clique, linear complementarity, pivoting methods, quadratic programming, combinatorial optimization, heuristics

AMS subject classifications. 90C27, 90C20, 90C33, 90C49, 90C59, 05C69

PII. S1052623400381413

1. Introduction.

Given an undirected graph, the maximum clique problem (MCP) consists of finding a subset of pairwise adjacent vertices (i.e., a *clique*) having largest cardinality. The problem is known to be NP-hard for arbitrary graphs and, according to recent theoretical results, so is the problem of approximating it within a constant factor. An important generalization of the MCP arises when positive weights are associated to the vertices of the graph. In this case the problem is known as the maximum weight clique problem (MWCP) and consists of finding a clique in the graph which has largest total weight. (Note that the maximum weight clique does not necessarily have largest cardinality.) It is clear that the classical unweighted version is a special case in which the weights assigned to the vertices are all equal. As an obvious corollary, the MWCP has at least the same computational complexity as its unweighted counterpart. The MWCP has important applications in such fields as computer vision, pattern recognition, and robotics, where weighted graphs are employed as a convenient means of representing high-level pictorial information (see, e.g., [17, 28]). We refer to [4] for a recent review concerning algorithms, applications, and complexity issues of this important problem.

Inspired by a classical result in graph theory contributed by Motzkin and Straus [24], Gibbons et al. [13] have recently formulated the MWCP in terms of a *standard quadratic optimization problem* (StQP), which consists of minimizing a quadratic form

*Received by the editors November 17, 2000; accepted for publication (in revised form) May 21, 2001; published electronically March 13, 2002.
http://www.siam.org/journals/siopt/12-4/38141.html
†FEI Electron Optics, SEM Software Group, Building HA F13, Postbus 218, 5600MD Eindhoven, The Netherlands (alessio.massaro@nl.feico.com).
‡Dipartimento di Informatica, Università Ca' Foscari di Venezia, Via Torino 155, I-30172 Venezia Mestre, Italy (pelillo@dsi.unive.it).
§Institut für Statistik und Decision Support Systems, Universität Wien, Universitätsstraße 5, A-1010 Wien, Austria (immanuel.bomze@univie.ac.at).

over the standard simplex [3]. As shown in [7], however, their original formulation suffers from the presence of "spurious" solutions, namely, solutions of the continuous problem that are not in one-to-one correspondence with solutions in the original combinatorial problem. To avoid this drawback, in [3, 7] a new regularized quadratic programming formulation is proposed in which local and global solutions are characterized in terms of cliques of maximal and maximum weight, respectively, and no spurious solutions exist. A further benefit of this modified formulation, as we will show in this paper, is that generically all of its Karush–Kuhn–Tucker (KKT) points exhibit strict complementarity. This is a regularity property which not only favors numerical stability but also plays an important role in simplifying (second-order) optimality conditions.

It is well known that KKT points of quadratic optimization problems with linear constraints, like StQPs, can be characterized as the solutions of a linear complementarity problem (LCP), a class of inequality systems for which a rich theory and a large number of algorithms have been developed [11]. Hence, once the MWCP is formulated in terms of an StQP, the use of LCP algorithms naturally suggests itself, and this is precisely the main idea proposed in the present paper. Among the many LCP methods presented in the literature, pivoting procedures are widely used, and within this class Lemke's method is certainly the best known. Unfortunately, like other pivoting schemes, its finite convergence is guaranteed only for nondegenerate problems, and ours is indeed degenerate. To avoid this drawback, we incorporated standard degeneracy resolution strategies into Lemke's "Scheme I" procedure and tested it over a number of DIMACS benchmark graphs, but the computational results obtained were rather discouraging. The inherent degeneracy of the problem, however, is beneficial as it leaves freedom in choosing the blocking variable, and we exploit this property to develop a variant of Lemke's algorithm which uses a new and effective "look-ahead" pivot rule. The procedure depends critically on the choice of a vertex in the graph which identifies the second blocking variable in the pivoting process. Since there is no obvious way to determine such a vertex in an optimal manner, we resort to iterating this procedure over most, if not all, vertices in the graph. Also, upon analyzing the overall behavior of our heuristic, we obtain a number of invariants which are exploited to reduce the amount of data and the complexity of certain operations needed to process the problem.

The paper is organized as follows. In section 2 we review and investigate the reformulation of the MWCP as an StQP such that maximal cliques correspond to local solutions, and vice versa. Further, we establish that, for an open and dense set of weights, for a given graph all KKT points are strictly complementary. The relevance of this property becomes even more obvious in light of the discussion of second-order optimality conditions for StQPs, which we include for background information in an appendix. In the present context it is important to discriminate between strict complementarity (as a sort of "geometric" regularity condition) and the LCP degeneracy (which can be viewed as "algebraic"). The latter is shown to be inherent in the LCPs emerging from our MWCP in section 3, where we also describe our pivoting-based heuristic. Section 4 contains experimental findings. We test our approach on unweighted DIMACS benchmark graphs and various types of randomly generated weighted graphs. The results obtained show the effectiveness of our method and its clear superiority compared to other continuous-based heuristics. It also compares well with other state-of-the-art (non–continuous-based) heuristics presented in the literature.

2. Continuous formulation of the MWCP.

2.1. Basic theory. Let $G = (V, E, w)$ be an arbitrary undirected and weighted graph, where $V = \{1, \ldots, n\}$ is the vertex set and $E \subseteq \binom{V}{2}$ is the edge set, $\binom{V}{2}$ denoting the system of all two-element subsets of V. Further, $w \in \mathbb{R}^n$ is the *weight* vector, the ith component of which corresponds to the weight assigned to vertex i. It is assumed that $w_i > 0$ for all $i \in V$. Two distinct vertices $i, j \in V$ are said to be *adjacent* if they are connected by an edge, i.e., if $\{i, j\} \in E$. The *neighborhood* of a vertex i will be indicated with $N(i) = \{j \in V : \{i, j\} \in E\}$, and its degree will be $\deg(i) = |N(i)|$, the cardinality of $N(i)$. Given a subset of vertices S, the weight assigned to S will be denoted by

$$W(S) = \sum_{i \in S} w_i.$$

As usual, the sum over the empty index set is defined to be zero.

A *clique* is a subset of V in which all vertices are pairwise adjacent. A clique S is called *maximal* if no strict superset of S is a clique. A maximal weight clique S is a clique which is not contained in any other clique having weight larger than $W(S)$. Since we are assuming that all weights are positive, it is clear that the concepts of maximal and maximal weight clique coincide; hence we shall not make any distinction between these throughout the paper. A maximum cardinality clique (or, simply, a *maximum clique*) is a clique whose cardinality is the largest possible. The maximum size of a clique in G is called the *clique number* (of G) and is denoted by $\omega(G)$. A *maximum weight clique* is a clique having largest total weight, and the maximum weight clique problem (MWCP) is the problem of finding such a clique. The *weighted clique number* of G, denoted by $\omega(G, w)$, is the maximum weight of a clique in G.

Let $G = (V, E)$ be an undirected (unweighted) graph, and let Δ denote the standard simplex in the n-dimensional Euclidean space $\mathbb{R}^n$:

$$\Delta = \left\{x \in \mathbb{R}^n : x_i \geq 0 \text{ for all } i \in V,\ e^T x = 1\right\},$$

where e is a vector of appropriate length, consisting of unit entries. (Hence $e^T x = \sum_{i \in V} x_i$.) We will also denote by e_i the ith column of the $n \times n$ identity matrix I_n.

Now consider the following quadratic function, which is sometimes called the *Lagrangian* of G:

$$g(x) = x^T A_G x = \sum_{\{i,j\} \in E} x_i x_j,$$

where $A_G = (a_{ij})_{i,j \in V}$ is the adjacency matrix of G—i.e., $a_{ij} = 1$ if $\{i, j\} \in E$, and $a_{ij} = 0$ if $\{i, j\} \notin E$—and let x^* be a global maximizer of g in Δ. Motzkin and Straus [24] showed that the clique number $\omega(G)$ of G is related to $g(x^*)$ according to the following formula:

$$\omega(G) = \frac{1}{1 - g(x^*)}.$$

Additionally, they proved that a subset of vertices S is a maximum clique of G if and only if its *characteristic vector* x^S, which is the vector in Δ defined by $x_i^S = 1/|S|$ if $i \in S$ and $x_i^S = 0$ otherwise, is a global maximizer of g in Δ.[1]

[1] Actually, in their original paper, Motzkin and Straus proved just the "only if" part of this theorem. The converse direction is, however, a straightforward consequence of their result [27].

Gibbons et al. [13] have generalized the Motzkin–Straus theorem to the weighted case. Given a weighted graph $G = (V, E, w)$, they introduced the concept of the *weighted characteristic vector* $x^{S,w} \in \Delta$ for a given vertex-set $S \subseteq V$, whose coordinates are

$$x_i^{S,w} = \begin{cases} \frac{w_i}{W(S)} & \text{if } i \in S, \\ 0 & \text{otherwise.} \end{cases}$$

Using a proof technique suggested by Lovász, they reformulated the Motzkin–Straus problem as a minimization problem and extended the correspondence between global minimizers that have the form of weighted characteristic vectors and the maximum weight cliques of G. Their results were proved over a whole class of matrices, rather than just a single matrix as in the original Motzkin–Straus formulation. Of course, both the latter and the matrix class considered in [13] depend on G.

However, the formulation of Gibbons et al. has a major drawback which, as in the unweighted case [27], relates to the presence of "spurious" solutions, i.e., local or global solutions that are not in the form of weighted characteristic vectors $x^{S,w}$ for some subset S of vertices. (See [7] for an in-depth study on this topic.) Even though in certain specific circumstances such solutions may provide useful information concerning the structure of the underlying graph, computationally they represent a nuisance, for we cannot extract the vertices comprising the clique directly from them; they just provide information about the weighted clique number.

This problem is solved in [3] by considering the matrix $Q_G = [q_{ij}]_{i,j \in V \times V}$ defined as

$$q_{ij} = \begin{cases} \frac{1}{2w_i} & \text{if } i = j, \\ 0 & \text{if } \{i, j\} \in E, \\ \frac{1}{2w_i} + \frac{1}{2w_j} & \text{otherwise} \end{cases} \tag{1}$$

and investigating the StQP

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \Delta \end{array} \tag{2}$$

with

$$f(x) = x^T Q_G x. \tag{3}$$

Indeed, a whole class of matrices serves the same purpose again; this class, of course, differs from that used in [13].

The following theorem proved in [3] summarizes the result around which our work is centered.

THEOREM 2.1. *Let $G = (V, E, w)$ be an arbitrary graph with positive weight vector $w \in \mathbb{R}^n$, and consider problem* (2)*. Then the following assertions hold.*

- *A vector $x \in \Delta$ is a local solution of* (2) *if and only if $x = x^{S,w}$, where S is a maximal clique of G.*
- *A vector $x \in \Delta$ is a global solution of* (2) *if and only if $x = x^{S,w}$, where S is a maximum weight clique of G.*

Moreover, all solutions of (2) *are strict.*

2.2. From KKT points to maximal cliques. It is a simple exercise to show that the KKT first-order optimality conditions for a point $x \in \Delta$ in program (2), with a general symmetric $n \times n$ matrix Q in place of Q_G, can be written as

$$(Qx)_i \begin{cases} = \lambda & \text{if } x_i > 0, \\ \geq \lambda & \text{if } x_i = 0, \end{cases} \tag{4}$$

for some real-valued constant λ.

Of course, a KKT point is not necessarily a local solution of (2), and hence a maximal clique of G, but in light of (4) it is possible to derive some useful properties that virtually eliminate the need to guarantee local optimality and give direct methods to attain it once a KKT point is available. For $x \in \Delta$, let us denote by

$$S(x) = \{i \in V : x_i > 0\}$$

the support of x.

THEOREM 2.2. *Let $G = (V, E, w)$ be a weighted graph and x a KKT point of (2). If $C = S(x)$ is a clique of G, then C is a maximal clique.*

Proof. If $C = S(x)$ is a nonmaximal clique, then there exists a $k \notin C$ such that $(i, k) \in E$ for all $i \in C$. For such a k we have

$$(Q_G x)_k = \sum_{j \in C} q_{kj} x_j = 0.$$

On the other hand, for any $i \in C$, we have

$$(Q_G x)_i = \sum_{j \in C} q_{ij} x_j = q_{ii} x_i > 0,$$

which contradicts the hypothesis that x is a KKT point for (2). □

The practical significance of Theorem 2.2 reveals itself in large graphs: Even if these are quite dense, cliques are usually much smaller than the graph itself. Now suppose we are returned a KKT point x by some method. Then we set $C = S(x)$ and check whether or not C is a clique. This requires $\mathcal{O}(s^2)$ steps if C contains s vertices, while checking whether this clique is maximal would require $\mathcal{O}(sn)$ steps and, as stressed above, usually $s \ll n$. But Theorem 2.2 now guarantees that the obtained clique C (if it is one) must automatically be maximal, and thus we are spared from trying to add external vertices. But how should one behave in the case of a nonclique KKT point? The answer is to be found in part of the proof of Theorem 5 in [13] and is summarized in the following result.

THEOREM 2.3. *Let $G = (V, E, w)$ be a weighted graph and x a KKT point of (2) with support $C = S(x)$. If $i, j \in V$ are two nonadjacent vertices of C, then $x_\delta = x + \delta(e_i - e_j)$ improves the objective function f of (2); i.e., $f(x_\delta) < f(x)$ for any $0 < \delta \leq x_j$.*

Proof. From the symmetry of Q_G, we have

$$\begin{aligned} f(x_\delta) &= (x + \delta(e_i - e_j))^T Q_G (x + \delta(e_i - e_j)) \\ &= x^T Q_G x + 2\delta(e_i - e_j)^T Q_G x + \delta^2 (e_i - e_j)^T Q_G (e_i - e_j). \end{aligned}$$

Since x is a KKT point, the second term is null, and hence

$$f(x_\delta) = f(x) + \delta^2 (q_{ii} + q_{jj} - 2q_{ij}).$$

But $\{i,j\} \notin E$ implies that $q_{ii} + q_{jj} - 2q_{ij} < 0$, and this concludes the proof. ∎

Once a KKT point has been obtained by some method, the most effective way to use Theorem 2.3 is to check for pairs $\{i,j\} \notin E$ with $\{i,j\} \subseteq C$. If there are none, C is a clique and hence a maximal clique. Otherwise, choose any pair of nonadjacent vertices in C and construct a new "better" point as described in Theorem 2.3. The proof of the theorem also provides us with a criterion to determine the best such points, namely, the one which minimizes $x_j^2\,(q_{ii} + q_{jj} - 2q_{ij})$. This can be done very quickly in $\mathcal{O}\left(\binom{m}{2}\right)$ time, where $m = |C|$.

Clearly, the new improved point does not necessarily correspond to a (maximal) clique, but by iterating this procedure, as suggested in [20] for the unweighted case, we can readily obtain one. Alternatively, and more interestingly, one can give the new improved point as input to any gradient-based technique. These are typically very efficient in terms of computation time and can be quite effective if kick-started from within a close range to a good suboptimal solution. An example of such techniques is given by the so-called replicator dynamics, a class of dynamical systems developed and studied in evolutionary game theory [15]. We refer to [26] for a recent review concerning the application of these dynamics to combinatorial optimization, and to [22, 23] for independent connections between this kind of dynamical equations and LCPs.

2.3. Strict complementarity is generic. We close this section by establishing easy-to-check regularity conditions for the StQP (2) based on the matrix Q_G, which ensure the strict complementarity of all KKT points of this StQP. It turns out that, when we fix the discrete structure (V,E) of the graph in an arbitrary way, strict complementarity holds for a set of weights w that is an open and dense subset of the positive orthant $\mathbb{R}^n_+$. Recall that a KKT point x satisfies the *strict complementarity condition* for the StQP (2), with a general symmetric $n \times n$ matrix Q in place of Q_G, if and only if all Lagrange multipliers are strictly positive: $\lambda_i > 0$ for all $i \in V \setminus S(x)$, where $\lambda_i = (Qx)_i - \lambda$ from (4).

We now characterize strict complementarity and establish easy-to-check sufficient conditions.

THEOREM 2.4. *Let $x \in \Delta$ be a KKT point for (2), and again set $S(x) = \{i \in V : x_i > 0\}$ as well as $T(x) = \{i \in V : (Qx)_i = x^TQx\}$. Then $S(x) \subseteq T(x)$. Further, the following assertions are equivalent:*

(a) *$S(x) = T(x)$ (which in particular holds true if $S(x) = V$);*

(b) *x satisfies the strict complementarity condition.*

Both conditions are met if for all $i \in V \setminus S(x)$ the matrices

$$Q_{S(x)}(i) = [q_{kj} - q_{ij}]_{(k,j)\in S(x)\times S(x)}$$

are nonsingular.

Proof. The inclusion $S(x) \subseteq T(x)$ is nothing other than (4); indeed, it easily follows that $\lambda = x^TQx$. Further, we also get $0 \le \lambda_i = (Qx)_i - \lambda = (Qx)_i - x^TQx$, from which the equivalence of (a) and (b) is immediate. Finally, suppose that there is an index $i \in T(x) \setminus S(x) \subseteq V \setminus S(x)$. Then we get $Q_{S(x)}(i)x_{S(x)} = [(Qx)_k - (Qx)_i]_{k\in S(x)} = o$ while $x_{S(x)} = [x_k]_{k\in S(x)} \neq o$, contradicting the assumption. ∎

Now we are ready to establish the main result for matrix $Q = Q_G$ used in the MWCP treatment; for almost all weights w in a given graph G, every KKT point has this property. We also specify simple explicit sufficient conditions which guarantee this.

THEOREM 2.5. *Let $G = (V, E, w)$ be a weighted graph and suppose that $w = \mu z$, where $\mu > 0$ and $z_i > 0$ are odd integers for all $i \in V$. Then, regardless of the structure of G, all the matrices $Q_{S(x)}(i)$ originating from the matrix $Q = Q_G$ are nonsingular, and hence all KKT points x for (2) satisfy strict complementarity.*

Further, the set of all weights such that the nonsingularity condition (and thus strict complementarity) holds is open and dense in $\mathbb{R}^n_+$.

Proof. As is easily seen, the entries of $Q_{S(x)}(i)$ all are sums of two terms belonging to the set $\{0, \pm\frac{1}{2w_j} : j \in V\}$. Hence multiplication of the weights w_j by a common factor μ does not alter any aspect of the assertion. Thus the result holds for all μ, given that we establish it for a special value of it, e.g., for $\mu = [2\prod_{i\in V} z_i]^{-1}$. But then $Q_{S(x)}(i)$ has odd integer diagonal entries while all other entries are even integers. Thus the determinant is an odd number, whence it follows that $Q_{S(x)}(i)$ is nonsingular. Turning to the genericness assertion, openness is clear from the continuity of the determinant, while denseness follows from an approximation argument; indeed, every positive w can be arbitrarily well approximated by a vector with positive rational entries n_i/d, where n_i and d are positive integers. Next choose an integer K large enough such that these ratios n_i/d in turn are close to $\tilde{w}_i = \mu(2Kn_i + 1)$ with $\mu = [2Kd]^{-1}$. Now $\tilde{w}$ satisfies the first condition of the theorem, and the result follows. □

Note that the result applies particularly to the nonweighted case; in fact, $w = e$ satisfies the first condition in the above theorem.

In spite of these results, namely, that this "geometric" form of degeneracy is highly unlikely, we will see in the next section that a sort of "algebraic" degeneracy is inherent to the problem class considered here. To promote the flow of the argument, we defer to an appendix a discussion of further aspects of strict complementarity in relation to the optimality condition.

3. Complementary pivoting.

3.1. Lemke's method. The KKT points of (2) can be computed by solving the LCP (q_G, M_G), which is the problem of finding a vector x satisfying the system

$$y = q_G + M_G x \geq 0, \quad x \geq 0, \quad x^T y = 0, \tag{5}$$

where

$$q_G = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -1 \\ 1 \end{bmatrix}, \quad M_G = \begin{bmatrix} Q_G & -e & e \\ e^T & 0 & 0 \\ -e^T & 0 & 0 \end{bmatrix}, \tag{6}$$

and Q_G is as in (1). With the above definitions, it is well known that if z is a complementary solution of (q_G, M_G) with $z^T = [x^T, y^T]$ and $x \in \mathbb{R}^n$, then x is a KKT point of (2). Note that Q_G is strictly $\mathbb{R}^n_+$-copositive; hence so is M_G, and this is sufficient to ensure that (q_G, M_G) always has a solution—see the fundamental book [11], where a large number of LCP algorithms can also be found. The most popular among them is probably Lemke's method, largely for its ability to provide a solution for several matrix classes. Lemke's "Scheme I" belongs to the family of pivoting algorithms. Given the generic LCP (q, M), it deals with the augmented problem

(q, d, M) defined by

$$y = q + [M, d] \begin{bmatrix} x \\ \theta \end{bmatrix} \geq 0, \quad \theta \geq 0, \quad x \geq 0, \quad x^T y = 0. \tag{7}$$

Vector d is called the *covering vector* and must satisfy $d_i > 0$ whenever $q_i < 0$. A solution of (q, d, M) with $\theta = 0$ promptly yields a solution for (q, M), and Lemke's method intends to compute precisely such a solution. We refer to [11] for a detailed description of Lemke's algorithm. In our implementation we chose $d = e$, as our problem does not expose peculiarities that would justify a deviation from this common practice.

Assuming the nondegeneracy of the LCP is a strategy commonly taken to prove the finiteness of pivoting schemes. In particular, Lemke's method is guaranteed to process any nondegenerate problem (q, M), where M is strictly $\mathbb{R}^n_+$-copositive, and to do so without terminating on a secondary ray [11].

Unfortunately (q_G, M_G) is degenerate, but it is possible to give an equivalent formulation of (2) in order to obtain a nondegenerate LCP. To this end, it is easy to see that program (2) is equivalent to the following program:

$$\begin{array}{ll} \text{minimize} & x^T \widehat{Q}_G x + c^T x \\ \text{subject to} & x \in \Delta, \end{array}$$

where $c \in \mathbb{R}^n$ and $\widehat{Q}_G = Q_G - \left(ce^T + ec^T\right)$. If $c \leq 0$, then copositivity is maintained, and if all its entries are different, then the corresponding LCP is nondegenerate. Furthermore, if $c_i \leq -1$ for some i, it is straightforward to check that even the first pivot step of Lemke's method changes.

The above method for degeneracy removal has a characteristic in common with the lexicographic degeneracy resolution method (LDR) [11]. Namely, they both require the introduction of extra data: vector c in the previous case, and a nonsingular square matrix as big as M with lexicographically positive rows in the case of LDR. Of course, these objects have to be assigned values and there are myriads of sensible methods for doing so, each one having a different theoretical ground and/or performance impact on the final result.

We report in Tables 1 and 2 (column *LDR*) the results obtained on the DIMACS benchmark graphs (see subsection 4.1) by running Lemke's "Scheme I" with LDR, using the identity matrix as extra data. Their order, density, and clique number are shown in columns *Order*, *Density*, and ω, respectively. A complete description of the table can be found in subsection 4.1. It is clear to see that in all but the most trivial cases LDR performs poorly, although it is extremely fast. The tendency of the previously discussed degeneracy treatment methods is that they lead to inefficient local minimizers, i.e., to maximal cliques of small size.

3.2. A pivoting-based heuristic. Rather than continuing to investigate the enormous variety of assignment techniques for removing degeneracy mentioned in the previous subsection, we focused on examining the original degenerate form of the LCP (q_G, M_G). Such degeneracy even turns out to be beneficial for performance, since it permits freedom in choosing the blocking variable within a successful variant of Lemke's method. This is opposed to the nondegenerate version of the latter method, in which those variables are uniquely determined. This is the topic of the present subsection.

As is customary, we will use an exponent for the problem data and, to make notations simpler, we will omit subscripts indicating the dependence on graph G. Hence, q^ν and M^ν will identify the situation after ν pivots, and Q^ν will indicate the $n \times n$ leading principal submatrix of M^ν. Consistently, y^ν and x^ν will indicate the vectors of basic and nonbasic variables, respectively, each made up of a combination of the original x_i and y_i variables. The notation $\langle x_i^\nu, y_j^\nu \rangle$ will be used to indicate pivoting transformations. The index set of the basic variables that satisfy the min-ratio test at iteration ν will be denoted with Ω^ν, i.e.,

$$\Omega^\nu = \arg\min_i \left\{ \frac{-q_i^\nu}{m_{is}^\nu} : m_{is}^\nu < 0 \right\},$$

where s is the index of the driving column. Also, in what follows, the auxiliary column that contains the covering vector d in (7) will be referred to as the column $n+3$ of matrix $M = M_G$. The nondegeneracy assumption basically amounts to having $|\Omega^\nu| = 1$ for all ν, thereby excluding any cycling behavior.

Here we employ the least-index rule, which amounts to blocking the driving variable with a basic one that has minimum index within a certain subset of Ω^ν, i.e., $r = \min \Phi^\nu$ for some $\Phi^\nu \subseteq \Omega^\nu$. The set Φ^ν is chosen in order to make the number of degenerate variables decrease as slowly as possible, i.e., among the index-set

$$\Phi^\nu = \arg\min_i \left\{ |\Omega^\nu| - \left|\Omega_i^{\nu+1}\right| > 0 : i \in \Omega^\nu \right\} \subseteq \Omega^\nu,$$

where $\Omega_i^{\nu+1}$ is the index-set of those variables that would satisfy the min-ratio test at iteration $\nu + 1$ if the driving variable at iteration ν were blocked with y_i^ν as $i \in \Omega^\nu$. The previous conditional implies that a pivot step is taken and then reset in a sort of "look-ahead" fashion; hence we will refer to this rule as the *look-ahead (pivot) rule*.

Before actually proceeding to illustrate a variant of Lemke's algorithm applied to the MWCP, let us take a look at the tableaus that it generates. This will help us to identify regularities that are reflected in the behavior of the algorithm itself. The initial tableau follows:

(8)

	q	x_1	$\cdots$	x_n	x_{n+1}	x_{n+2}	θ
y_1	0				-1	1	1
$\vdots$	$\vdots$		Q_G		$\vdots$	$\vdots$	$\vdots$
y_n	0				-1	1	1
y_{n+1}	-1	1	$\cdots$	1	0	0	1
y_{n+2}	1	-1	$\cdots$	-1	0	0	1

As q_{n+1} is the only negative entry for the column of q, the first pivot to occur during initialization is $\langle y_{n+1}, \theta \rangle$, thereby producing the following transformation:

(9)

	q	x_1	$\cdots$	x_n	x_{n+1}	x_{n+2}	y_{n+1}
y_1	1				-1	1	1
$\vdots$	$\vdots$		$Q_G - ee^T$		$\vdots$	$\vdots$	$\vdots$
y_n	1				-1	1	1
θ	1	-1	$\cdots$	-1	0	0	1
y_{n+2}	2	-2	$\cdots$	-2	0	0	1

The driving variable for the second pivot is x_{n+1}. Since $m_{i,n+1}^1 = -1$ for all $i = 1, \ldots, n$, it is clear to see that the relative blocking variable can be any one of

ALGORITHM 3.1. LEMKE'S "SCHEME I" WITH THE "LOOK-AHEAD" RULE APPLIED TO THE MWCP.

Input: A graph $G = (V, E, w)$ and $p \in V$.
Let (q_G, e, M_G) be the augmented LCP, where q_G and M_G are defined in (6).
$\nu \leftarrow 0$, perform $\langle y_{n+1}, \theta \rangle$ and $\langle y_p, x_{n+1} \rangle$.
The driving variable is x_p.
Infinite loop
  $\nu \leftarrow \nu + 1$.
  Let x_s^ν denote the driving variable.
  $\Omega^\nu = \arg\min_i \{-q_i^\nu / m_{is}^\nu \ : m_{is}^\nu < 0\}$.
  If $|\Omega^\nu| = 1$, then $r = \min \Omega^\nu$;
  else $\Phi^\nu = \arg\min_i \{|\Omega^\nu| - |\Omega_i^{\nu+1}| > 0 : i \in \Omega^\nu\}$, $r = \min \Phi^\nu$.
  Perform $\langle y_r^\nu, x_s^\nu \rangle$.
  If $y_r^\nu \equiv \theta$, then:
    Let x denote the complementary solution of (q_G, M_G) found.
    The result is $\operatorname{supp}(x) \cap V$.
  The new driving variable is the variable complementary to y_r^ν.

$y_1, \ldots, y_n$. In this case we apply no degeneracy resolution criterion but rather allow for user intervention by catering for the possibility of deciding the second blocking variable a priori. Thus let y_p be the (arbitrary) variable that shall block x_{n+1}. After performing $\langle y_p, x_{n+1} \rangle$, we have the following tableau:

(10)

	q	x_1	$\cdots$	x_n	y_p	x_{n+2}	y_{n+1}
y_1	0				1	0	0
$\vdots$	$\vdots$				$\vdots$	$\vdots$	$\vdots$
y_{p-1}	0				1	0	0
x_{n+1}	1		Q_p		-1	1	1
y_{p+1}	0				1	0	0
$\vdots$	$\vdots$				$\vdots$	$\vdots$	$\vdots$
y_n	0				1	0	0
θ	1	-1	$\cdots$	-1	0	0	1
y_{n+2}	2	-2	$\cdots$	-2	0	0	1

,

where Q_p denotes the matrix whose rows are defined as

$$(Q_p)_i = \begin{cases} (Q_G)_p - e^T & \text{if } i = p, \\ (Q_G)_i - (Q_G)_p, & \text{otherwise.} \end{cases}$$

Algorithm 3.1 formalizes the above statements. We now introduce a number of invariants aimed at reducing the size of the data required by the process and the complexity of certain operations.

PROPOSITION 3.1. *Within Algorithm* 3.1*, after the first pivot and as long as none occurs within the last* 2 *rows of* M^ν*, the ratios* $m_{n+1,j}^\nu / m_{n+2,j}^\nu = q_{n+1}^\nu / q_{n+2}^\nu = \frac{1}{2}$ *for* $j = 1, \ldots, n$ *do not change.*

Proof. The proof is elementary by the definition of pivot operation and the structure of tableau (9). □

COROLLARY 3.2. *In Algorithm* 3.1*, in the event that after* ν *pivot operations the*

whole driving column of Q^ν be nonnegative, the schema will pivot on the row of θ and terminate.

Proof. The proof follows immediately from the fact that termination on the secondary ray cannot occur, and from Proposition 3.1. □

PROPOSITION 3.3. *After* 2 *pivot operations within Algorithm* 3.1*, the columns of q, x_{n+2}, y_{n+1} do not change as long as no pivot is performed on the rows of x_{n+1}, θ, or y_{n+2}.*

Proof. The proof follows from tableau (10), observing that the hypothesis implies that either m^ν_{is}, m^ν_{rj}, or q^ν_r is null when calculating the successive transforms of these columns. □

COROLLARY 3.4. *After* 2 *pivot operations within Algorithm* 3.1*, if a pivot on the row of $x_{n+1} \equiv y^\nu_r$ occurs with x^ν_s as the driving variable, then $m^\nu_{i,s} \geq 0$ for all $i = 1, \ldots, r-1, r+1, \ldots, n$. Moreover, if $m^\nu_{n+1,s} < 0$, then $m^\nu_{n+1,s} \geq m^\nu_{r,s}$.*

Proof. If there were other negative entries for $i = 1, \ldots, r-1, r+1, \ldots, n$ for Proposition 3.3, they would have a null ratio. On the other hand, the ratio for the row of x_{n+1} is certainly positive. A similar argument proves the remaining part of the corollary. □

PROPOSITION 3.5. *After* 2 *pivot operations within Algorithm* 3.1*, pivoting on the row of x_{n+1} ends the schema with the pivot sequence $\langle x_{n+1}, x^\nu_s \rangle$, $\langle y_{n+2}, y_{n+1} \rangle$, $\langle \theta, x_{n+2} \rangle$.*

Proof. After $\langle x_{n+1}, x^\nu_s \rangle$, for Proposition 3.1 and Corollary 3.4 we have $m^\nu_{i,n+2} = m^\nu_{i,n+3} \geq 0$ for all $i = 1, \ldots, n$. Corollary 3.4 yields $m^\nu_{n+1,n+3} = 1 - m^\nu_{n+1,s}/m^\nu_{r,s} \geq 0$, and this, together with the fact that no secondary ray termination can occur, implies $m^\nu_{n+2,n+3} < 0$, thereby indicating $\langle y_{n+2}, y_{n+1} \rangle$ as the following pivot. Similar arguments prove the remaining part of the proposition. □

The above statements show that the $x_1, \ldots, x_n$ variables remain within the Q^ν block for the whole duration of Algorithm 3.1. Furthermore, we do not need to perform the terminal pivot sequence of Proposition 3.5 for, as soon as x_{n+1} blocks the driving variable, we know which of the x_i with $i \in V$ will be basic, and that is enough to compute the final clique. This is sufficient to derive that the rows and columns associated with the simplex constraints and the covering vector are not needed to process (q_G, M_G). On top of that, for Proposition 3.3 we can also discard the vector q and reduce the min-ratio test to a mere negativity test. All these concepts are formalized in Algorithm 3.2.

Empirical evidence indicated p as a key parameter for the quality of the final result of Algorithm 3.2. Unfortunately we could not identify any effective means to restrict the choice of values in V that can guarantee a good suboptimal solution. We thus had to consider iterating for most, if not all, vertices of V as outlined in Algorithm 3.3. Here we employ a very simple criterion to avoid considering those nodes that cannot drive to larger cliques than the one we already have, because their weights and those of their neighborhoods are too small. It is easy to comment that such a criterion is effective only for very sparse graphs.

We also observed that the schema is sensitive to the ordering of nodes and found that the best figures were obtained by reordering G by the decreasing weight of each node and its neighborhood. This feature too is formalized in Algorithm 3.3. We will refer to this scheme by the name *pivoting-based heuristic* (PBH).

Before concluding this section, it is worth mentioning the fact that we were not able to prove that Algorithms 3.1 and 3.2 cannot terminate prematurely with an empty Φ^ν set, or to loop indefinitely with Ω^ν being a singleton. However, neither

ALGORITHM 3.2. A REDUCED VERSION OF ALGORITHM 3.1.

Input: A graph $G = (V, E, w)$ and $p \in V$.
Let $Q_p = (q_{ij})$. $\nu \leftarrow 2$. $K \leftarrow \emptyset$.
The driving variable is x_p.
Infinite loop
    Let x_s^ν denote the driving variable.
    $\Omega^\nu = \{i : q_{is}^\nu < 0\}$.
    If $\Omega^\nu \subseteq \{p\}$, stop: the result is K.
    $\Phi^\nu = \arg\min_i \left\{ |\Omega^\nu| - \left|\Omega_i^{\nu+1}\right| > 0 : i \in \Omega^\nu \right\}$.
    $r = \min \Phi^\nu$.
    If $y_r^\nu \equiv x_i$ for some i, then $K \leftarrow K \setminus \{i\}$.
    Perform $\langle y_r^\nu, x_s^\nu \rangle$.
    The new driving variable is the variable complementary to y_r^ν.
    $\nu \leftarrow \nu + 1$.
    If $y_r^\nu \equiv x_i$ for some i, then $K \leftarrow K \cup \{i\}$.

ALGORITHM 3.3. THE PIVOTING-BASED HEURISTIC (PBH) FOR THE MWCP.

Input: A graph $G = (V, E, w)$.
Let $G' = (V', E', w')$ be a permutation of G
with $W(u' \cup N(u')) \geq W(v' \cup N(v'))$ for all $u', v' \in V'$ with $u' < v'$.
$K^\star \leftarrow \emptyset$.
For $v' = 1, \ldots, n : W(v' \cup N(v')) > W(K^\star)$ do:
    Run Algorithm 3.2 with G' and v' as input.
    Let K be the obtained result.
    If $W(K) > W(K^\star)$, then $K^\star \leftarrow K$.
The result is the mapping of $K^\star$ in G.

of these circumstances ever actually occurred in practice. Instead, for all the several thousand graphs we tested them on, we observed that once an x_i variable with $i \in V$ had entered the basis, it never exited it. In fact, if Algorithm 3.1 found a clique with s nodes, it always performed exactly $s + 3$ pivot steps. This fact led us to consider a simplified implementation of Algorithm 3.2 which was in fact used to produce the results presented in the following section. This simplified version simply lacks the tests to remove an x variable from the basis. A thorough empirical analysis has confirmed that both the original and simplified versions of the algorithm behave identically.

Computing $|\Omega_i^{\nu+1}|$ can be done with $\mathcal{O}(n)$ time complexity, as only the driving column is needed for this purpose, and a pivotal transformation takes $\mathcal{O}(n^2)$ computations. This, together with our previous observation, gives us strong empirical evidence that PBH is $\mathcal{O}(sn^3)$, where s is the size of the clique found. Note, however, that it is quite straightforward to parallelize the algorithm over n processors, thereby reducing its time complexity to $\mathcal{O}(sn^2)$. With respect to space complexity, our implementation was $\mathcal{O}(n^2)$, as we could not find better techniques than implementing tableau-style pivoting.

4. Experimental results. To practically assess the effectiveness of the proposed approach, we conducted a large number of experiments. First, we focused on unweighted DIMACS graphs, which constitute a standard benchmark for clique-

finding heuristics [19].[2] Next, we considered various types of randomly generated weighted graphs.

4.1. Unweighted DIMACS graphs. Tables 1 and 2 show the performance figures obtained by running PBH (column *PBH*) over a selection of DIMACS benchmark graphs. Their order, density, and clique number are reported in columns *Order*, *Density*, and ω, respectively. The column marked with *LDR* lists the results pertaining to Lemke's method with the lexicographic degeneracy resolution criterion; see subsection 3.1. Computing time (column *Time*) for LDR as well as PBH is in seconds and refers to a C++ implementation for a Linux machine with a 655MHz Celeron CPU (77MHz FSB $\times$ 8.5). Some figures are missing because the Unix `clock` system call could not time periods longer than approximately 30 minutes on our test machine.

We compare our methods with three other heuristics based, as ours is, on the Motzkin–Straus formulation. The first method considered is the *continuous-based heuristic* (CBH) of Gibbons, Hearn, and Pardalos [12], which employs a parameterized version of the original Motzkin–Straus program. The problem is divided into a series of subproblems with the simplex constraints relaxed into spherical ones. Their schema uses a combinatorial postprocessing phase to round the solutions produced by a relaxation procedure that solves the subproblems.

The second algorithm is *annealed replication* (AR) [5]. It uses a different parameterized and unweighted maximization form of problem (2) that has $x^T (A_G + \alpha I)\, x$ as objective function. The heuristic uses the replicator dynamics as a local search technique and is based on a proper variation of α after a model similar to simulated annealing, but it is motivated by more principled arguments.

The third method is the *RD-algorithm* (RD), a recent heuristic of Kuznetsova and Strekalovsky [20]. They approach the approximate solution of the regularized Motzkin–Straus (unweighted) program by splitting its objective function into two convex terms, for which they obtain a set of global optimality conditions. At each iteration their method improves upon a KKT point which is sought by some conventional procedure.

Before commenting on the results presented in Tables 1 and 2, we note that LDR performed very poorly in all but the most trivial instances, although it converged very quickly. In fact, it is even worse than plain replicator dynamics, which are essentially gradient-based procedures (see [6]).

The *c-fat*, *Hamming*, and *Johnson* graph categories are certainly those that have proven most vulnerable to the different approaches. All methods, in fact, managed to systematically attain a maximum clique, except for one Hamming graph. *Hamming* and *Johnson* graphs are borrowed from coding theory, whereas the *c-fat* ones are used in fault diagnosis. The notation "-" in columns AR, CBH, and RD indicates data not presented in the original papers, from which the values here were taken.

The *p_hat* graphs are generalized random graphs with a wider node degree spread. The generation procedure is described in [30]. In 10 out of 15 cases PBH produced the best results, and 6 of them were maximum cliques. The largest known clique was actually reached in 9 cases.

Graphs prefixed with *MANN* are a reduction to the MCP of the minimum set covering problem. For the two smallest problems, RD and our method performed equally well, obtaining a maximum clique and a largest maximal one. For the third, bigger problem, a clique very close to the maximum one (342 vs. 345) was obtained

[2] Data can be found at http://dimacs.rutgers.edu.

TABLE 1

Performance of LDR, PBH, and other competing heuristics on unweighted DIMACS graphs (part I*). Entries that correspond to the best result for a given graph are boldfaced.*

Graph	Ord.	Dens.	ω	AR	CBH	RD	LDR	Time	PBH	Time
c-fat200-1	200	7.7%	12	-	**12**	**12**	**12**	0.07	**12**	5.0
c-fat200-2	200	16.3%	24	-	**24**	**24**	**24**	0.12	**24**	9.0
c-fat200-5	200	42.6%	58	-	**58**	**58**	**58**	0.28	**58**	22.5
c-fat500-1	500	3.6%	14	-	**14**	**14**	**14**	0.48	**14**	100.3
c-fat500-2	500	7.3%	26	-	**26**	**26**	**26**	0.79	**26**	185.2
c-fat500-5	500	18.6%	64	-	**64**	**64**	**64**	1.83	**64**	464.5
c-fat500-10	500	37.4%	126	-	**126**	**126**	**126**	3.59	**126**	1024.2
hamming6-2	64	90.5%	32	-	**32**	**32**	**32**	0.01	**32**	0.4
hamming6-4	64	34.9%	4	-	**4**	**4**	**4**	0.00	**4**	0.1
hamming8-2	256	96.9%	128	-	**128**	**128**	**128**	0.98	**128**	252.6
hamming8-4	256	63.9%	16	-	**16**	**16**	**16**	0.14	**16**	22.8
hamming10-2	1024	99.0%	512	-	**512**	-	**512**	61.01	**512**	-
hamming10-4	1024	82.9%	≥ 40	-	**35**	-	32	4.1	32	-
johnson8-2-4	28	55.6%	4	-	**4**	**4**	**4**	0.00	**4**	0.0
johnson8-4-4	70	76.8%	14	-	**14**	**14**	**14**	0.01	**14**	0.3
johnson16-2-4	120	76.5%	8	-	**8**	**8**	**8**	0.01	**8**	1.1
johnson32-2-4	496	87.9%	≥ 16	-	**16**	**16**	**16**	0.54	**16**	184.8
p_hat300-1	300	24.4%	8	**8**	**8**	**8**	6	0.10	**8**	14.0
p_hat300-2	300	48.9%	25	**25**	**25**	**25**	16	0.20	**25**	34.9
p_hat300-3	300	74.5%	36	35	**36**	34	21	0.25	35	61.0
p_hat500-1	500	25.3%	9	**9**	**9**	**9**	6	0.27	**9**	83.5
p_hat500-2	500	50.5%	36	**36**	35	35	26	0.82	**36**	282.5
p_hat500-3	500	75.2%	≥ 50	47	**49**	**49**	30	0.94	48	485.7
p_hat700-1	700	24.9%	11	9	**11**	**11**	5	0.47	10	249.4
p_hat700-2	700	49.8%	44	41	**44**	**44**	20	1.26	**44**	1022.3
p_hat700-3	700	74.8%	≥ 62	59	60	**62**	29	1.76	**62**	1804.0
p_hat1000-1	1000	24.5%	≥ 10	**10**	**10**	-	7	1.17	**10**	798.0
p_hat1000-2	1000	49.0%	≥ 46	44	**46**	-	18	2.37	**46**	-
p_hat1000-3	1000	74.4%	≥ 66	62	**65**	-	31	3.82	64	-
p_hat1500-1	1500	25.3%	12	10	11	-	9	3.12	**12**	-
p_hat1500-2	1500	50.6%	≥ 65	**64**	63	-	28	7.69	**64**	-
p_hat1500-3	1500	75.4%	≥ 94	91	**94**	-	43	11.43	91	-

by PBH. Note that here LDR performs remarkably well.

The test graphs prefixed with *keller* arise in conjunction with Keller's conjecture on tilings using hypercubes [10]. Here we could run PBH on only the two smallest instances due to memory restrictions. RD and PBH computed a maximum clique, and the latter also obtained the largest clique for the second instance.

Brockington and Culberson [9] developed their method that produced the graphs prefixed with *brock*. Their method uses a form of degree equalization to hide a large clique in a multitude of smaller ones. Also for this category PBH reached the largest cliques, except for one instance in which CBH found the maximum one. All other computed cliques are not maximum and the size gap between them and the maximum ones grows with the order of the graphs. The latter fact shows the effectiveness of Brockington and Culberson's approach for producing hard problems for algorithms based on the Motzkin–Straus continuous formulation.

The generation procedure for the Sanchis graphs (*san*) is described in [18, 29]. In 12 out of 15 cases, PBH produced the best results, and 11 of them were maximum cliques. In only one case did we obtain a clique smaller than that of AR, and in two cases RD performed slightly better. It is interesting to notice that AR and CBH obtained cliques that are, on average, half the size of those returned by PBH and RD.

TABLE 2

Performance of LDR, PBH, and other competing heuristics on unweighted DIMACS graphs (part II*). Entries that correspond to the best result for a given graph are boldfaced.*

Graph	Ord.	Dens.	ω	AR	CBH	RD	LDR	Time	PBH	Time
MANN_a9	45	92.7%	16	**16**	**16**	**16**	**16**	0.00	**16**	0.1
MANN_a27	378	99.0%	126	117	121	**125**	**125**	2.18	**125**	699.7
MANN_a45	1035	99.6%	345	-	336	-	340	43.72	**342**	-
keller4	171	64.9%	11	8	10	**11**	7	0.03	**11**	3.6
keller5	776	75.2%	27	16	21	25	15	1.27	**26**	1093.5
keller6	3361	81.8%	≥ 59	-	-	-	**31**	45.54	-	-
brock200_1	200	74.5%	21	19	**20**	**20**	13	0.07	**20**	9.7
brock200_2	200	49.6%	12	10	**12**	11	7	0.04	11	5.1
brock200_3	200	60.5%	15	13	**14**	**14**	10	0.6	**14**	6.4
brock200_4	200	65.8%	17	14	**16**	15	11	0.06	**16**	7.3
brock400_1	400	74.8%	27	20	23	**24**	17	0.37	**24**	111.6
brock400_2	400	74.9%	29	23	**24**	**24**	17	0.37	**24**	113.3
brock400_3	400	74.8%	31	23	23	**24**	17	0.37	**24**	111.2
brock400_4	400	74.9%	33	23	**24**	**24**	16	0.35	**24**	112.7
brock800_1	800	64.9%	23	18	20	**21**	13	1.18	**21**	858.6
brock800_2	800	65.1%	24	18	19	**20**	13	1.19	**20**	866.4
brock800_3	800	64.9%	25	19	**20**	**20**	15	1.34	**20**	864.5
brock800_4	800	65.0%	26	19	19	**20**	16	1.40	**20**	862.4
san200_0.7_1	200	70.0%	30	15	15	**30**	16	0.09	**30**	9.9
san200_0.7_2	200	70.0%	18	12	12	**18**	12	0.08	17	8.2
san200_0.9_1	200	90.0%	70	45	46	**70**	38	0.19	**70**	28.8
san200_0.9_2	200	90.0%	60	39	36	**60**	30	0.16	**60**	22.8
san200_0.9_3	200	90.0%	44	31	30	**44**	25	0.13	**44**	19.0
san400_0.5_1	400	50.0%	13	7	8	**13**	7	0.20	**13**	52.3
san400_0.7_1	400	70.0%	40	20	20	**40**	20	0.43	**40**	142.0
san400_0.7_2	400	70.0%	30	15	15	**30**	15	0.35	**30**	110.7
san400_0.7_3	400	70.0%	22	12	14	**19**	14	0.31	17	93.8
san400_0.9_1	400	90.0%	100	50	50	**100**	45	0.88	**100**	397.8
sanr200_0.7	200	69.7%	18	16	**18**	**18**	12	0.07	**18**	8.2
sanr200_0.9	200	89.8%	42	**41**	**41**	**41**	32	0.16	**41**	21.4
sanr400_0.5	400	50.1%	13	**13**	12	12	10	0.25	**13**	059.5
sanr400_0.7	400	70.0%	≥ 21	**21**	20	20	16	0.36	20	101.9
san1000	1000	50.2%	15	8	8	-	8	1.34	**15**	1185.0

Overall, these results show the clear superiority of PBH over both AR and CBH. It also turns out that PBH and RD perform equally well. However, the authors report in [20] that for graphs of order up to 500, the computational time of RD on a PC Pentium 166 MMX varied from 30 to 40 minutes on average, with a maximum of 1 h. 43 min. On larger graphs (i.e., up 800 vertices), the algorithm took from 17 min. to 8 h. 22 min. to converge. These high computational times prevented them from applying RD on graphs with more than 800 nodes.

Comparing complexity and computational times, however, is very difficult for this kind of heuristics. In fact we completely lack a clear complexity assessment of CBH, AR, and RD, and the computing times provided with each method refer to architectures and implementation solutions too different to be worth analyzing.

A remarkable empirical finding was that Algorithms 3.1 and 3.2 never failed to return a clique; hence, by Theorem 2.2, they always returned a maximal clique. Thus, we never needed to invoke any local search procedure in order to reach a nearby local minimizer. We tried to find exceptions by running them on random unweighted graphs with nonclique regular subgraphs, which do correspond to nonoptimal KKT points of (2). Hundreds of experiments were conducted on random instances with different

degrees of noise, but they never failed to return a maximal clique. At the moment we cannot give a formal proof of this fact.

Before we present the results obtained by PBH on weighted graphs, it is worth discussing how it compares with other maximum clique heuristics that are *not* based on the Motzkin–Straus or related continuous formulations. Many such heuristics were presented at the second DIMACS implementation challenge on cliques, coloring, and satisfiability [19], and those based on tabu search, simulated annealing, and neural networks are among the most powerful. In the following discussion we shall neglect the easiest graph families, i.e., *c-fat*, *Hamming*, *Johnson*, and *MANN*, where straightforward greedy heuristics (and indeed Lemke's algorithm) already provide satisfactory results (see [30]).

In [30], Soriano and Gendreau presented three variants of tabu search for maximum cliques. The first two versions are deterministic algorithms. One uses a single tabu list of the last solutions visited, while the other uses an additional list (with an associated aspiration mechanism) containing the last vertices deleted. The third algorithm is probabilistic in nature and uses the same two tabu lists and aspiration mechanism as the second one. As it turns out, overall their results are comparable with those obtained with PBH. On the *p-hat* graphs, PBH obtained the same clique size as the three tabu search algorithms 8 times, it got smaller cliques in 6 cases (the difference being typically of one or two nodes), and in one case it yielded a larger clique. On the *keller* and the *brock* graphs, tabu search worked slightly better. In a few cases it obtained a larger clique, but when this happened the difference consisted of just a single vertex. Finally, on the *san* family the three tabu search heuristics did not perform equally well, the probabilistic one being the poorest. Here PBH obtained the same clique sizes as the double list variant in 11 cases, it returned a larger clique in 2 cases, and a smaller one twice. Compared to the single list heuristic, a similar picture emerges. Here PBH obtained a larger clique in three cases and a smaller one twice. It should be noticed that when PBH outperforms tabu search, the difference in clique size is significant (e.g., 30 vs. 19, 15 vs. 10, etc.), while the opposite is not true.

Homer and Peinado [16] compare three heuristics for maximum clique, namely, a straightforward greedy heuristic, a randomized version of Boppana and Halldórsson's subgraph-exclusion algorithm [8], and a version of simulated annealing with a simple cooling schedule. The algorithms were tested over very large graphs, and the overall conclusion was that simulated annealing outperforms the other competing algorithms. As far as comparison with PBH is concerned, it turns out that the average clique sizes obtained by simulated annealing in 1000 trials per graph on a selection of graphs from the *p-hat*, *keller*, and *brock* families (no results are presented on the Sanchis graphs) are always rather smaller than those obtained by PBH, which, by contrast, is run only once. There is only one exception: the *p_hat*1500-3 graph, where PBH found a clique of 91 vertices and the average clique size found by simulated annealing was 92.2. Looking at the best results obtained over the 1000 runs, it turns out that simulated annealing equaled PBH 8 times, found a slightly larger clique in another 8 cases (usually one vertex larger, except for *p_hat*1500-3), and in a single case PBH got a better result.

In [18], Jagota, Sanchis, and Ganesan developed several neural-network heuristics based on the so-called Hopfield model to approximate maximum clique. Overall, the best results were obtained using a *greedy steep descent* (GSD) dynamics, although it was slower than the others. The best results on the Sanchis graphs, in contrast, were

obtained using a stochastic steep descent heuristic endowed with a "reinforcement learning" strategy that automatically adjusts the internal parameters as the process evolves (SSD_{RL}). PBH significantly outperforms these models. Specifically, on the *brock*, *keller*, and *p-hat* graphs, PBH always found cliques of size larger than or equal to those found by GSD. On the *san* family the contrast is even more evident. Here PBH found a clique larger than SSD_{RL} 11 times and obtained the same clique size in the remaining 4 cases. In a few cases, the cliques found by PBH were substantially larger than those found by SSD_{RL} (44 vs. 33, 30 vs. 18, etc.).

Grossman [14] also proposed a neural-network heuristic based on the Hopfield model, originally designed for an all-optical implementation. The model has a threshold parameter which determines the character of the stable states of the network. The author suggests an annealing strategy on this parameter, and an adaptive procedure to choose the network's initial state and threshold. Experiments over random as well as selected DIMACS graphs are reported. (Being a randomized procedure, for each graph hundreds of trials were performed.) Compared to PBH, a picture similar to simulated annealing emerges. The average clique sizes found by Grossman's heuristic are substantially smaller than those returned by PBH on all graph families. (No results on the Sanchis family are presented in [14].) Taking the best results found, out of 17 instances PBH found a larger clique in 5 cases, a smaller one in 4 cases, and the same clique size in the remaining 8 instances. Again, we stress the fact that PBH is run only once on each graph instance and no randomization takes place.

4.2. Weighted graphs. For the weighted case there are no widely accepted benchmark graphs, and therefore we adopted weighted random graphs as a testbed for Algorithm 3.3. To obtain the weighted clique number for each test graph, we used Babel's method [1], which is one of the most efficient algorithms available in the MWCP literature. Babel uses a branch and bound approach as follows: Upper and lower bounds for the maximum weight clique are found by coloring the weighted graph, where the number of colors represents the total sum of all weights. The branching part of Babel's algorithm divides the bounded search-tree into smaller subproblems, the branching decisions depending on a specific order of all possible remaining nodes. By applying these steps recursively, the maximum weight clique will be found in finite time, and for not too big and too dense graphs in very short time. Unfortunately, the coloring heuristic employed by this method severely restricts node weights to discrete values. For example, if we consider graphs with floating point weights between 1 and 10, and with 3 significant digits, this would lead to as many as 9,000 possible discrete weights. This means that Babel's method could use up to 900,000 colors in a graph of order 100. To accommodate this deficiency we generated random graphs with random integer weights ranging between 1 and 10.

In this series of experiments we did not run the LDR algorithm, because of the poor performance obtained on unweighted graphs. Given the clique C found by Algorithm 3.3, as a success measure we took the ratio $R = W(C)/\omega(G, w)$. Table 3 lists average results (*Avg. R* columns) and their standard deviations (*St. Dev.* columns) for families of 20 random graphs with 100 vertices and various density values p.

Usual random graphs (*Normal* in Table 3) tend to be very regular (i.e., the degree of all nodes is nearly the same). This feature is typically not shared by real-world instances; hence we used Algorithm 4.1, borrowed from [7], to generate more irregular instances (*Irregular*). The same intent drove the choice of performing tests over families of DIMACS *p-hat* graphs (*p-hat* columns).

On all types of graphs we obtained very positive figures. In particular, for normal

ALGORITHM 4.1. AN EDGE GENERATION PROCEDURE FOR RANDOM IRREGULAR GRAPHS.

$\mu = p\binom{n}{2}$.
while $(\mu > 0)$
    choose randomly $v \in \{1, \ldots, n\}$ and $d \in \{1, \ldots, n-1\}$;
    add d edges to randomly chosen neighbors of v;
    if this is not possible
      add the maximum of free edges to neighbors of v;
    $\mu = \mu -$ number of actually added edges.
endwhile;

TABLE 3
Performance of Algorithm 3.3 *on weighted random graphs with* 100 *vertices (see text for explanation).*

p	Normal		Irregular		p-hat	
	Avg. R	St. Dev.	Avg. R	St. Dev.	Avg. R	St. Dev.
0.10	97.95%	±0.15	98.44%	±0.13	99.33%	±0.09
0.20	97.73%	±0.16	98.63%	±0.12	97.17%	±0.17
0.30	97.25%	±0.17	98.84%	±0.11	96.38%	±0.20
0.40	95.04%	±0.23	98.53%	±0.12	97.54%	±0.16
0.50	94.61%	±0.24	98.74%	±0.12	94.56%	±0.24
0.60	94.71%	±0.23	99.64%	±0.06	96.20%	±0.20
0.70	96.10%	±0.20	98.94%	±0.11	94.44%	±0.24
0.80	93.13%	±0.26	98.56%	±0.12	94.64%	±0.23
0.90	94.29%	±0.24	99.56%	±0.07	95.26%	±0.22
0.95	96.49%	±0.19	99.75%	±0.05	94.49%	±0.24

random graphs one can see how efficiency slowly decreases with increasing density but always remains above 93%. For irregular graphs these figures improve considerably, never falling below 98.4% efficiency. The same can be said for the *p-hat* graphs. But in this last case it must be taken into account that for p close to 0.5, the node degree variance is largest. The table reflects this fact in that performance is optimal for sparse graphs, is worst for p close to 0.5, and then slowly improves while moving toward $p = 1$. At this end-point the increased density becomes the dominant reason for not reaching the heaviest clique.

The above experiments were conducted on a machine equipped with a 400MHz Alpha CPU. On this machine, computing times for PBH ranged (approximately) between 0.6 and 9 seconds.

5. Conclusions. We have presented an effective heuristic for the MWCP which employs a pivoting algorithm on an LCP problem formulation derived from a development of the Motzkin–Straus theorem. The remarkable effectiveness of our approach and the empirical immunity of Lemke's method to saddle points seems to indicate that pivoting-based methods offer a promising new way to tackle this and related combinatorial problems. The algorithm has already been applied with success to graph matching problems arising in computer vision and pattern recognition [21]. Note also that our algorithm is completely devoid of working parameters, a valuable feature which distinguishes it from other heuristics proposed in the literature (see, e.g., [4]). In future investigations we will try to give a formal proof of convergence to local minimizers, and we will tackle the problem of reducing the time and space complexity of our method.

Appendix. Optimality and strict complementarity. Here we provide a discussion of second-order optimality conditions for (2), where $f(x) = x^T Q x$ with a general symmetric $n \times n$ matrix rather than Q_G, in relation to the strict complementarity condition. First we rephrase optimality in terms of copositivity with respect to a polyhedral cone Γ. Recall that, given a cone $\Gamma \subseteq \mathbb{R}^n$, a symmetric matrix Q is said to be Γ-*copositive* if $x^T Q x \geq 0$ for all $x \in \Gamma$. If the inequality holds strictly for all $x \in \Gamma \setminus \{o\}$, then Q is said to be *strictly* Γ-*copositive*. As usual, define $e^\perp = \{v \in \mathbb{R}^n : e^T v = 0\}$.

THEOREM A.1. *Let $x \in \Delta$ and $\gamma = x^T Q x$. If x is a KKT point of* (2)*, then set*

$$\Gamma^\star(x) = \{v \in e^\perp : v_i \geq 0 \text{ if } i \in V \setminus S(x) \text{ and } v^T Q x = 0\}\,. \tag{11}$$

Then

(a) *x is a local solution to* (2) *if and only if Q is $\Gamma^\star(x)$-copositive;*

(b) *x is a strict local solution to* (2) *if and only if Q is strictly $\Gamma^\star(x)$-copositive.*

Proof. If $\Gamma(x) = \{v \in \mathbb{R}^n : v_i \geq 0 \text{ if } i \in V \setminus S(x)\}$ denotes the tangent cone of Δ at x, then $\Gamma^\star(x)$ as defined in (11) satisfies $\Gamma^\star(x) = \{v \in \Gamma(x) : v^T \nabla f(x) = 0\}$, i.e., coincides with the reduced tangent cone. Hence (a) is established by Theorem 2 of [2], while (b) can be proved by a simpler variant of the argument therein. □

Observe that, in light of the above conditions, the last statement of Theorem 2.1 can be rephrased as follows: If Q_G is $\Gamma^\star(x)$-copositive, then Q_G is even strictly so. (This holds also for every other matrix $Q \in \mathcal{C}(G, w)$, the entire class introduced in [3].)

For quadratic problems over polyhedra more general than Δ, there are similar second-order optimality conditions, also for global optimality; see, e.g., [2]. All conditions involve checking copositivity, which from a practical point of view should be avoided, as checking copositivity is NP-hard [25] whereas checking definiteness (see below) can be done in polynomial time. In contrast with several other problems (e.g., the simplex method in linear optimization), this difference in worst-case complexity is also reflected in the actual average case behavior. Thus an additional aspect of the significance of strict complementarity becomes evident.

THEOREM A.2. *If $x \in \Delta$ is a KKT point of* (2) *which satisfies the strict complementarity condition, then the reduced tangent cone*

$$\Gamma^\star(x) = \{v \in e^\perp : v_i = 0 \text{ if } i \in V \setminus S(x)\} \tag{12}$$

becomes a linear subspace.

Further, if x is a vertex of Δ, then x is a strict local solution to (2)*.*

Otherwise, assume that x has $r + 1 \geq 2$ strictly positive coordinates, pick a fixed $i \in S(x)$, and form the symmetric $r \times r$ matrix

$$\bar{Q} = [q_{ii} + q_{jk} - q_{ij} - q_{ik}]_{(j,k) \in S(x) \setminus \{i\} \times S(x) \setminus \{i\}}\,. \tag{13}$$

Then

(a) *x is a strict local solution to* (2) *if and only if $\bar{Q}$ is positive-definite;*

(b) *x is a local solution to* (2) *if and only if $\bar{Q}$ is positive-semidefinite.*

Proof. To show (12), we employ the KKT conditions (4). Then $\lambda_i > 0$ for all $i \in V \setminus S(x)$ implies via (11) and

$$0 = v^T \nabla f(x) = \sum_{i \in V \setminus S(x)} \lambda_i v_i - \lambda e^T v = \sum_{i \in V \setminus S(x)} \lambda_i v_i$$

that $v_i = 0$ for all $i \in V \setminus S(x)$ if $v \in \Gamma^\star(x)$. The converse is also obvious. Next, if x is a vertex of the feasible set, then $\Gamma^\star(x) = \{o\}$, so that the copositivity condition in Theorem A.1(a) is void. (Note that local optimality of vertices which are strictly complementary KKT points holds in a much more general context.) Now assume that x is no vertex. Denoting again by e_i the ith column of the $n \times n$ identity matrix I_n, we obtain a basis for $\Gamma^\star(x)$ by $\{e_i - e_j : j \in S(x) \setminus \{i\}\}$ and collect these vectors as columns of an $n \times r$ matrix U so that $\Gamma^\star(x) = U(\mathbb{R}^r)$. But U can be written, after suitable reordering, as $U = [e, -I_r, O]^T$. Now partition Q into appropriate blocks to arrive at $\bar{Q} = U^T Q U$ as in (13). As a consequence, Q is $\Gamma^\star(x)$-copositive if and only if $\bar{Q}$ is positive-semidefinite, and similarly for the strict versions. □

A consequence of the last statement in Theorem 2.1 is that, under strict complementarity, the (positive-semidefinite) matrices $\bar{Q}$ are nonsingular if $Q = Q_G$. (Again, this holds for every $Q \in \mathcal{C}(G, w)$, the entire class introduced in [3].) Thus Theorem A.2 can be viewed as a sort of converse of Theorem 2.4, where nonsingularity of certain matrices in turn guarantees strict complementarity.

Acknowledgments. The authors wish to thank Richard Cottle for the valuable assistance he was willing to provide during early and late stages of this work, and the anonymous referees for their helpful comments. M.P. would also like to thank Steven Zucker for stimulating discussions which triggered this research.

REFERENCES

[1] L. Babel, *A fast algorithm for the maximum weight clique problem*, Computing, 52 (1994), pp. 31–38.

[2] I. M. Bomze, *Copositivity conditions for global optimality in indefinite quadratic programming problems*, Czech. J. Oper. Res., 1 (1992), pp. 7–19.

[3] I. M. Bomze, *On standard quadratic optimization problems*, J. Global Optim., 13 (1998), pp. 369–387.

[4] I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo, *The maximum clique problem*, in Handbook of Combinatorial Optimization—Suppl. Vol. A, D.-Z. Du and P. M. Pardalos, eds., Kluwer Academic Publishers, Boston, MA, 1999, pp. 1–74.

[5] I. M. Bomze, M. Budinich, M. Pelillo, and C. Rossi, *Annealed replication: A new heuristic for the maximum clique problem*, Discrete Appl. Math., 2002, to appear.

[6] I. M. Bomze, M. Pelillo, and R. Giacomini, *Evolutionary approach to the maximum clique problem: Empirical evidence on a larger scale*, in Developments in Global Optimization, I. M. Bomze, T. Csendes, R. Horst, and P. M. Pardalos, eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997, pp. 95–108.

[7] I. M. Bomze, M. Pelillo, and V. Stix, *Approximating the maximum weight clique using replicator dynamics*, IEEE Trans. Neural Networks, 11 (2000), pp. 1228–1241.

[8] R. Boppana and M. Halldórsson, *Approximating maximum independent sets by excluding subgraphs*, BIT, 32 (1992), pp. 180–196.

[9] M. Brockington and J. C. Culberson, *Camouflaging independent sets in quasi-random graphs*, in Cliques, Coloring and Satisfiability, D. Johnson and M. A. Trick, eds., DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 26, AMS, Providence, RI, 1996, pp. 75–88.

[10] K. Corrádi and S. Szabó, *A combinatorial approach for Keller's conjecture*, Period. Math. Hungar., 21 (1990), pp. 95–100.

[11] R. W. Cottle, J. Pang, and R. E. Stone, *The Linear Complementarity Problem*, Academic Press, Boston, MA, 1992.

[12] L. E. Gibbons, D. W. Hearn, and P. M. Pardalos, *A continuous-based heuristic for the maximum clique problem*, in Cliques, Coloring and Satisfiability, D. Johnson and M. A. Trick, eds., DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 26, AMS, Providence, RI, 1996, pp. 103–124.

[13] L. E. Gibbons, D. W. Hearn, P. M. Pardalos, and M. V. Ramana, *Continuous characterizations of the maximum clique problem*, Math. Oper. Res., 22 (1997), pp. 754–768.

[14] T. Grossman, *Applying the INN model to the maximum clique problem*, in Cliques, Coloring and Satisfiability, D. Johnson and M. A. Trick, eds., DIMACS Ser. Discrete Math. Theoret.

Comput. Sci. 26, AMS, Providence, RI, 1996, pp. 125–145.
[15] J. Hofbauer and K. Sigmund, *Evolutionary Games and Population Dynamics*, Cambridge University Press, Cambridge, UK, 1998.
[16] S. Homer and M. Peinado, *Experiments with polynomial-time CLIQUE approximation algorithms on very large graphs*, in Cliques, Coloring and Satisfiability, D. Johnson and M. A. Trick, eds., DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 26, AMS, Providence, RI, 1996, pp. 147–167.
[17] R. Horaud and T. Skordas, *Stereo correspondence through feature grouping and maximal cliques*, IEEE Trans. Pattern Anal. Machine Intell., 11 (1989), pp. 1168–1180.
[18] A. Jagota, L. Sanchis, and R. Ganesan, *Approximately solving maximum clique using neural network and related heuristics*, in Cliques, Coloring and Satisfiability, D. Johnson and M. A. Trick, eds., DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 26, AMS, Providence, RI, 1996, pp. 169–204.
[19] D. Johnson and M. A. Trick, eds., *Cliques, Coloring and Satisfiability: Second DIMACS Implementation Challenge*, DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 26, AMS, Providence, RI, 1996.
[20] A. Kuznetsova and A. Strekalovsky, *On solving the maximum clique problem*, J. Global Optim., 21 (2001), pp. 265–288.
[21] A. Massaro and M. Pelillo, *A complementary pivoting approach to graph matching*, in Energy Minimization Methods in Computer Vision and Pattern Recognition, M. Figueiredo, J. Zerubia, and A. K. Jain, eds., Lecture Notes in Comput. Sci. 2134, Springer-Verlag, Berlin, 2001, pp. 469–479.
[22] A. Menon, K. Mehrotra, C. K. Mohan, and S. Ranka, *Optimization using replicators*, in Proceedings of the 6th International Conference on Genetic Algorithms, Pittsburg, PA, 1995, Morgan Kaufmann, San Francisco, CA, pp. 209–216.
[23] D. A. Miller and S. W. Zucker, *Efficient simplex-like methods for equilibria of nonsymmetric analog networks*, Neural Computation, 4 (1992), pp. 167–190.
[24] T. S. Motzkin and E. G. Straus, *Maxima for graphs and a new proof of a theorem of Turán*, Canad. J. Math., 17 (1965), pp. 533–540.
[25] K. G. Murty and S. N. Kabadi, *Some NP-complete problems in quadratic and linear programming*, Math. Programming, 39 (1987), pp. 117–129.
[26] M. Pelillo, *Replicator dynamics in combinatorial optimization*, in Encyclopedia of Optimization, C. A. Floudas and P. M. Pardalos, eds., Kluwer Academic Publishers, Boston, MA, 2001.
[27] M. Pelillo and A. Jagota, *Feasible and infeasible maxima in a quadratic program for the maximum clique problem*, J. Artificial Neural Networks, 2 (1995), pp. 411–420.
[28] M. Pelillo, K. Siddiqi, and S. W. Zucker, *Matching hierarchical structures using association graphs*, IEEE Trans. Pattern Anal. Machine Intell., 21 (1999), pp. 1105–1120.
[29] L. Sanchis, *Test case construction for the vertex cover problem*, in Computational Support for Discrete Mathematics, N. Dean and G. E. Shannon, eds., DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 15, AMS, Providence, RI, 1994, pp. 315–326.
[30] P. Soriano and M. Gendreau, *Tabu search algorithms for the maximum clique problem*, in Cliques, Coloring and Satisfiability, D. Johnson and M. A. Trick, eds., DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 26, AMS, Providence, RI, 1996, pp. 221–242.

SIAM J. OPTIM.
Vol. 12, No. 4, pp. 949–978

A SUPERLINEARLY CONVERGENT SEQUENTIAL QUADRATICALLY CONSTRAINED QUADRATIC PROGRAMMING ALGORITHM FOR DEGENERATE NONLINEAR PROGRAMMING*

MIHAI ANITESCU†

Abstract. We present an algorithm that achieves superlinear convergence for nonlinear programs satisfying the Mangasarian–Fromovitz constraint qualification and the quadratic growth condition. This convergence result is obtained despite the potential lack of a locally convex augmented Lagrangian. The algorithm solves a succession of subproblems that have quadratic objectives and quadratic constraints, both possibly nonconvex. By the use of a trust-region constraint we guarantee that any stationary point of the subproblem induces superlinear convergence, which avoids the problem of computing a global minimum. We compare this algorithm with sequential quadratic programming algorithms on several degenerate nonlinear programs.

Key words. sequential quadratic programming, degenerate constraints, quadratic constraints, superlinear convergence

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

PII. S1052623499365309

1. Introduction.

Recently, there has been renewed interest in analyzing and modifying algorithms for constrained nonlinear optimization for cases where the traditional regularity conditions do not hold [5, 17, 16, 26, 30, 29]. This research has been motivated by the fact that large-scale nonlinear programming problems tend to be almost degenerate (have large condition numbers for the Jacobian of the active constraints). It is therefore important to define algorithms that are as little dependent as possible on the ill-conditioning of the constraints. In this work, we term as degenerate those nonlinear programs (NLPs) for which the gradients of the active constraints are linearly dependent. In this case there may be several feasible Lagrange multipliers.

Many of the previous analyses and rate-of-convergence results for degenerate NLPs [5, 17, 16, 26, 30, 29] are based on the validity of some second-order conditions. These are essentially equivalent to the condition in unconstrained optimization that, for a critical point of a function $f(x)$ to be a local minimum, $f_{xx} \succeq 0$ is a necessary condition and $f_{xx} \succ 0$ is a sufficient condition. Here $\succeq$ is the positive semidefinite ordering. For these conditions the place of f_{xx} in constrained optimization is taken by L_{xx}, the Hessian of the Lagrangian, which is now required to be positive definite on the critical cone for one or all of the Lagrange multipliers [10, 27].

This work differs from previous approaches in that we assume only that

1. at a local solution x^* of the constrained NLP, the first-order Mangasarian–Fromovitz [23, 22] constraint qualification holds;

*Received by the editors December 22, 1999; accepted for publication (in revised form) November 26, 2001; published electronically April 19, 2002. This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing, U.S. Department of Energy, under contract W-31-109-Eng-38. This work was also supported by award DMS-9973071 of the National Science Foundation.

http://www.siam.org/journals/siopt/12-4/36530.html

†Thackeray 301, Department of Mathematics, University of Pittsburgh, Pittsburgh, PA 15213 (anitescu@math.pitt.edu). Part of this work was completed while the author was the Wilkinson Fellow at the Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL.

2. the quadratic growth condition (QG) (see [6, 20])

$$f(x) \geq f(x^*) + \sigma \, ||x - x^*||^2 \tag{1.1}$$

   is satisfied for some $\sigma > 0$ and all x feasible in a neighborhood of x^*;
3. the data of the problem are twice continuously differentiable.

These assumptions are equivalent to a weaker form of the second-order sufficient conditions [19, 6], which do not require the positive semidefiniteness of the Hessian of the Lagrangian on the entire critical cone. In a recent paper [2] it was shown that these conditions guarantee that x^* is an isolated stationary point and that a steepest-descent-like algorithm induces linear convergence to x^*. The framework used here accommodates even problems for which no locally convex augmented Lagrangian exists [2], and which do not satisfy the assumptions of most other convergence results [5, 17, 16, 26, 30].

In this paper we define an algorithm that is superlinearly convergent even in the very general conditions outlined above. The trade-off is that the subproblems to be solved are more complex than a quadratic program. The algorithm can be justified by a particular perspective on Newton's method for unconstrained optimization. If $f(x)$ is the function to be minimized without constraints, then, sufficiently close to a solution x^*, Newton's direction d is a solution of the quadratic minimization problem

$$\min_{d \in \mathbb{R}^n} f(x) + \nabla_x f(x)^T d + \tfrac{1}{2} d^T \nabla^2_{xx} f(x) d.$$

If we have an inequality constrained NLP,

$$\begin{aligned} \min_x \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0, \qquad i = 1, 2, \ldots, m, \end{aligned}$$

its second-order approximation at x is the following problem:

$$\begin{aligned} \min_{d \in \mathbb{R}^n} \quad & f(x) + \nabla_x f(x)^T d + \tfrac{1}{2} d^T \nabla^2_{xx} f(x) d \\ \text{subject to} \quad & g_i(x) + \nabla_x g_i(x)^T d + \tfrac{1}{2} d^T \nabla^2_{xx} g_i(x) d \leq 0, \quad i = 1, 2, \ldots, m. \end{aligned}$$

We call such a problem a quadratically constrained quadratic program (QCQP). To ensure that the problem is bounded even for x far from the solution x^*, we add to the problem a trust-region constraint, which is also quadratic:

$$d^T d \leq \gamma^2.$$

The problem is generally not convex, and thus finding the global optimum may be a difficult problem. Also, the trust-region constraint may interfere with the order of convergence. However, we show that for x close to x^* and for γ sufficiently small but fixed,

1. the trust-region constraint is inactive at any stationary point of the QCQP;
2. any stationary point d of the QCQP used as a progress direction induces superlinear convergence.

Therefore, finding a local solution to the QCQP is sufficient to induce superlinear convergence of the iterates, which considerably reduces the conceptual complexity of a sequential QCQP (SQCQP) algorithm. Note that the QCQP subproblem is identical to the one used in [21], although the analysis conditions in this work are more general.

The paper is structured as follows. In subsection 1.1 we discuss the different conditions defining a stationary point of an NLP and the QG. Section 2 characterizes stationary points of the second-order approximation (QCQP) of the NLP at x^*. We show that if the trust-region constraint defines a sufficiently small region, then the Mangasarian–Fromovitz constraint qualification is satisfied at any feasible point, and $d = 0$ is the unique stationary point of the QCQP. As a result, in section 3 we prove that, for x sufficiently close to x^*, the trust-region constraint is inactive at any stationary point of QCQP, and we prove the superlinear convergence of the SQCQP algorithm. In subsection 3.1 we show that the subproblems, which include a trust-region constraint, solved by sequential quadratic programming (SQP) algorithms applied to degenerate NLPs do not necessarily have an inactive trust region at a solution. In section 4 we compare the SQCQP algorithm and two SQP algorithms on three degenerate NLPs.

1.1. Previous work, framework, and notations. We deal with the NLP problem

$$\min_x f(x) \quad \text{subject to } g(x) \le 0, \tag{1.2}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}^m$ are twice continuously differentiable.

We call x a stationary point if the Fritz John conditions hold: There exist $\lambda \in \mathbb{R}^m$, $\lambda_0 \in \mathbb{R}$ with $(\lambda, \lambda_0) \neq 0$ such that

$$\nabla_x \mathcal{L}(x, \lambda_0, \lambda) = 0, \quad \lambda_0 \ge 0, \quad \lambda \ge 0, \quad g(x) \le 0, \quad \lambda^T g(x) = 0. \tag{1.3}$$

Here $\mathcal{L}$ is the Lagrangian function

$$\mathcal{L}(x, \lambda_0, \lambda) = \lambda_0 f(x) + \lambda^T g(x). \tag{1.4}$$

A local solution x^* of (1.2) is a stationary point [25]. If certain regularity conditions hold at x^* (discussed below), then there exists $\lambda \ge 0$ such that x^* with λ and $\lambda_0 = 1$ satisfy (1.3). In that case (1.3) are referred to as the KKT (Karush–Kuhn–Tucker) conditions [3, 4, 11], and λ are referred to as the Lagrange multipliers. For that case, which is the one that most often appears in this work, we define the Lagrangian as

$$\mathcal{L}(x, \lambda) = f(x) + \lambda^T g(x), \tag{1.5}$$

and the KKT conditions become

$$\nabla_x \mathcal{L}(x, \lambda) = 0, \quad \lambda \ge 0, \quad g(x) \le 0, \quad \lambda^T g(x) = 0. \tag{1.6}$$

Since our analysis is limited to a neighborhood of a point x^* that is a strict local minimum, we assume that all constraints are active at x^*, or $g(x^*) = 0$. Such a situation can be obtained by choosing a sufficiently small trust region and simply dropping the constraints i for which $g_i(x^*) < 0$, since this relationship holds in an entire neighborhood of x^*. This does not reduce the generality of our results, but it simplifies the notation because now we do not have to refer separately to the active set.

The regularity condition, or constraint qualification, ensures that a linear approximation of the feasible set in the neighborhood of x^* captures the geometry of the feasible set. Often in local convergence analysis of constrained optimization algorithms, it is assumed that the constraint gradients $\nabla_x g_i(x^*)$, $i = 1, 2, \ldots, m$, are

linearly independent, so that the Lagrange multiplier in (1.6) is unique. We assume instead the Mangasarian–Fromovitz constraint qualification (MFCQ) [23, 22]:

$$\nabla_x g_i(x^*)^T p \leq -\zeta_0, \quad i = 1, 2, \ldots, m, \quad \text{for some } \zeta_0 > 0,\ p \in \mathbb{R}^n,\ ||p|| = 1. \tag{1.7}$$

It is well known [13] that MFCQ is equivalent to the boundedness and nonemptiness of the set $\mathcal{M}(x^*)$ of Lagrange multipliers that satisfy (1.6), that is,

$$\mathcal{M}(x^*) = \{\lambda \geq 0 \mid (x^*, \lambda) \text{ satisfy (1.6)}\}. \tag{1.8}$$

Note that $\mathcal{M}(x^*)$ is certainly polyhedral in any case. Another condition equivalent to MFCQ (1.7) is (see [14])

$$0 \neq \sum_{i=1}^{m} \lambda_i \nabla_x g_i(x^*) \quad \forall \lambda_i \geq 0, \quad i = 1, 2, \ldots, m, \quad \text{such that} \quad \sum_{i=1}^{m} \lambda_i > 0. \tag{1.9}$$

The critical cone at x^* is (see [10, 28])

$$\mathcal{C} = \left\{u \in \mathbb{R}^n \mid \nabla_x g_i(x^*)^T u \leq 0,\ i = 1, 2, \ldots, m;\ \nabla_x f(x^*)^T u = 0\right\}. \tag{1.10}$$

We briefly review some of the second-order conditions in the literature. In the framework of [10], the second-order sufficient conditions for x^* to be an isolated local solution of (1.2) are (see [10, 11])

$$\exists \lambda^* \in \mathcal{M}(x^*),\ \exists \sigma > 0 \text{ such that } v^T \mathcal{L}_{xx}(x^*, \lambda^*) v \geq \sigma \|v\|_2^2 \quad \forall v \in \mathcal{C}. \tag{1.11}$$

If these conditions hold at x^* for some λ^*, then the QG is satisfied, irrespective of the validity of the first-order constraint qualification [10, 11]. An important consequence of condition (1.11) is that x^* is a local minimum of the augmented Lagrangian

$$\mathcal{L}_c(x, \lambda^*) = \mathcal{L}(x, \lambda^*) + c\,||g(x)||^2$$

for a sufficiently large constant c.

A refinement of the second-order conditions was introduced in [19]. In the presence of MFCQ, those conditions require that

$$\forall u \in \mathcal{C},\ \exists \lambda^* \in \mathcal{M}(x^*) \text{ such that } u^T \nabla_{xx}^2 \mathcal{L}(x^*, \lambda^*) u > 0. \tag{1.12}$$

Further analysis shows that, in the presence of MFCQ, these conditions are necessary and sufficient for the QG to hold [6, 19, 20, 28].

If condition (1.12) holds but (1.11) does not, then there may be no augmented Lagrangian with a positive semidefinite Hessian, as is shown with an example in [2]. This is an interesting feature since it invalidates the usual working assumption of Lagrange multiplier methods [4]. It also shows that the analysis in this paper is done without assuming the existence of an augmented Lagrangian that has x^* as an unconstrained minimum.

In our analysis we use the L_∞ nondifferentiable exact penalty function:

$$P(x) = \max\{0, g_1(x), \ldots, g_m(x)\}. \tag{1.13}$$

If the MFCQ (1.7) conditions hold at x^*, then the QG (1.1) and the second-order conditions (1.12) are each equivalent to the following condition (see [6]):

$$\max\{f(x) - f(x^*), P(x)\} \geq \sigma\,||x - x^*||^2 \tag{1.14}$$

for some $\sigma > 0$ and all x in a neighborhood of x^*.

For some function $h : \mathbb{R}^n \to \mathbb{R}^k$ we denote by c_{1h}, c_{2h} bounds depending on the first and second derivatives of h. The positive and negative parts of $h(x)$ are $h^+(x) = \max\{h(x), 0\}$ and, respectively, $h^-(x) = \max\{-h(x), 0\}$, both taken componentwise. With this notation, $h(x) = h^+(x) - h^-(x)$. Also, in our notation, $\nabla_x g_i(x)$, λ, and $\nabla_x g(x)\lambda$ are column vectors.

In this work we need to estimate distances to sets described by linear constraints:

$$\mathcal{P} \quad = \quad \{d \in \mathbb{R}^n \mid M_{eq}d + q_{eq} = 0,\ M_{in}d + q_{in} \le 0\}, \tag{1.15}$$

where M_{eq} and M_{in} are $n_{eq} \times n$ and, respectively, $n_{in} \times n$ matrices, and q_{eq} and q_{in} are n_{eq}- and, respectively, n_{in}-dimensional vectors. By Hoffman's lemma [18], if $\mathcal{P} \neq \emptyset$, there exists $c_{\mathcal{P}} > 0$ such that

$$\forall \tilde{d} \in \mathbb{R}^n, \quad D(\tilde{d}, \mathcal{P}) \le c_{\mathcal{P}} \max\left\{\|M_{eq}\tilde{d} + q\|_{eq}\|_\infty, \|(M_{in}\tilde{d} + q_{in})^+\|_\infty\right\}, \tag{1.16}$$

where by $D(\tilde{d}, \mathcal{P})$ we denote the distance from $\tilde{d}$ to the set $\mathcal{P}$. This result allows us to relate the distance from a point $\tilde{d}$ to a polyhedral set in terms of the infeasibility of $\tilde{d}$ in the representation (1.15).

2. Stationary points of QCQPs. In this section we investigate the stationary points of the QCQP

$$\text{TRQCQP}(\gamma) \qquad \begin{array}{lrcl} \min_{d\in\mathbb{R}^n} & a^T d + \frac{1}{2} d^T A d & & \\ \text{subject to} & b_i^T d + \frac{1}{2} d^T B_i d & \le & 0, \quad i = 1, 2, \ldots, m, \\ & d^T d & \le & \gamma^2, \end{array}$$

where $\gamma > 0$ defines a trust-region constraint; A, B_i, $i = 1, 2, \ldots, m$, are $n \times n$ symmetric matrices; and $a \in \mathbb{R}^n$, $b_i \in \mathbb{R}^n$, $i = 1, 2, \ldots, m$. We denote this program by TRQCQP(γ). Our assumptions concerning TRQCQP(γ) are the following:

1. At $d = 0$, MFCQ (1.7) holds:

$$b_i^T p \le -\zeta_0 \ \ \forall i \in 1, 2, \ldots, m \text{ and for some } \zeta_0 > 0,\ p \in \mathbb{R}^n,\ ||p|| = 1. \tag{2.1}$$

2. QG (1.1) is satisfied near $d = 0$: There exist $\gamma_1' > 0$ and $\sigma_1 > 0$ such that

$$\begin{array}{lrcll} & a^T d + \frac{1}{2} d^T A d & \ge & \sigma_1 \|d\|^2 & \\ \text{whenever} & b_i^T d + \frac{1}{2} d^T B_i d & \le & 0, & i = 1, 2, \ldots, m, \\ & d^T d & \le & \gamma_1'^2. & \end{array} \tag{2.2}$$

A local solution of TRQCQP(γ) is clearly $d = 0$.

The aim of this section is to show that under assumptions (2.2) and (2.1) there exists $\gamma_5 > 0$ such that $d = 0$ is the only stationary point of TRQCQP(γ), for any $0 \le \gamma \le \gamma_5$. As a consequence, any algorithm that reaches a stationary point of TRQCQP(γ) finds its global optimum. The results from [2] ensure that $d = 0$ is an isolated stationary point of TRQCQP(γ). However, the developments of this section are necessary to ensure that additional stationary points are not introduced by the trust-region constraint.

The proof has the following steps, each stated for sufficiently small γ.

- Lemma 2.4 proves that MFCQ (1.7) is satisfied for all stationary points $\tilde{d}$ of TRQCQP(γ). Therefore, at any stationary point there exist Lagrange multipliers that satisfy (1.6) applied to TRQCQP(γ).

- Lemma 2.3 ultimately implies that for any Lagrange multiplier λ at a stationary point $\tilde{d}$ of TRQCQP(γ) there exists a sufficiently close Lagrange multiplier λ^* at $d = 0$ whose active subset is included in the active subset of λ. This leads to the identity $(\lambda_i + \lambda_i^*)(b_i^T \tilde{d} + \frac{1}{2}\tilde{d}^T B_i \tilde{d}) = 0$, which helps bound from above the variations in the objective function of TRQCQP(γ) in the proof of Theorem 2.7.
- Lemma 2.5 proves that the multiplier of the trust-region constraint is bounded above. This in turn implies Lemma 2.6; the Lagrange multipliers of all potential stationary points are uniformly bounded.
- Theorem 2.7, the main result of this section, proves that $\tilde{d} = 0$ is the unique stationary point of TRQCQP(γ).

Subsection 2.1 contains additional results implied by Hoffman's lemma (1.16), which are used in section 3.

2.1. Sensitivity results for Lagrange multipliers. An immediate consequence of MFCQ (2.1) is that the set of Lagrange multipliers of TRQCQP(γ) at $d = 0$,

$$\mathcal{M}^* = \left\{ \lambda^* \in \mathbb{R}^m \mid a + \sum_{i=1}^m \lambda_i^* b_i = 0,\ \lambda^* \geq 0 \right\}, \tag{2.3}$$

is nonempty and bounded.

LEMMA 2.1. *There exists $c_{\mathcal{M}^*} > 0$ such that, for any $w \in \mathbb{R}^n$ and for any $\lambda \in \mathbb{R}^m$ satisfying*

$$a + \sum_{i=1}^m \lambda_i b_i = w, \quad \lambda \geq 0,$$

there exists a $\lambda^* \in \mathcal{M}^*$ such that $||\lambda - \lambda^*|| \leq c_{\mathcal{M}^*} ||w||$.

Proof. The proof follows by direct application of Hoffman's lemma (1.16), after using the fact that $||w||_\infty \leq ||w||$. ∎

LEMMA 2.2. *There exists $\eta > 0$ such that for all $w \in \mathbb{R}^n$ with $||w|| \leq \eta$ and any λ satisfying*

$$a + \sum_{i=1}^m \lambda_i b_i = w, \quad \lambda \geq 0,$$

there exists $\lambda^* \in \mathcal{M}^*$ such that $\lambda_i = 0 \Rightarrow \lambda_i^* = 0$.

Proof. Assume the contrary: For any $k \in \mathbb{N}$ there exists $w^k \in \mathbb{R}^n$ such that $\|w^k\| \leq \frac{1}{k}$ and there exists λ^k satisfying

$$a + \sum_{i=1}^m \lambda_i^k b_i = w^k, \quad \lambda^k \geq 0,$$

and an index set $\mathcal{I}^k \subset \{1, 2, \ldots, m\}$ such that $\lambda_{\mathcal{I}^k}^k = 0$ but $\lambda_{\mathcal{I}^k}^* \neq 0$ for all $\lambda^* \in \mathcal{M}^*$. From Lemma 2.1, $D(\lambda^k, \mathcal{M}^*) \leq c_{\mathcal{M}^*} \|w^k\| \leq c_{\mathcal{M}^*} \frac{1}{k} \to 0$ as $k \to \infty$. Since $\mathcal{M}^*$ is a compact set and the set of subsets of $\{1, 2, \ldots, m\}$ is finite, there exist a subsequence k_q, an $\mathcal{I}^* \subset \{1, 2, \ldots, m\}$, and a $\lambda^* \in \mathcal{M}^*$ such that $\mathcal{I}^{k_q} = \mathcal{I}^*$ for all $q \in \mathbb{N}$ and $\lambda^{k_q} \to \lambda^*$. From our assumptions, $\lambda_{\mathcal{I}^*}^* \neq 0$ for all $\lambda^* \in \mathcal{M}^*$. On the other hand, since $\lambda_{\mathcal{I}^*}^{k_q} = \lambda_{\mathcal{I}^{k_q}}^{k_q} = 0$ and $\lambda^{k_q} \to \lambda^*$, we must have $\lambda_{\mathcal{I}^*}^* = 0$, which is a contradiction. The proof is complete. ∎

LEMMA 2.3. *There exist $c_{\mathcal{M}} > 0$ and $\eta > 0$ such that for any $w \in \mathbb{R}^n$ with $||w|| \leq \eta$ and any λ satisfying*

$$a + \sum_{i=1}^{m} \lambda_i b_i = w, \quad \lambda \geq 0, \tag{2.4}$$

there exists $\lambda^* \in \mathcal{M}^*$ with $||\lambda - \lambda^*|| \leq c_{\mathcal{M}} ||w||$ and such that $\lambda_i = 0 \Rightarrow \lambda_i^* = 0$ for all $i \in \{1, 2, \ldots, m\}$.

Proof. Let η be the quantity defined by Lemma 2.2. Let $\mathcal{I} \subset \{1, 2, \ldots, m\}$ such that there exists a λ satisfying (2.4) and $\lambda_{\mathcal{I}} = 0$. Lemma 2.2 implies that there exists $\lambda^* \in \mathcal{M}^*$ such that $\lambda_{\mathcal{I}}^* = 0$. Let $\mathcal{M}^*{}_{\mathcal{I}}$ be the set of all such λ^*; that is,

$$\mathcal{M}^*{}_{\mathcal{I}} = \left\{ \nu \in \mathbb{R}^m \mid a + \sum_{i=1}^{m} \nu_i b_i = 0,\ \nu \geq 0, \quad \nu_I = 0 \right\}. \tag{2.5}$$

From Lemma 2.2, $\mathcal{M}^*{}_{\mathcal{I}}$ is not empty. From Hoffman's lemma (1.16), there exists $c_{\mathcal{M}_{\mathcal{I}}^*} > 0$ such that, for all $\mu \in \mathbb{R}^m$, we have

$$D(\mu, \mathcal{M}^*{}_{\mathcal{I}}) \leq c_{\mathcal{M}_{\mathcal{I}}^*} \max \left\{ \left|\left| a + \sum_{i=1}^{m} \mu_i b_i \right|\right|_\infty, ||\mu_{\mathcal{I}}||_\infty, \left|\left|\mu^-\right|\right|_\infty \right\}. \tag{2.6}$$

From Lemma 2.1 choose $\bar{\lambda}^* \in \mathcal{M}^*$ such that

$$\left|\left|\lambda - \bar{\lambda}^*\right|\right| \leq c_{\mathcal{M}^*} ||w||. \tag{2.7}$$

From the definition of $\mathcal{M}^*$ in (2.3) we have that

$$\left|\left| a + \sum_{i=1}^{m} \bar{\lambda}_i^* b_i \right|\right|_\infty = 0, \qquad \left|\left|(\bar{\lambda}^*)^-\right|\right|_\infty = 0.$$

Thus, from (2.6) we must have

$$D(\bar{\lambda}^*, \mathcal{M}_{\mathcal{I}}^*) \leq c_{\mathcal{M}_{\mathcal{I}}^*} \left|\left|\bar{\lambda}_{\mathcal{I}}^*\right|\right|_\infty. \tag{2.8}$$

We also have from our choice of $\bar{\lambda}^*$ in (2.7) that

$$\left|\left|\lambda_{\mathcal{I}} - \bar{\lambda}_{\mathcal{I}}^*\right|\right|_\infty \leq \left|\left|\lambda - \bar{\lambda}^*\right|\right| \leq c_{\mathcal{M}^*} ||w||.$$

Since $\lambda_{\mathcal{I}} = 0$, we thus have $\|\bar{\lambda}_{\mathcal{I}}^*\|_\infty = \|\lambda_{\mathcal{I}} - \bar{\lambda}_{\mathcal{I}}^*\|_\infty$, which, in conjunction with the preceding inequality and (2.8), implies that

$$D(\bar{\lambda}^*, \mathcal{M}^*{}_{\mathcal{I}}) \leq c_{\mathcal{M}_{\mathcal{I}}^*} c_{\mathcal{M}^*} ||w||.$$

Hence, from (2.7) and the preceding inequality, we have that

$$\begin{aligned} D(\lambda, \mathcal{M}^*{}_{\mathcal{I}}) &\leq \left|\left|\lambda - \bar{\lambda}^*\right|\right| + D(\bar{\lambda}^*, \mathcal{M}^*{}_{\mathcal{I}}) \\ &\leq c_{\mathcal{M}^*} ||w|| + c_{\mathcal{M}^*} c_{\mathcal{M}_{\mathcal{I}}^*} ||w|| = c_{\mathcal{M}^*} (1 + c_{\mathcal{M}_{\mathcal{I}}^*}) ||w||. \end{aligned}$$

The conclusion now follows, after taking

$$c_{\mathcal{M}} = \max_{\mathcal{I} \subset \{1,2,\ldots,m\},\ \exists \lambda^* \in \mathcal{M}^*,\ \lambda_{\mathcal{I}}^* = 0} c_{\mathcal{M}}^* (1 + c_{\mathcal{M}_{\mathcal{I}}^*}). \qquad \square$$

2.2. Stationary points of QCQPs. In this section we analyze the stationary points of TRQCQP(γ) for sufficiently small values of the parameter γ. We choose γ_1'' such that

$$||d|| \leq \gamma_1'' \Rightarrow (b_i + B_i d)^T p \leq -\frac{\zeta_0}{2} \quad \forall i \in 1, 2, \ldots, m, \tag{2.9}$$

where ζ_0, p are the quantities appearing in MFCQ (2.1) with $||p|| = 1$. We choose

$$\gamma_1 = \min\{\gamma_1', \gamma_1''\} > 0, \tag{2.10}$$

which guarantees that whenever $||d|| \leq \gamma_1$, both (2.9) and the QG (2.2) hold.

LEMMA 2.4. *There exists $\gamma_2 > 0$ such that TRQCQP(γ) satisfies MFCQ* (1.7) *at all its stationary points d, with γ such that $0 < \gamma \leq \gamma_2$.*

The important consequence of this lemma is that Lagrange multipliers exist at any stationary point of TRQCQP(γ). Note that the result of this lemma does not immediately follow from the fact that MFCQ (2.1) holds at $d = 0$ for TRQCQP(γ) and that MFCQ is stable under perturbations. In this lemma we also consider stationary points d at which the trust region may be active and at which we have no initial guarantee of the satisfaction of a constraint qualification.

Proof. Take the QCQP

$$\begin{array}{ll} \min_{d \in \mathbb{R}^n} & d^T d \\ \text{subject to} & b_i^T d + \frac{1}{2} d^T B_i d \leq 0, \quad i = 1, 2, \ldots, m, \end{array} \tag{2.11}$$

with global solution $d = 0$. At $d = 0$, (2.11) satisfies MFCQ (2.1) as well as the QG (1.1). From [2], $d = 0$ is an isolated stationary point of (2.11). Therefore there exists a $\gamma_2' > 0$ such that the only stationary point d of (2.11) that satisfies $d^T d \leq (\gamma_2')^2$ is $d = 0$.

Now take $\gamma_2 = \min\{\gamma_1, \gamma_2'\}$. Assume that there exists γ, $0 < \gamma \leq \gamma_2$, such that MFCQ (1.7) is not satisfied at some stationary point $\bar{d}$ of TRQCQP(γ). From (1.9) and (1.3) it follows that there exist $\bar{\lambda} \geq 0$ and $\bar{\lambda}_0 \geq 0$, not both equal to 0, such that

$$\begin{array}{rcll} \bar{\lambda}_0 \bar{d} + \sum_{i=1}^m \bar{\lambda}_i (b_i + B_i \bar{d}) & = & 0, & \\ b_i^T \bar{d} + \frac{1}{2} \bar{d}^T B_i \bar{d} & \leq & 0, & i = 1, 2, \ldots, m, \\ \bar{\lambda}_i (b_i^T \bar{d} + \frac{1}{2} \bar{d}^T B_i \bar{d}) & = & 0, & i = 1, 2, \ldots, m, \\ \bar{\lambda}_0 (\bar{d}^T \bar{d} - \gamma^2) & = & 0. & \end{array} \tag{2.12}$$

If $\bar{\lambda}_0 = 0$, this would imply

$$\sum_{i=1}^m \bar{\lambda}_i (b_i + B_i \bar{d}) = 0,$$

or, after multiplying with p from (2.9), we get

$$-\sum_{i=1}^m \bar{\lambda}_i \frac{\zeta_0}{2} \geq \sum_{i=1}^m \bar{\lambda}_i (b_i + B_i \bar{d})^T p = 0,$$

which implies $\bar{\lambda} = 0$, a contradiction with the assumption that not both $\bar{\lambda}_0$ and $\bar{\lambda}$ are 0. Therefore $\bar{\lambda}_0 > 0$, and from (2.12) we get $\bar{d}^T \bar{d} = \gamma^2 > 0$ and, after dividing

with $\bar{\lambda}_0$,

$$\begin{array}{rcl} \bar{d} + \sum_{i=1}^m \frac{\bar{\lambda}_i}{\bar{\lambda}_0}(b_i + B_i\bar{d}) &=& 0, \\ b_i^T\bar{d} + \frac{1}{2}\bar{d}^T B_i\bar{d} &\leq& 0, \quad i = 1, 2, \ldots, m, \\ \frac{\bar{\lambda}_i}{\bar{\lambda}_0}(b_i^T\bar{d} + \frac{1}{2}\bar{d}^T B_i\bar{d}) &=& 0, \quad i = 1, 2, \ldots, m. \end{array} \tag{2.13}$$

But this means that $\bar{d} \neq 0$ is a stationary point of (2.11) with a Lagrange multiplier $\bar{\lambda}/\bar{\lambda}_0$, which contradicts the properties of our choice of γ_2. The proof is complete. ∎

LEMMA 2.5. *Consider the following QCQP:*

$$\begin{array}{rcl} \min_{d\in\mathbb{R}^n} \quad \Psi(d) &=& a^T d + \frac{1}{2}d^T A d + \frac{1}{2}c_1 d^T d \\ \text{subject to} \quad \Gamma_i(d) &=& b_i^T d + \frac{1}{2}d^T B_i d \leq 0, \quad i = 1, 2, \ldots, m. \end{array} \tag{2.14}$$

Then there exist $\gamma_3 > 0$ and $c_\delta \geq 0$ such that, whenever $c_1 \geq c_\delta$, the only stationary point of (2.14) *that satisfies $||d|| \leq \gamma_3$ is $d = 0$.*

Note that the constraints of (2.14) are the same as those of TRQCQP(γ), with the exception of the trust-region constraint. If d is a stationary point of (2.14) such that $||d|| < \gamma$, then d must be feasible for TRQCQP(γ) with an inactive trust-region constraint.

Proof. Choose $\gamma_3' = \gamma_1$. From (2.10) this implies that for all d with $||d|| \leq \gamma_3'$ (2.9) and the QG (2.2) hold. Since (2.9) implies MFCQ (1.7) at any feasible point of (2.14), this implies in turn that any stationary point of (2.14) that satisfies $||d|| \leq \gamma_3'$ will have a nonempty and bounded Lagrange multiplier set.

Choose now

$$\widetilde{c}_B = \frac{2}{\zeta_0}\left(\max_{i=1,2,\ldots,m} ||B_i|| + 1\right), \tag{2.15}$$

$$\gamma_3 = \min\left\{\frac{1}{2\widetilde{c}_B}, \frac{1}{||A||\,\widetilde{c}_B}, \gamma_3'\right\}, \tag{2.16}$$

$$c_\delta = ||A|| + 2\widetilde{c}_B\,||a|| + 2. \tag{2.17}$$

Take $\tilde{d} \neq 0$ a feasible point of (2.14) such that $\|\tilde{d}\| \leq \gamma_3$. Assume also that $c_1 \geq c_\delta$, as specified in the statement of the lemma. We now estimate the variation of the constraints and objective function in a specific direction from $\tilde{d}$, in order to decide under what conditions $\tilde{d} \neq 0$ can be a stationary point of (2.14). Let the active set at $\tilde{d}$ be

$$\mathcal{B}_{\tilde{d}} = \{i = 1, 2, \ldots, m \mid \Gamma_i(\tilde{d}) = 0\}. \tag{2.18}$$

We estimate the first-order behavior of $\Gamma_i(d)$ in the direction $-\tilde{d} + \beta p$, where p is the vector from (2.9) and $\beta \geq 0$. For $i \in \mathcal{B}_{\tilde{d}}$ we get

$$\begin{array}{rcl} (\nabla_d\Gamma_i(\tilde{d}))^T(-\tilde{d} + \beta p) &=& (b_i + B_i\tilde{d})^T(-\tilde{d} + \beta p) \\ &=& -b_i^T\tilde{d} - \tilde{d}^T B_i\tilde{d} + \beta(b_i + B_i\tilde{d})^T p \\ &=& -b_i^T\tilde{d} - \frac{1}{2}\tilde{d}^T B_i\tilde{d} - \frac{1}{2}\tilde{d}^T B_i\tilde{d} + \beta(b_i + B_i\tilde{d})^T p \\ &\leq& -\beta\frac{\zeta_0}{2} - \frac{1}{2}\tilde{d}^T B_i\tilde{d}, \end{array} \tag{2.19}$$

where we have used (2.9), and that, from (2.18), if $i \in \mathcal{B}_{\tilde{d}}$, then $-b_i^T\tilde{d} - \frac{1}{2}\tilde{d}^T B_i\tilde{d} = 0$.

For the objective function we have that

$$\begin{aligned}(\nabla_d \Psi(\tilde{d}))^T(-\tilde{d}+\beta p) &= (a+A\tilde{d}+c_1\tilde{d})^T(-\tilde{d}+\beta p)\\ &= -a^T\tilde{d}-\tilde{d}^TA\tilde{d}-c_1\tilde{d}^T\tilde{d}+\beta a^Tp+\beta\tilde{d}^TAp+\beta c_1\tilde{d}^Tp\\ &\leq -\sigma_1\tilde{d}^T\tilde{d}-c_1\tilde{d}^T\tilde{d}-\tfrac{1}{2}\tilde{d}^TA\tilde{d}+\beta a^Tp+\beta\tilde{d}^TAp+\beta c_1\tilde{d}^Tp,\end{aligned} \tag{2.20}$$

where we used QG (2.2).

Choose now

$$\beta = \widetilde{c}_B\|\tilde{d}\|^2. \tag{2.21}$$

Using that $||p||=1$, we obtain

$$\begin{aligned}&\left|-\tfrac{1}{2}\tilde{d}^TA\tilde{d}+\beta a^Tp+\beta\tilde{d}^TAp+\beta c_1\tilde{d}^Tp\right|\\ &\quad\leq \|\tilde{d}\|^2\left(\tfrac{1}{2}\,||A||+\widetilde{c}_B\,||a||+\widetilde{c}_B\|\tilde{d}\|\,||A||\,\right)+c_1\widetilde{c}_B\|\tilde{d}\|\|\tilde{d}\|^2\\ &\quad\leq \|\tilde{d}\|^2\left(\tfrac{1}{2}\,||A||+\widetilde{c}_B\,||a||+1\right)+\tfrac{1}{2}c_1\|\tilde{d}\|^2\\ &\quad\leq \tfrac{1}{2}c_1\|\tilde{d}\|^2+\tfrac{1}{2}c_1\|\tilde{d}\|^2=c_1\|\tilde{d}\|^2,\end{aligned} \tag{2.22}$$

where we used that from our choice of γ_3 (2.16) and since $\|\tilde{d}\|\leq\gamma_3$ we have $\widetilde{c}_B\|\tilde{d}\|\leq\frac{1}{2}$ and $\widetilde{c}_B\|\tilde{d}\|\,||A||\leq 1$. We also used the definition of c_δ in (2.17) and that $c_1\geq c_\delta$.

Using (2.22) in (2.20), we get

$$\nabla_d\Psi(\tilde{d})^T(-\tilde{d}+\beta p)\leq -\sigma_1\|\tilde{d}\|^2-c_1\|\tilde{d}\|^2+c_1\|\tilde{d}\|^2<0. \tag{2.23}$$

Using (2.15) and (2.21) in (2.19), we get for all $i\in\mathcal{B}_{\tilde{d}}$

$$\begin{aligned}\nabla_d\Gamma_i(\tilde{d})^T(-\tilde{d}+\beta p) &\leq -\tfrac{2}{\zeta_0}\left(\max_{i=1,2,\ldots,m}||B_i||+1\right)\|\tilde{d}\|^2\tfrac{\zeta_0}{2}+\tfrac{1}{2}\|\tilde{d}\|^2\,||B_i||\\ &\leq -\tfrac{1}{2}\,||B_i||\,\|\tilde{d}\|^2-\|\tilde{d}\|^2<0.\end{aligned} \tag{2.24}$$

From (2.24) and (2.23) we get that if $\tilde{d}\neq 0$ is feasible for (2.14), if $c_1\geq c_\delta$ (see (2.17)) and $\|\tilde{d}\|\leq\gamma_3$ (see (2.16)), then there exists a direction

$$\tilde{\Delta}=-\tilde{d}+\beta p$$

that produces strict decreases in the objective function and the active constraints. Therefore $\tilde{d}$ cannot be a stationary point of (2.14). Otherwise (1.3) implies that there exist the multipliers $\lambda_0\geq 0$, $\lambda\geq 0$, $\lambda\in\mathbb{R}^m$, not all of them of 0, such that

$$\lambda_0\nabla_d\Psi(\tilde{d})+\sum_{i\in\mathcal{B}_{\tilde{d}}}\lambda_i\nabla_d\Gamma_i(\tilde{d})=0.$$

From (2.24) and (2.23) we get, after multiplying with $\tilde{\Delta}$, that

$$0>\lambda_0\nabla_d\Psi(\tilde{d})^T\tilde{\Delta}+\sum_{i\in\mathcal{B}_{\tilde{d}}}\lambda_i\nabla_d\Gamma_i(\tilde{d})^T\tilde{\Delta}=0,$$

which is a contradiction that proves the lemma, with c_δ defined in (2.17) and γ_3 defined in (2.16). ∎

LEMMA 2.6. *There exist $\Lambda_\infty > 0$ and $\gamma_4 > 0$ such that, if $\tilde{d}$ with $\|\tilde{d}\| \leq \gamma_4$ is a stationary point of TRQCQP(γ) with Lagrange multipliers $\lambda \in \mathbb{R}^m$ and $c_1 \in \mathbb{R}$, where $0 < \gamma \leq \gamma_4$, then $||\lambda||_\infty \leq \Lambda_\infty$.*

Proof. We take

$$\gamma_4 = \min\{\gamma_1, \gamma_2, \gamma_3\}, \tag{2.25}$$

where γ_1 is defined in (2.10), γ_2 is the quantity from Lemma 2.4, and γ_3 is the quantity from Lemma 2.5. Lemma 2.4 ensures that the Lagrange multipliers exist at any stationary point of TRQCQP(γ).

Assume the contrary of the conclusion of the lemma: For any $k \in \mathbb{N}$, there exists $\tilde{d}^k$ a stationary point of TRQCQP(γ^k) with $0 < \gamma^k \leq \gamma_4$ and with Lagrange multipliers $\lambda^k \geq 0$, $c_1^k \geq 0$ satisfying $\left|\left|\lambda^k\right|\right|_\infty \geq k$ and the KKT conditions for TRQCQP(γ^k), or

$$\begin{array}{rcll} a + A\tilde{d}^k + \sum_{i=1}^m \lambda_i^k (b_i + B_i\tilde{d}^k) + c_1\tilde{d}^k &=& 0, & \\ b_i^T(\tilde{d}^k) + \frac{1}{2}\tilde{d}^{k^T} B_i \tilde{d}^k &\leq& 0, & i = 1, 2, \ldots, m, \\ \tilde{d}^{k^T}\tilde{d}^k &\leq& (\gamma^k)^2, & \\ \lambda_i^k (b_i^T\tilde{d}^k + \frac{1}{2}\tilde{d}^{k^T} B_i \tilde{d}^k) &=& 0, & i = 1, 2, \ldots, m, \\ c_1^k(\tilde{d}^{k^T}\tilde{d}^k - (\gamma^k)^2) &=& 0. & \end{array} \tag{2.26}$$

By Lemma 2.5, since $\|\tilde{d}^k\| \leq \gamma_3$, we must have $c_1^k < c_\delta$. Since $\left\|\frac{\lambda^k}{||\lambda^k||_\infty}\right\|_\infty = 1$, we can choose λ^* such that for a subsequence k_q, $q \to \infty$, we have $\lim_{q\to\infty} \lambda^{k_q}/\|\lambda^{k_q}\|_\infty = \lambda^*$, with $\|\lambda^*\|_\infty = 1$ and $\lim_{q\to\infty} \tilde{d}^{k_q} = \tilde{d}^*$, where $\|\tilde{d}^*\| \leq \gamma_4$. We can now divide through the first equation of (2.26) with $\|\lambda^{k_q}\|_\infty$ and take the limit as $q \to \infty$ and $\|\lambda^{k_q}\|_\infty \to \infty$. We obtain

$$\sum_{i=1}^m \lambda_i^*(b_i + B_i\tilde{d}^*) = 0.$$

Since $\tilde{d} \leq \gamma_4 \leq \gamma_1$, we can multiply with p and use (2.9) and the fact that $||\lambda^*||_\infty = 1$ to get

$$-\frac{\zeta_0}{2} \geq \sum_{i=1}^m \lambda_i^* p^T(b_i + B_i\tilde{d}^*) = 0,$$

which is a contradiction. This proves the lemma. ∎

THEOREM 2.7. *There exists $\gamma_5 > 0$ such that, for any γ such that $0 < \gamma \leq \gamma_5$, TRQCQP(γ) has the unique stationary point $d = 0$.*

Proof. Choose

$$c_\lambda = \left(m\Lambda_\infty \max_{i=1,2,\ldots,m} ||B_i|| + ||A|| + c_\delta\right), \tag{2.27}$$

$$\gamma_5' = \min\left\{\gamma_1, \gamma_2, \gamma_3, \gamma_4, \frac{\eta}{c_\lambda}\right\}, \tag{2.28}$$

where η is the quantity from Lemma 2.3, c_δ is the quantity from Lemma 2.5, Λ_∞ is the quantity from Lemma 2.6, and γ_j, $j = 1, 2, 3, 4$, are the bounds on the trust regions that ensure that all preceding results hold.

Let $\tilde{d} \neq 0$ be a stationary point of TRQCQP(γ) with $0 < \gamma \leq \gamma_5$. By Lemma 2.4, TRQCQP(γ) satisfies MFCQ (1.7) at $\tilde{d}$. Therefore there exist the Lagrange multipliers $\lambda \geq 0$, $c_1 \geq 0$, which, together with $\tilde{d}$, satisfy (1.6), or

$$\begin{array}{rcll} a + A\tilde{d} + \sum_{i=1}^m \lambda_i(b_i + B_i\tilde{d}) + c_1\tilde{d} & = & 0, & \\ b_i^T(\tilde{d}) + \frac{1}{2}\tilde{d}^T B_i\tilde{d} & \leq & 0, & i = 1, 2, \ldots, m, \\ \tilde{d}^T\tilde{d} & \leq & (\gamma)^2, & \\ \lambda_i(b_i^T\tilde{d} + \frac{1}{2}\tilde{d}^T B_i\tilde{d}) & = & 0, & i = 1, 2, \ldots, m, \\ c_1(\tilde{d}^T\tilde{d} - (\gamma)^2) & = & 0. & \end{array} \tag{2.29}$$

Since $\|\tilde{d}\| \leq \gamma_5' \leq \gamma_3$, Lemma 2.5 applies to give that $c_1 < c_\delta$. Since $\|\tilde{d}\| \leq \gamma_5' \leq \gamma_4$, we have that $||\lambda||_\infty \leq \Lambda_\infty$ from Lemma 2.6. We define

$$-w = A\tilde{d} + \sum_{i=1}^m \lambda_i B_i\tilde{d} + c_1\tilde{d}. \tag{2.30}$$

After applying the triangle inequality and using (2.27), we have that

$$\begin{aligned} ||w|| &\leq \|A\tilde{d}\| + \sum_{i=1}^m \|\lambda_i B_i\tilde{d}\| + \|c_1\tilde{d}\| \\ &\leq \|\tilde{d}\| \left(||A|| + m\Lambda_\infty \left(\max_{i=1,2,\ldots,m} ||B_i|| \right) + c_\delta \right) = c_\lambda \|\tilde{d}\| \leq c_\lambda \frac{\eta}{c_\lambda} = \eta, \end{aligned} \tag{2.31}$$

where in the last inequality we have used (2.28), since $\gamma \leq \gamma_5' \leq \frac{\eta}{c_\lambda}$ and $\|\tilde{d}\| \leq \gamma$. From (2.29) and (2.30) we have that

$$a + \sum_{i=1}^m \lambda_i b_i = w.$$

This implies, from Lemma 2.3, that there exists $\lambda^* \in \mathcal{M}^*$ (a Lagrange multiplier for TRQCQP(γ) at $d = 0$) such that

$$||\lambda - \lambda^*|| \leq c_{\mathcal{M}} ||w|| \quad \text{and} \quad \lambda_i = 0 \Rightarrow \lambda_i^* = 0 \qquad \forall i \in \{1, 2, \ldots, m\}. \tag{2.32}$$

Since $\lambda^* \in \mathcal{M}^*$, it satisfies

$$a + \sum_{i=1}^m \lambda_i^* b_i = 0.$$

Adding the last equality to the first equation in (2.29), dividing by 2, and multiplying with $\tilde{d}^T$, we obtain

$$a^T\tilde{d} + \frac{1}{2}\tilde{d}A\tilde{d} + \frac{1}{2}\sum_{i=1}^m \left[\lambda_i^* b_i^T\tilde{d} + \lambda_i(b_i^T\tilde{d} + \tilde{d}^T B_i\tilde{d}) \right] + \frac{1}{2}c_1\|\tilde{d}\|^2 = 0.$$

We now use the identity $u_1v_1 + u_2v_2 = \frac{1}{2}(u_1 + u_2)(v_1 + v_2) + \frac{1}{2}(u_1 - u_2)(v_1 - v_2)$ as well as the fact that $(\lambda_i^* + \lambda_i)(b_i^T\tilde{d} + \frac{1}{2}\tilde{d}^T B_i\tilde{d}) = 0$, for $i = 1, 2, \ldots, m$, which follows from (2.32) and (2.29), to obtain

$$\begin{aligned} 0 &= a^T\tilde{d} + \frac{1}{2}\tilde{d}A\tilde{d} + \frac{1}{2}\sum_{i=1}^m \left[\frac{1}{2}(\lambda_i^* + \lambda_i)(2b_i^T\tilde{d} + \tilde{d}^T B_i\tilde{d}) - \frac{1}{2}(\lambda_i^* - \lambda_i)(\tilde{d}^T B_i\tilde{d}) \right] + \frac{1}{2}c_1\|\tilde{d}\|^2 \\ &= a^T\tilde{d} + \frac{1}{2}\tilde{d}A\tilde{d} - \frac{1}{4}\sum_{i=1}^m (\lambda_i^* - \lambda_i)(\tilde{d}^T B_i\tilde{d}) + \frac{1}{2}c_1\|\tilde{d}\|^2, \end{aligned}$$

which results in

$$a^T\tilde{d} + \frac{1}{2}\tilde{d}A\tilde{d} + \frac{1}{2}c_1\|\tilde{d}\|^2 = \frac{1}{4}\sum_{i=1}^{m}(\lambda_i^* - \lambda_i)(\tilde{d}^T B_i\tilde{d}). \tag{2.33}$$

Since $\tilde{d}$ is feasible for TRQCQP(γ) and since $\|\tilde{d}\| \le \gamma_5' \le \gamma_1 \le \gamma_1'$, the last inequality following from (2.10), QG (2.2) holds to give that $a^T\tilde{d} + \frac{1}{2}\tilde{d}A\tilde{d} \ge \sigma_1\|\tilde{d}\|^2$. Define

$$c_B = \max_{i=1,2,\ldots,m} ||B_i|| .$$

From (2.32), (2.30), and (2.31) we have $||\lambda^* - \lambda|| \le c_{\mathcal{M}}c_\lambda\|\tilde{d}\|$. Using all these bounds in (2.33), together with the arithmetic-quadratic mean inequality, we get

$$\begin{aligned}\sigma_1\|\tilde{d}\|^2 &\le a^T\tilde{d} + \frac{1}{2}\tilde{d}A\tilde{d} + \frac{1}{2}c_1\|\tilde{d}\|^2 = \frac{1}{4}\sum_{i=1}^{m}(\lambda_i^* - \lambda_i)(\tilde{d}^T B_i\tilde{d})\\ &\le \frac{1}{4}\sqrt{m}c_B\|\tilde{d}\|^2\,||\lambda^* - \lambda|| \le \frac{1}{4}\sqrt{m}c_B\|\tilde{d}\|^2 c_{\mathcal{M}}c_\lambda\|\tilde{d}\|.\end{aligned}$$

Since $\|\tilde{d}\| \ne 0$, from our assumption, we obtain, after dividing through the previous inequality by $\|\tilde{d}\|^2$, that

$$\|\tilde{d}\| \ge \frac{4\sigma_1}{\sqrt{m}c_B c_{\mathcal{M}}c_\lambda}. \tag{2.34}$$

Choose now

$$\gamma_5 = \min\left\{\gamma_5', \frac{2\sigma_1}{\sqrt{m}c_B c_{\mathcal{M}}c_\lambda}\right\}.$$

From (2.34) it follows that the unique stationary point of TRQCQP(γ) with $0 < \gamma \le \gamma_5$ is $\tilde{d} = 0$. The proof is complete. □

3. SQCQP. In this section, we introduce the SQCQP algorithm. We prove that under the conditions set forth in the introduction, the algorithm induces superlinear convergence. Since our main interest is the rate of convergence of the algorithm, we do not address global convergence issues.

We consider the following form of the algorithm:

1. Choose a starting point x^k, $k = 0$.
2. Let $x = x^k$, and determine d^k, a stationary point of

$$\begin{array}{rrcl} \min_{d\in\mathbb{R}^n} & \nabla_x f(x)^T d + \frac{1}{2}d^T\nabla^2_{xx}f(x)d & & \\ \text{subject to} & g_i(x) + \nabla_x g_i(x)^T d + \frac{1}{2}d^T\nabla^2_{xx}g_i(x)d & = & \Gamma_i(x,d) \le 0,\\ & i & = & 1,2,\ldots,m,\\ & d^T d & \le & \gamma^2. \end{array} \tag{3.1}$$

3. Take $x^{k+1} = x^k + d^k$ and $k = k + 1$ and restart.

At every step, the algorithm solves a problem with quadratic constraints and a quadratic objective, none of which are assumed to be convex. We name the above algorithm sequential quadratically constrained quadratic programming or SQCQP.

As outlined in subsection 1.1, we assume without loss of generality that $g_i(x^*) = 0$, for all $i = 1, 2, \ldots, m$, after eventually considering a sufficiently small trust region,

and that the QG (1.1) and MFCQ (1.7) hold at a local solution x^* of the NLP (1.2). From [19, 6] these conditions are equivalent to MFCQ (1.7) and (1.12), which are expressed only in terms of the derivatives of the data up to the second order. We show that (3.1) is feasible for γ fixed and x in some neighborhood of x^*. Since it is also bounded, a stationary point must exist.

Due to the fact that it captures all information up to second order for (1.2) at x^*, the QCQP

$$\begin{array}{lll} \min_{d\in\mathbb{R}^n} & \nabla_x f(x^*)^T d + \frac{1}{2} d^T \nabla^2_{xx} f(x^*) d & \\ \text{subject to} & \nabla_x g_i(x^*)^T d + \frac{1}{2} d^T \nabla^2_{xx} g_i(x^*) d \leq 0, & i = 1, 2, \ldots, m, \\ & d^T d \leq \gamma^2 & \end{array} \tag{3.2}$$

satisfies MFCQ (1.7) and (1.12) at $d = 0$. As a result of [19, 6] it follows that (3.2) satisfies MFCQ (2.1) and QG (2.2). Therefore, all the results from section 2 apply for (3.2). We follow a line of proof similar to the one in section 2.

- Theorem 3.1 proves that MFCQ (1.7) is satisfied by (3.1) in a neighborhood of x^* and that the trust-region constraint is inactive at any stationary point d of (3.1). Corollary 3.2 further insures that, in a neighborhood of x^*, the Lagrange multipliers of (3.1) are uniformly bounded.
- Lemma 3.3 ultimately implies that for any Lagrange multiplier λ at a stationary point d of (3.1) at $x = x^*$ there exists a sufficiently close Lagrange multiplier λ^* at $x = x^*$ whose active subset is included in the active subset of λ. This in turn leads to the conclusions of Lemma 3.4 that $(\lambda_i + \lambda_i^*) g_i(x+d) = o(||d||^2)$ and that $P(x + d) = o(||d||^2)$, where d is a stationary point of (3.1). This helps bound above the variations in the objective function of (3.1) in the proof of Theorem 3.5.
- Theorems 3.5 and 3.6 prove the superlinear convergence of a sequence $x^{k+1} = x^k + d^k$, initiated sufficiently close to x^*, where d^k is any stationary point of (3.1).

THEOREM 3.1. *There exists $\gamma_6 > 0$ and a neighborhood $\mathcal{N}_{\gamma_6}(x^*)$ such that, for any γ with $0 < \gamma \leq \gamma_6$, there exists a neighborhood $\mathcal{N}_\gamma(x^*)$ of x^* such that the following hold:*

(i) *The QCQP (3.1) is feasible for any $x \in \mathcal{N}_\gamma(x^*)$.*

(ii) *For any $x \in \mathcal{N}_{\gamma_6}(x^*)$ and any d with $||d|| \leq \gamma_6$ we have*

$$(\nabla_x g_i(x) + \nabla^2_{xx} g_i(x) d)^T p \leq \frac{-\zeta_0}{2},$$

where ζ_0 and p are the quantities entering the definition of MFCQ (1.7).

(iii) *For any sequence $x^k \in \mathcal{N}_\gamma(x^*)$ that satisfies $x^k \to x^*$ as $k \to \infty$, and with $\tilde{d^k}$ a stationary point of (3.1) at $x = x^k$, we must have $\tilde{d^k} \to 0$ as $k \to \infty$.*

(iv) *The constraint $d^T d \leq \gamma^2$ is inactive for any $x \in \mathcal{N}_\gamma(x^*)$ and d a stationary point of (3.1).*

Note that the size of the neighborhood $\mathcal{N}_\gamma(x^*)$ for which the conclusions of parts (i), (iii), and (iv) hold must be a function of γ. For example, if x is close to x^* but infeasible for (1.2) and if γ is too small, then the trust-region constraint will prevent the satisfaction of the other constraints in (3.1), and (i) will not hold in this case. This implies that for any x infeasible there exists γ sufficiently small such that $x \notin \mathcal{N}_\gamma(x^*)$. Generally, as γ decreases, $\mathcal{N}_\gamma(x^*)$ will decrease as well.

Proof. Since (3.2) satisfies MFCQ (2.1) and the QG (2.2) at $d = 0$, from Theorem 2.7 there exists γ_6' such that, for any $0 < \gamma \leq \gamma_6'$, $\tilde{d} = 0$ is the only stationary point

of (3.2). Choose now γ such that $0 < \gamma \leq \gamma_6'$. Since (3.2) satisfies MFCQ (2.1), then, from [27], for any sufficiently small perturbation of (3.2) we still obtain a feasible NLP. We regard (3.1) as a perturbation of (3.2), and we therefore have, from the fact that f, g are twice continuously differentiable, that there exists a neighborhood $\mathcal{N}_\gamma^2(x^*)$ of x^* such that (3.1) is feasible for any $x \in \mathcal{N}_\gamma^2(x^*)$, which proves part (i) as long as $\mathcal{N}_\gamma(x^*) \subset \mathcal{N}_\gamma^2(x^*)$, as we will later choose $\mathcal{N}_\gamma(x^*)$. We also have that, for all $i = 1, 2, \ldots, m$,

$$\begin{aligned}\left(\nabla_x g_i(x) + \nabla_{xx}^2 g_i(x) d\right)^T p &= \nabla_x g_i(x^*)^T p + \left(\nabla_x g_i(x) - \nabla_x g_i(x^*) + \nabla_{xx}^2 g_i(x) d\right)^T p \\ &\leq \nabla_x g_i(x^*)^T p + c_{2g} \left|\left|x - x^*\right|\right| + c_{2g} \left|\left|d\right|\right| \leq -\zeta_0 + c_{2g} \left|\left|x - x^*\right|\right| + c_{2g} \left|\left|d\right|\right|,\end{aligned}$$

where c_{2g} is a bound on the second derivatives of $g_i(x)$, $i = 1, 2, \ldots, m$, since, from MFCQ (1.7), $||p|| = 1$. If we choose $\gamma_6'' = \frac{\zeta_0}{4c_{2g}}$, d with $||d|| \leq \gamma_6''$, and $\mathcal{N}_{\gamma_6}(x^*) = B(x^*, \frac{\zeta_0}{4c_{2g}})$, we get from the previous bound that, since now $c_{2g} \left|\left|x - x^*\right|\right| \leq \frac{\zeta_0}{4}$ and $c_{2g} \left|\left|d\right|\right| \leq \frac{\zeta_0}{4}$,

$$(\nabla_x g_i(x) + \nabla_{xx}^2 g_i(x) d) p \leq -\frac{\zeta_0}{2},$$

which shows part (ii), after defining $\gamma_6 = \min\{\gamma_6', \gamma_6''\}$. We now choose

$$\mathcal{N}_\gamma^3(x^*) = \mathcal{N}_{\gamma_6}(x^*) \cap \mathcal{N}_\gamma^2(x^*).$$

For $0 < \gamma \leq \gamma_6$, both the conclusions of (i) and (ii) hold. In particular, for any $\gamma \in (0, \gamma_6]$, $x \in \mathcal{N}_\gamma^3(x^*)$, the QCQP (3.1) must have a stationary point since it is feasible and bounded.

Take a sequence x^k that satisfies $x^k \to x^*$ as $k \to \infty$. For k sufficiently large we must have $x^k \in \mathcal{N}_\gamma^3(x^*)$, and thus (3.1) will have a finite stationary point d^k. Assume now that conclusion (iii) does not hold: There exists $\gamma > 0$, with $\gamma \leq \gamma_6$ and a sequence $x^k \to x^*$, such that the corresponding stationary points d^k of (3.1) are bounded below $\|d^k\| \geq c_f > 0$ for all k sufficiently large. Since d^k is a stationary point of (3.1) at $x = x^k$, it must satisfy the first-order necessary conditions (1.3) for some multipliers $\lambda_i^k \geq 0$, $i = 0, 1, \ldots, m+1$, with $\sum_{i=0}^{m+1} \lambda_i^k = 1$:

$$\begin{aligned}\lambda_0^k(\nabla_x f(x^k) + \nabla_{xx}^2 f(x^k) d^k) + \textstyle\sum_{i=1}^m \lambda_i^k(\nabla_x g_i(x^k) + \nabla_{xx}^2 g_i(x^k) d^k) + \lambda_{m+1}^k d^k &= 0 \\ \text{for} \quad i = 1, 2, \ldots, m: \quad \Gamma_i(x^k, d^k) \leq 0, \quad \Gamma_i(x^k, d^k)\lambda_i^k &= 0, \\ (d^k)^T d^k \leq \gamma^2, \quad ((d^k)^T d^k - \gamma^2)\lambda_{m+1}^k &= 0.\end{aligned} \tag{3.3}$$

Since the multipliers $\lambda^k = (\lambda_0^k, \lambda_1^k, \ldots, \lambda_{m+1}^k)$ satisfy $\|\lambda^k\|_1 = 1$ and the direction d^k satisfies $c_f \leq \|d^k\| \leq \gamma$, we can extract a subsequence k_q such that $x^{k_q} \to x^*$, $\lambda^{k_q} \to \lambda^*$, $d^{k_q} \to d^* \neq 0$ as $q \to \infty$. Taking the limit as $q \to \infty$ in (3.3), we obtain from the continuity of all data involved in terms of (x, d) that d^* is a stationary point of (3.2). Since $d^* \neq 0$ this contradicts the outcome of Theorem 2.7 that is valid due to our choice of γ_6. This proves (iii).

Assume now that (iv) does not hold. It then follows that there exists a sequence $x^k \to x^*$ with d^k a stationary point and such that $\|d^k\| = \gamma$. But this contradicts the conclusion of (iii), and thus there exists a neighborhood $\mathcal{N}_\gamma(x^*) \subset \mathcal{N}_\gamma^3(x^*) \subset \mathcal{N}_\gamma^2(x^*)$ such that for $x \in \mathcal{N}_\gamma(x^*)$ any stationary point of (3.1) satisfies $d^T d < \gamma^2$, and for which the conclusions of parts (i), (ii), and (iii) hold. The proof is complete. ∎

COROLLARY 3.2. *Any stationary point of* (3.1) *satisfies the KKT conditions with an inactive trust-region constraint,*

$$\begin{array}{rcl} \nabla_x f(x) + \nabla^2_{xx} f(x)d + \sum_{i=1}^m \lambda_i(\nabla_x g_i(x) + \nabla^2_{xx} g_i(x)d) & = & 0 \\ \text{for} \quad i = 1, 2, \ldots, m: \quad \Gamma_i(x,d) \le 0, \quad \Gamma_i(x,d)\lambda_i & = & 0, \\ d^T d & < & \gamma^2, \end{array} \tag{3.4}$$

for any $0 \le \gamma \le \gamma_6$, $x \in \mathcal{N}_\gamma(x^*)$ *and for some* $\lambda \in R^m$, $\lambda \ge 0$. *Here* γ_6 *is the constant defined in Theorem* 3.1. *There exists* Λ_∞ *such that, for any* $x \in \mathcal{N}_\gamma(x^*)$, *any stationary point* d *of* (3.1), *and any Lagrange multipliers* λ *satisfying the KKT conditions, we have* $||\lambda||_\infty \le \Lambda_\infty$.

Proof. Since $0 < \gamma \le \gamma_6$, then, by Theorem 3.1(iv), we have that for any $x \in \mathcal{N}_\gamma(x^*)$ and any stationary point d, we must have $||d|| < \gamma$. Therefore only the constraints $\Gamma_i(x,d)$, $i = 1, 2, \ldots, m$, can be active at a stationary point d. Then by Theorem 3.1(ii), MFCQ (1.7) is satisfied at d and thus there exist multipliers $\lambda \ge 0$ satisfying the KKT conditions and, in particular,

$$\nabla_x f(x) + \nabla^2_{xx} f(x)d + \sum_{i=1}^m \lambda_i(\nabla_x g_i(x) + \nabla^2_{xx} g_i(x)d) = 0.$$

Multiplying both sides by p, we get

$$\begin{aligned} 0 &= (\nabla_x f(x) + \nabla^2_{xx} f(x)d)^T p + \sum_{i=1}^m \lambda_i(\nabla_x g_i(x) + \nabla^2_{xx} g_i(x)d)^T p \\ &\le (\nabla_x f(x) + \nabla^2_{xx} f(x)d)^T p - ||\lambda||_\infty \frac{\zeta_0}{2}. \end{aligned}$$

Since $||p|| = 1$, $||d|| < \gamma$, and after using the usual norm inequalities, we get

$$||\lambda||_\infty \le \frac{2}{\zeta_0}\left(||\nabla_x f(x)|| + \gamma \left|\left|\nabla^2_{xx} f(x)\right|\right|\right).$$

Since on $\mathcal{N}_\gamma(x^*)$ the expression from the right-hand side is bounded above, there exists Λ_∞ for which the conclusion of this corollary holds. ∎

LEMMA 3.3. *There exist* $\gamma_7 > 0$ *and a constant* $c_* > 0$ *such that for any* γ *with* $0 < \gamma \le \gamma_7$ *there exists a neighborhood* $\mathcal{N}^1_\gamma(x^*)$ *such that, whenever* $x \in \mathcal{N}^1_\gamma(x^*)$ *and for any* d *a stationary point of* (3.1) *with Lagrange multipliers* λ, *there exists* $\lambda^* \in \mathcal{M}(x^*)$ *such that* $||\lambda - \lambda^*|| \le c_*(||x - x^*|| + ||d||)$ *and* $\lambda_i = 0 \Rightarrow \lambda^*_i = 0$ *for all* $i = 1, 2, \ldots, m$.

Proof. Take γ such that $0 < \gamma \le \gamma_6$ and $x \in \mathcal{N}_\gamma(x^*)$. Let d be a stationary point of (3.1) with Lagrange multipliers $\lambda \ge 0$ (which exist from Corollary 3.2). From the KKT conditions we obtain

$$\nabla_x f(x) + \nabla^2_{xx} f(x)d + \sum_{i=1}^m \lambda_i(\nabla_x g_i(x) + \nabla^2_{xx} g_i(x)d) = 0, \tag{3.5}$$

and thus

$$\begin{aligned} &\nabla_x f(x^*) + \sum_{i=1}^m \lambda_i \nabla_x g_i(x^*) \\ &\quad = \nabla_x f(x^*) - \nabla_x f(x) + \sum_{i=1}^m \lambda_i \left(\nabla_x g(x^*) - \nabla_x g(x)\right) - \nabla^2_{xx} f(x)d - \sum_{i=1}^m \lambda_i \nabla^2_{xx} g_i(x)d. \end{aligned} \tag{3.6}$$

Using that $||\nabla_x f(x) - \nabla_x f(x^*)|| \leq c_{2f}\,||x - x^*||$, and that $||\nabla_x g_i(x) - \nabla_x g_i(x^*)|| \leq c_{2g}\,||x - x^*||$, where c_{2f} and c_{2g} are bounds on the second derivatives of f and g, we get from (3.6) and Corollary 3.2 that

$$\begin{aligned}(3.7)\qquad &||\nabla_x f(x^*) + \textstyle\sum_{i=1}^m \lambda_i \nabla_x g_i(x^*)|| \leq c_{2f}\,||x - x^*|| + mc_{2g}\Lambda_\infty\,||x - x^*|| \\ &\quad + c_{2f}\,||d|| + mc_{2g}\Lambda_\infty\,||d|| = (c_{2f} + mc_{2g}\Lambda_\infty)(||x - x^*|| + ||d||).\end{aligned}$$

We choose $\beta = \frac{\eta}{2(c_{2f} + m\Lambda_\infty c_{2g})}$, where η is the quantity from Lemma 2.2. From (3.7) it follows that, for any $\gamma \leq \min\{\beta, \gamma_6\}$ and $x \in \mathcal{N}_\gamma^1(x^*) = \mathcal{N}_\gamma(x^*) \cap B(x^*, \beta)$, we have that, since $||d|| \leq \gamma \leq \beta$ and $||x - x^*|| \leq \beta$,

$$\left|\left|\nabla_x f(x^*) + \sum_{i=1}^m \lambda_i \nabla_x g_i(x^*)\right|\right| \leq \frac{\eta}{2} + \frac{\eta}{2} = \eta.$$

We can therefore apply Lemma 2.2 and (3.7) to get that there exists $\lambda^* \in \mathcal{M}(x^*)$ with the properties required, after taking $\gamma_7 = \min\{\beta, \gamma_6\}$ and $c_* = c_{\mathcal{M}}(c_{2f} + mc_{2g}\Lambda_\infty)$, where $c_{\mathcal{M}}$ is the constant from Lemma 2.3. ∎

LEMMA 3.4. *Let $x \in \mathcal{N}_\gamma^1(x^*)$ and $0 < \gamma \leq \gamma_7$, where γ_7 and $\mathcal{N}_\gamma^1(x^*)$ are defined in Lemma* 3.3. *Let λ be a Lagrange multiplier associated with a stationary point d at x of* (3.1). *Let $\lambda^* \in \mathcal{M}(x^*)$ such that $\lambda_i = 0 \Rightarrow \lambda_i^* = 0$ and such that $||\lambda^*||_\infty \leq \Lambda_\infty$. Then*

(i) $P(x + d) \leq \Theta_P(d)\,||d||^2$,

(ii) $|(\lambda_i + \lambda_i^*) g_i(x + d)| \leq 2\Lambda_\infty \Theta_P(d)\,||d||^2$ *for all* $i = 1, 2, \ldots, m$,

where $\Theta_P(d)$ is a continuous function that satisfies $\Theta_P(0) = 0$. If, in addition, $g \in \mathcal{C}^3(\mathcal{N}_\gamma^1(x^*))$, then there exists a constant C_Θ such that, whenever $||d|| \leq \gamma_7$, we have $\Theta_P(d) \leq C_\Theta\,||d||$.

Proof. Using the first-order Taylor remainder formula [1] for $g_i(y)$ around $y = x$ for $y = x + w$ and the fact that $g_i(x)$ is twice continuously differentiable for $i = 1, 2, \ldots, m$, we obtain that

$$\begin{aligned}(3.8)\qquad g_i(x + w) &= g_i(x) + \nabla_x g_i(x)^T w + \frac{1}{2} w^T \nabla_{xx}^2 g_i(x) w \\ &\quad + \int_0^1 w^T [\nabla_{xx}^2 g_i(x + tw) - \nabla_{xx}^2 g_i(x)] w (1 - t)\,dt \\ &\leq g_i(x) + \nabla_x g_i(x)^T w + \frac{1}{2} w^T \nabla_{xx}^2 g_i(x) w \\ &\quad + ||w||^2 \max_{t \in [0,1]} \|\nabla_{xx}^2 g_i(x + tw) - \nabla_{xx}^2 g_i(x)\|.\end{aligned}$$

Since $\nabla_{xx}^2 g_i(x)$ is a continuous function, it follows that

$$\Theta_i(w) = \max_{x \in \mathcal{N}_\gamma^1(x^*)} \max_{t \in [0,1]} \left|\left|\nabla_{xx}^2 g_i(x + tw) - \nabla_{xx}^2 g_i(x)\right|\right|$$

is a continuous function on $||w|| \leq \gamma_7$ with the property that $\Theta_i(0) = 0$. If, in addition, $g \in \mathcal{C}^3(\mathcal{N}_\gamma^1(x^*))$, then there exists a constant C_Θ^i such that, whenever $||w|| \leq \gamma_7$, we have $\Theta_i(w) \leq C_\Theta^i\,||w||$.

We have that d is a stationary point of (3.1) and, as a result, satisfies $g_i(x) + \nabla_x g_i(x)^T d + \frac{1}{2} d^T \nabla_{xx}^2 g_i(x) d \leq 0$. Replacing w with d in (3.8), we obtain

$$g_i(x + d) \leq \Theta_i(d)\,||d||^2, \qquad i = 1, 2, \ldots, m.$$

We now define

$$\Theta_P(d) = \max_{i=1,2,\ldots,m} \Theta_i(d). \tag{3.9}$$

From the definition of $\Theta_i(d)$ we have that Θ_P is continuous and that $\Theta_P(0) = 0$. If, in addition, $g \in \mathcal{C}^3(\mathcal{N}^1_\gamma(x^*))$, then we can choose

$$C_\Theta = \max_{i=1,2,\ldots,m} C^i_\Theta$$

to obtain that, whenever $||d|| \leq \gamma_7$, we have $\Theta_P(d) \leq C_\Theta \, ||d||$.

From the definition of $P(x)$ in (1.13), we get that, for all $i = 1, 2, \ldots, m$,

$$P(x+d) \leq \max_{i=1,2,\ldots,m} \Theta_i(d) \, ||d||^2 = \Theta_P(d) \, ||d||^2 .$$

This proves point (i). Since λ^* is such that $\lambda_i = 0 \Rightarrow \lambda^*_i = 0$, from our hypothesis, and since d is a stationary point of (3.1) and thus satisfies the complementarity condition

$$\lambda_i \left(g(x) + \nabla_x g(x)^T d + \frac{1}{2} d^T \nabla^2_{xx} g(x) d \right) = 0,$$

this implies that, for $i = 1, 2, \ldots, m$,

$$(\lambda_i + \lambda^*_i) \left(g(x) + \nabla_x g(x)^T d + \frac{1}{2} d^T \nabla^2_{xx} g(x) d \right) = 0,$$

or, by using (3.8),

$$(\lambda_i + \lambda^*_i) \left(g(x+d) - \int_0^1 \left(d^T [\nabla^2_{xx} g_i(x+td) - \nabla^2_{xx} g_i(x)] d \right) (1-t) dt \right) = 0,$$

and thus

$$\begin{aligned} |(\lambda_i + \lambda^*_i) g(x+d)| &= \left| (\lambda_i + \lambda^*_i) \int_0^1 \left(d^T [\nabla^2_{xx} g_i(x+td) - \nabla^2_{xx} g_i(x)] d \right) (1-t) dt \right| \\ &\leq 2\Lambda_\infty \Theta_i(d) \, ||d||^2 \leq 2\Lambda_\infty \Theta_P(d) \, ||d||^2 , \end{aligned}$$

which completes the proof of (ii) and of Lemma 3.4. □

From here on we use extensively that, for h twice continuously differentiable, we have

$$\left\| h(x) - h(\bar{x}) - \frac{(\nabla_x h(x) + \nabla_x h(\bar{x}))^T}{2} (x - \bar{x}) \right\| \leq \psi_{3h}(||x - \bar{x}||) \, ||x - \bar{x}||^2 , \tag{3.10}$$

where x and $\bar{x}$ are points in a neighborhood of x^* and where $\psi_{3h}(z) : \mathbb{R} \to \mathbb{R}$ is a continuous function with $\psi_{3h}(0) = 0$. Indeed by Taylor's theorem we have that there exist continuous functions $\psi^1_{3h}, \widetilde{\psi}_{3h}, \psi^2_{3h} : \mathbb{R} \to \mathbb{R}$ which satisfy $\psi^1_{3h}(0) = \widetilde{\psi}_{3h}(0) = \psi^2_{3h}(0) = 0$ such that

$$\begin{aligned} \left\| h(x) - h(\bar{x}) - \nabla_x h(\bar{x})^T (x - \bar{x}) - \tfrac{1}{2} (x - \bar{x})^T \nabla^2_{xx} h(\bar{x}) (x - \bar{x}) \right\| \\ \leq \psi^1_{3h}(||x - \bar{x}||) \, ||x - \bar{x}||^2 \end{aligned} \tag{3.11}$$

and

$$\left\|\nabla_x h(x) - \nabla_x h(\bar{x}) - \nabla^2_{xx} h(\bar{x})(x-\bar{x})\right\| \leq \widetilde{\psi}_{3h}(||x-\bar{x}||)\,||x-\bar{x}||\,. \tag{3.12}$$

Indeed, by using integration by parts for $h(x)$ we obtain

$$\begin{aligned} h(\bar{x}) &= h(x) + \nabla_x h(x)^T(\bar{x}-x) + \tfrac{1}{2}(\bar{x}-x)^T\nabla^2_{xx}h(x)(\bar{x}-x) \\ &\quad + \int_0^1 (x-\bar{x})^T[\nabla^2_{xx}h(x+t(\bar{x}-x)) - \nabla^2_{xx}h(x)](\bar{x}-x)(1-t)dt. \end{aligned} \tag{3.13}$$

We define $\mathcal{B}(x^*,\epsilon)$ to be the neighborhood of x^*, where x and $\bar{x}$ are chosen. It is immediate that (3.11) holds if we choose

$$\psi^1_{3h}(\alpha) = \max_{x\in\mathcal{B}(x^*,\epsilon)} \max_{t\in[0,1]} \max_{||w||\leq\alpha} \left|\left|\nabla^2_{xx}h(x+tw) - \nabla^2_{xx}h(x)\right|\right| \quad \forall\, 0\leq\alpha\leq 2\epsilon.$$

If, in addition, h is three times continuously differentiable, then there will exist a constant $\widetilde{C}_h$ such that $\psi^1_{3h}(\alpha)\leq\widetilde{C}_h\alpha$, since now the term in the above inequality is bounded above by $L_{3h}\,||w||$ for an appropriate constant L_{3h}.

Inequalities (3.12) and (3.11) in turn imply, after choosing $\psi^2_{3h} = \frac{1}{2}\widetilde{\psi}_{3h}$ and using the Cauchy–Schwarz inequality, that

$$\begin{aligned} &\left\|\tfrac{(\nabla_x h(x)+\nabla_x h(\bar{x}))^T}{2}(x-\bar{x}) - \tfrac{(\nabla_x h(\bar{x})+\nabla_x h(\bar{x}))^T}{2}(x-\bar{x}) - \tfrac{1}{2}(x-\bar{x})^T\nabla^2_{xx}h(\bar{x})(x-\bar{x})\right\| \\ &\qquad\qquad \leq \psi^2_{3h}(||x-\bar{x}||)\,||x-\bar{x}||^2\,. \end{aligned} \tag{3.14}$$

Relation (3.10) now follows by comparing (3.10), (3.11), and (3.14) and taking $\psi_{3h}(z) = \psi^1_{3h}(z) + \psi^2_{3h}(z)$. If h were three times continuously differentiable, then ψ_{3h} would be related to the third derivative of h, from the error formula of trapezoidal integration [1], which is the origin of our subscript notation.

By a preceding argument, if, in addition, $h\in\mathcal{C}^3(\mathcal{N}_h(x^*))$, where $\mathcal{N}_h(x^*)$ is a sufficiently small neighborhood of x^*, then ψ_{3h}, ψ^1_{3h}, $\psi^2_{3h}(z)$, $\tilde{\psi}_{3h}(z)$ can be chosen together with a constant C_h such that, whenever $x,\bar{x}\in\mathcal{N}_h(x^*)$, we have

$$\max\{\psi_{3h}(||x-\bar{x}||), \psi^1_{3h}(||x-\bar{x}||), \psi^2_{3h}(||x-\bar{x}||), \tilde{\psi}_{3h}(||x-\bar{x}||)\} \leq C_h\,||x-x^*||\,. \tag{3.15}$$

THEOREM 3.5. *Let $(x^k)_{k\in\mathbb{N}}$ be a sequence such that $x^k\to x^*$, $x^k\neq x^*$. Let d^k be a stationary point of* (3.1) *for $x=x^k$ for $0<\gamma\leq\gamma_7$, where γ_7 is the quantity from Lemma* 3.3. *Then*

$$\lim_{k\to\infty}\frac{\left||x^k+d^k-x^*|\right|}{||x^k-x^*||} = 0.$$

If, in addition, the data $f(x), g(x)$ of the NLP (1.2) *are three times continuously differentiable, then there exists a constant C_ψ such that, for all k sufficiently large, we will have that*

$$\left||x^k+d^k-x^*|\right| \leq C_\psi\left||x^k-x^*|\right|^{\frac{3}{2}}.$$

Proof. Since $x^k\to x^*$, the sequence x^k eventually reaches $\mathcal{N}^1_\gamma(x^*)$. Since $0<\gamma\leq\gamma_7$, this means that Lemmas 3.4 and 3.3, as well as all preceding results, apply for sufficiently large k. Using (3.10), we get that

$$\begin{aligned} f(x^k+d^k) - f(x^*) &\leq \tfrac{1}{2}\left(\nabla_x f(x^k+d^k) + \nabla_x f(x^*)\right)^T(x^k+d^k-x^*) \\ &\quad + \psi_{3f}(\|x^k+d^k-x^*\|)\|x^k+d^k-x^*\|^2 \\ &\leq \tfrac{1}{2}\left(\nabla_x f(x^k) + \nabla^2_{xx}f(x^k)d^k + \nabla_x f(x^*)\right)^T(x^k+d^k-x^*) \\ &\quad + \widetilde{\psi}_{3f}(\|d^k\|)\|d^k\|\|x^k+d^k-x^*\| + \psi_{3f}(\|x^k+d^k-x^*\|)\|x^k+d^k-x^*\|^2, \end{aligned} \tag{3.16}$$

where $\widetilde{\psi}_{3f}(\|d^k)\|\|d^k\|$ is a bound obtained by using (3.12) for $f(x)$ between $x^k + d^k$ and x^k. Here $\widetilde{\psi}_{3f}$ is a continuous function satisfying $\widetilde{\psi}_{3f}(0) = 0$.

From Corollary 3.2, there exists the Lagrange multiplier λ^k, which, together with d^k, satisfies the KKT conditions for (3.1) at $x = x^k$ with an inactive trust-region constraint,

$$\begin{array}{rrcl} \nabla_x f(x^k) + \nabla^2_{xx} f(x^k) d^k + \sum_{i=1}^m \lambda_i^k (\nabla_x g_i(x^k) + \nabla^2_{xx} g_i(x^k) d^k) & = & 0 \\ \text{for} \quad i = 1, 2, \ldots, m: \quad \Gamma_i(x^k, d^k) \le 0, \quad \Gamma_i(x^k, d^k)\lambda_i^k & = & 0, \\ (d^k)^T d^k & < & \gamma^2. \end{array} \tag{3.17}$$

From Lemma 3.3, there exists a $\lambda^{*k} \in \mathcal{M}(x^*)$ such that

$$\|\lambda^k - \lambda^{*k}\| \le c_*(\|x^k - x^{*k}\| + \|d^k\|) \ \text{ and } \ \lambda_i^k = 0 \Rightarrow \lambda_i^{*k} = 0. \tag{3.18}$$

Using (3.17) and the KKT conditions (1.6) to replace $\nabla_x f(x^k) + \nabla^2_{xx} f(x^k) d^k$ and $\nabla_x f(x^*)$ in terms of g and the Lagrange multipliers, and using the bounds $\|\lambda^k\|_\infty \le \Lambda_\infty$, $||\lambda^*||_\infty \le \Lambda_\infty$, that follow from Corollary 3.2, we get from (3.16) that

$$\begin{aligned} & f(x^k + d^k) - f(x^*) \\ & \quad \le \tfrac{1}{2} \left(- \sum_{i=1}^m \lambda_i^k \left(\nabla_x g_i(x^k) + \nabla^2_{xx} g_i(x^k) d^k \right) - \sum_{i=1}^m \lambda_i^{*k} \nabla_x g_i(x^*) \right)^T (x^k + d^k - x^*) \\ & \qquad + \widetilde{\psi}_{3f}(\|d^k\|)\|d^k\|\|x^k + d^k - x^*\| + \psi_{3f}(\|x^k + d^k - x^*\|)\|x^k + d^k - x^*\|^2 \\ & \quad \le \tfrac{1}{2} \left(- \sum_{i=1}^m \lambda_i^k \nabla_x g_i(x^k + d^k) - \sum_{i=1}^m \lambda_i^{*k} \nabla_x g_i(x^*) \right)^T (x^k + d^k - x^*) \\ & \qquad + m\Lambda_\infty \widetilde{\psi}_{3g}(\|d^k\|)\|d^k\|\|x^k + d^k - x^*\| + \widetilde{\psi}_{3f}(\|d^k\|)\|d^k\|x^k + d^k - x^* \\ & \qquad + \psi_{3f}(\|x^k + d^k - x^*\|)\|x^k + d^k - x^*\|^2, \end{aligned} \tag{3.19}$$

where $\widetilde{\psi}_{3g}(\|d^k\|)\|d^k\|$ is a bound obtained from applying (3.12) to $g_i(x)$, $i = 1, 2, \ldots, m$, between the points $x^k + d^k$ and x^k and taking the maximum among the resulting bounds. Here $\widetilde{\psi}_{3g}$ is a continuous function satisfying $\widetilde{\psi}_{3g}(0) = 0$. We now make use of the identity $ab + cd = \frac{1}{2}[(a + c)(b + d) + (a - c)(b - d)]$ for the terms $(\lambda_i^k \nabla_x g_i(x^k + d^k)^T + \lambda_i^{*k} \nabla_x g_i(x^*)^T)(x^k + d^k - x^*)$, $i = 1, 2, \ldots, m$. Continuing the bounding in (3.19), we get

$$\begin{aligned} f(x^k + d^k) - f(x^*) \le & -\frac{1}{4} \left(\sum_{i=1}^m \left(\lambda_i^k + \lambda_i^{*k} \right) \left(\nabla_x g_i(x^k + d^k) + \nabla_x g_i(x^*) \right) \right. \\ & \left. + \sum_{i=1}^m \left(\lambda_i^k - \lambda_i^{*k} \right) \left(\nabla_x g_i(x^k + d^k) - \nabla_x g_i(x^*) \right) \right)^T (x^k + d^k - x^*) \\ & + \left(m\Lambda_\infty \widetilde{\psi}_{3g}(\|d^k\|) + \widetilde{\psi}_{3f}(\|d^k\|) \right) \|d^k\|\|x^k + d^k - x^*\| \\ & + \psi_{3f}(\|x^k + d^k - x^*\|)\|x^k + d^k - x^*\|^2. \end{aligned} \tag{3.20}$$

We now bound all terms involving λ and λ^*. Using that $\|\lambda - \lambda^*\| \le c_*(\|x^k - x^*\| + \|d^k\|)$ from (3.18) and that g is twice continuously differentiable and thus

$$\left|\left| \nabla_x g_i(x^k + d^k) - \nabla_x g_i(x^*) \right|\right| \le c_{2g} \left|\left| x^k + d^k - x^* \right|\right|, \quad i = 1, 2, \ldots, m,$$

we get

$$\begin{aligned} & -\tfrac{1}{4} \sum_{i=1}^m (\lambda_i^k - \lambda_i^{*k}) \left(\nabla_x g_i(x^k + d^k) - \nabla_x g_i(x^*) \right)^T (x^k + d^k - x^*) \\ & \qquad \le m c_* c_{2g} (\|x^k - x^*\| + \|d^k\|)\|x^k + d^k - x^*\|^2. \end{aligned} \tag{3.21}$$

Using that $||\lambda||_\infty \leq \Lambda_\infty$ from Corollary 3.2, (3.10) for $g_i(x)$, and that $g_i(x^*) = 0$, $i = 1, 2, \ldots, m$, as well as Lemma 3.4 (ii), we obtain that

$$\begin{aligned} &-\tfrac{1}{4}\textstyle\sum_{i=1}^m(\lambda_i^k + \lambda_i^{*k})(\nabla_x g_i(x^k + d^k) + \nabla_x g_i(x^*))^T(x^k + d^k - x^*) \\ &\quad \leq -\tfrac{1}{2}\textstyle\sum_{i=1}^m(\lambda_i^k + \lambda_i^{*k})g_i(x^k + d^k) \\ &\qquad + m\Lambda_\infty\psi_{3g}(\|x^k + d^k - x^*\|)\|x^k + d^k - x^*\|^2 \\ &\quad \leq m\Lambda_\infty\Theta_P(d^k)\|d^k\|^2 + m\Lambda_\infty\psi_{3g}(\|x^k + d^k - x^*\|)\|x^k + d^k - x^*\|^2. \end{aligned} \tag{3.22}$$

Putting together the bounds from (3.20), (3.21), and (3.22), we obtain

$$\begin{aligned} f(x^k + d^k) - f(x^*) \leq \big(&mc_*c_{2g}(\|x^k - x^*\| + \|d^k\|) + m\Lambda_\infty\psi_{3g}(\|x^k + d^k - x^*\|) \\ &+ \psi_{3f}(\|x^k + d^k - x^*\|)\big)\|x^k + d^k - x^*\|^2 \\ + \big(m\Lambda_\infty\widetilde{\psi}_{3g}(\|d^k\|) + \widetilde{\psi}_{3f}(\|d^k\|)\big)&\|d\|\|x^k + d^k - x^*\| + m\Lambda_\infty\Theta_P(d^k)\|d_k\|^2. \end{aligned} \tag{3.23}$$

Since the bound on the right-hand side is nonnegative, we can use Lemma 3.4(i) and the QG (1.14) to get that

$$\begin{aligned} \sigma\left|\left|x^k + d^k - x^*\right|\right|^2 &\leq \max\left\{f(x^k + d^k) - f(x^*), P(x^k + d^k)\right\} \\ &\leq \Phi_1(x^k - x^*, d^k)\left|\left|x^k + d^k - x^*\right|\right|^2 \\ &+ \Phi_2(d^k)\left|\left|d^k\right|\right|\left|\left|x^k + d^k - x^*\right|\right| \\ &+ m\Lambda_\infty\Theta_P(d^k)\left|\left|d^k\right|\right|^2 + \Theta_P(d^k)\left|\left|d^k\right|\right|^2, \end{aligned} \tag{3.24}$$

where

$$\begin{aligned} \Phi_1(x^k - x^*, d^k) &= mc_*c_{2g}(\|x^k - x^*\| + \|d^k\|) \\ &\quad + m\Lambda_\infty\psi_{3g}(\|x^k + d^k - x^*\|) + \psi_{3f}(\|x^k + d^k - x^*\|), \\ \Phi_2(d^k) &= m\Lambda_\infty\widetilde{\psi}_{3g}(\|d^k\|) + \widetilde{\psi}_{3f}(\|d^k\|). \end{aligned}$$

Here Φ_1 and Φ_2 are continuous functions of their arguments that satisfy $\Phi_1(0, 0) = 0$ and $\Phi_2(0) = 0$. If, in addition, we have that $f(x), g(x)$, the data of NLP (1.2), are three times continuously differentiable, then, by taking $\bar{x}^k = x^k + d^k$ in (3.15) and using that $\|d^k\| \leq \gamma_7$, we find that there exist $\mathcal{N}_\psi^1(x^*)$, a suitably small neighborhood of x^*, and a constant C_ψ^1 such that, whenever $x^k \in \mathcal{N}_\psi^1(x^*)$, we have that $\Phi_2(d^k) \leq C_\psi^1\|d^k\|$. We now use that $ab \leq \frac{1}{2}(a^2 + b^2)$ to get from (3.24) that

$$\begin{aligned} \sigma\left|\left|x^k + d^k - x^*\right|\right|^2 \leq &\left(\Phi_1(x^k - x^*, d^k) + \tfrac{1}{2}\Phi_2(d^k)\right)\left|\left|x^k + d^k - x^*\right|\right|^2 \\ &+ \left(\tfrac{1}{2}\Phi_2(d^k) + (m\Lambda_\infty + 1)\Theta_P(d^k)\right)\left|\left|d^k\right|\right|^2. \end{aligned} \tag{3.25}$$

From Theorem 3.1(iii), and since Φ_1 and Φ_2 are continuous mappings, we obtain that there exists a neighborhood $\mathcal{N}_\psi^2(x^*)$ such that whenever $x^k \in \mathcal{N}_\psi^2(x^*)$ we have

$$\Phi_1(x^k - x^*, d^k) + \frac{1}{2}\Phi_2(d^k) \leq \frac{\sigma}{2}. \tag{3.26}$$

Indeed, if such a neighborhood $\mathcal{N}_\psi^2(x^*)$ would not exist, then there would exist a subsequence x^{k_q}, $x^{k_q} \to x^*$ as $q \to \infty$, such that

$$\Phi_1(x^{k_q} - x^*, d^{k_q}) + \frac{1}{2}\Phi_2(d^{k_q}) > \frac{\sigma}{2}.$$

However, by Theorem 3.1(iii) we must have $d^{k_q} \to 0$. By taking $q \to \infty$ in the preceding inequality we obtain a contradiction with the continuity of Φ_1 and Φ_2 and the fact that $\Phi_1(0,0) = \Phi_2(0) = 0$.

Taking the term from (3.26) to the right-hand side of (3.25), we obtain that whenever $x^k \in \mathcal{N}^2_\psi(x^*)$ we have that

$$\frac{\sigma}{2}\left|\left|x^k + d^k - x^*\right|\right|^2 \leq \left(\frac{1}{2}\Phi_2(d^k) + (m\Lambda_\infty + 1)\Theta_P(d^k)\right)\left|\left|d^k\right|\right|^2. \tag{3.27}$$

Now, using the continuity of Φ_2 and Θ_P and that, from Theorem 3.1(iii), $d_k \to 0$, we get that

$$\lim_{k\to\infty} \frac{\left|\left|x^k + d^k - x^*\right|\right|^2}{\left|\left|d^k\right|\right|^2} \leq \frac{2}{\sigma}\lim_{k\to\infty}\left(\frac{1}{2}\Phi_2(d^k) + (m\Lambda_\infty + 1)\Theta_P(d^k)\right) = 0$$

or that

$$\lim_{k\to\infty} \frac{\left|\left|x^k + d^k - x^*\right|\right|}{||d^k||} = 0. \tag{3.28}$$

Using now the consequence of the triangle inequality

$$\left|\left|\left|x^k - x^*\right|\right| - \left|\left|d^k\right|\right|\right| \leq \left|\left|x^k - x^* + d^k\right|\right|$$

and dividing the relation with $\left|\left|d^k\right|\right|$ and taking the limit, this implies that

$$\lim_{k\to\infty}\left|\frac{\left|\left|x^k - x^*\right|\right|}{||d^k||} - 1\right| \leq \lim_{k\to\infty}\frac{\left|\left|x^k - x^* + d^k\right|\right|}{||d^k||} = 0,$$

and thus

$$\lim_{k\to\infty} \frac{\left|\left|x^k - x^*\right|\right|}{||d^k||} = 1. \tag{3.29}$$

Dividing (3.28) by the last limit, we get that

$$\lim_{k\to\infty} \frac{\left|\left|x^k + d^k - x^*\right|\right|}{||x^k - x^*||} = 0,$$

which proves the first part of the claim of the theorem. If, in addition, we assume that the data of the problem is three times continuously differentiable, then, since $\|d^k\| \leq \gamma_7$ and by using Lemma 3.4 and our previous results for Φ_2, we obtain that

$$x^k \in \mathcal{N}^1_\gamma(x^*) \cap \mathcal{N}^1_\psi(x^*) \Rightarrow \Phi_2(d^k) \leq C^1_\psi\left|\left|d^k\right|\right| \quad \text{and} \quad \Theta_P(d^k) \leq C_\Theta\left|\left|d^k\right|\right|.$$

Using these relationships in (3.27) and choosing

$$C^2_\psi = \frac{1}{2}C^1_\psi + (m\Lambda_\infty + 1)C_\Theta,$$

we obtain that, whenever $x^k \in \mathcal{N}^1_\gamma(x^*) \cap \mathcal{N}^1_\psi(x^*) \cap \mathcal{N}^2_\psi(x^*)$, we must have that

$$\left|\left|x^k + d^k - x^*\right|\right|^2 \leq \frac{2}{\sigma}C^2_\psi\left|\left|d^k\right|\right|^3$$

or

$$\left|\left|x^k + d^k - x^*\right|\right| \leq \sqrt{\frac{2}{\sigma} C_\psi^2} \left|\left|d^k\right|\right|^{\frac{3}{2}}.$$

From (3.29) it follows that, for k sufficiently large, we must have that $\|d^k\| \leq 2\|x^k - x^*\|$. Therefore, for k sufficiently large we must have that

$$\left|\left|x^k + d^k - x^*\right|\right| \leq 2^{\frac{3}{2}} \sqrt{\frac{2}{\sigma} C_\psi^2} \left|\left|x^k - x^*\right|\right|^{\frac{3}{2}}.$$

The claim follows after choosing

$$C_\psi = 2^{\frac{3}{2}} \sqrt{\frac{2}{\sigma} C_\psi^2}. \qquad \square$$

THEOREM 3.6. *Let γ be such that $0 < \gamma \leq \gamma_7$, where γ_7 is the quantity from Lemma* 3.3. *There exists a radius r^* such that for any $x \in B(x^*, r^*)$, $x \neq x^*$; if d is a stationary point of* (3.1), *then*

$$\frac{||x + d - x^*||}{||x - x^*||} \leq \frac{1}{2}.$$

Whenever started inside $B(x^*, r^*)$, the SQCQP algorithm produces a sequence $x^k \to x^*$ that is superlinearly convergent,

$$\lim_{k \to \infty} \frac{\left|\left|x^{k+1} - x^*\right|\right|}{||x^k - x^*||} = 0.$$

If, in addition, the data of the NLP is three times continuously differentiable, then for k sufficiently large we will have

$$\left|\left|x^{k+1} - x^*\right|\right| \leq C_\psi \left|\left|x^k - x^*\right|\right|^{\frac{3}{2}},$$

where C_ψ is the constant from Theorem 3.5.

Proof. Assume the contrary: For any $q \in \mathbb{N}$, there exists $x^q \neq x^*$ such that $||x^q - x^*|| \leq \frac{1}{q}$ and d^q a stationary point of (3.1) such that

$$\frac{||x^q + d^q - x^*||}{||x^q - x^*||} \geq \frac{1}{2}. \tag{3.30}$$

Therefore $x^q \to x^*$, and by Theorem 3.5

$$\lim_{q \to \infty} \frac{||x^q + d^q - x^*||}{||x^q - x^*||} = 0,$$

which contradicts (3.30). As a result there exists r^* with the properties required by the theorem. When started with $x^0 \in B(x^*, r^*)$, the SQCQP algorithm produces a sequence $x^{k+1} = x^k + d^k$ such that

$$\frac{\left|\left|x^1 - x^*\right|\right|}{||x^0 - x^*||} \leq \frac{1}{2},$$

which implies $x^1 \in B(x^*, r^*)$ and thus, by induction, $x^k \in B(x^*, r^*)$ for all $k \in \mathbb{N}$ and $x^k \to x^*$ as $k \to \infty$. We can now use Theorem 3.5 to claim that

$$\lim_{k\to\infty} \frac{\left|\left|x^{k+1} - x^*\right|\right|}{||x^k - x^*||} = \lim_{k\to\infty} \frac{\left|\left|x^k + d^k - x^*\right|\right|}{||x^k - x^*||} = 0,$$

which proves the superlinear convergence of x^k to x^*. The proof for the case in which the data of the problem is three times continuously differentiable follows immediately from Theorem 3.5. The proof is complete. □

Under the assumptions considered here, it is possible that, for x^k in a neighborhood of x^*, (3.1) will have multiple stationary points. This comes from the fact that the QG (1.1), which holds at $x = x^*$ for (3.1), is generally not maintained under perturbations, as opposed to other, stronger, second-order conditions [7]. In this case x can be considered a perturbation parameter whose nominal value is $x = x^*$. Note, however, that in our results we do not assume uniqueness of the stationary points of (3.1). Any stationary point d^k of (3.1) at $x = x^k$ will induce superlinear convergence of the sequence x^k to x^*.

3.1. Comparison with SQP. An appealing class of methods for approaching NLP (1.2) is the one of SQP algorithms, solving at each point x^k a quadratic program (QP) using the Hessian of the Lagrangian $\mathcal{L}(x, \lambda)$ (see (1.5)) as the matrix of the QP

$$\begin{array}{lrcl} \min_{d\in\mathbb{R}^n} & \nabla_x f(x)^T d + \frac{1}{2} d^T \nabla^2_{xx} \mathcal{L}(x,\lambda) d & & \\ \text{subject to} & g_i(x) + \nabla_x g_i(x)^T d & \leq & 0, \quad i = 1, 2, \ldots, m, \\ & d^T d & \leq & \gamma^2, \end{array} \tag{3.31}$$

where λ is some estimate of the Lagrange multipliers. This is the approach used, for example, by FilterSQP [12], with an infinity norm trust region instead of a two norm trust region. Take the following example, used in [2]:

$$\begin{array}{lll} & \min_{(x,y,z)} z & \\ \text{subject to} & g_0(x,y,z) = x^2 - 2y^2 - z & \leq 0, \\ & g_1(x,y,z) = -\frac{1}{2}(x^2 + y^2) + 3xy - z & \leq 0, \\ & g_2(x,y,z) = -2x^2 + y^2 - z & \leq 0, \\ & g_3(x,y,z) = -\frac{1}{2}(x^2 + y^2) - 3xy - z & \leq 0. \end{array} \tag{3.32}$$

The global solution of this problem is $x^* = (0, 0, 0)^T$, at which all inequalities are active. At x^*, both MFCQ (1.7) and the QG (1.1) hold. The Lagrange multiplier set is

$$\mathcal{M}(x^*) = \left\{ \lambda = (\lambda_1, \lambda_2, \lambda_3, \lambda_4) \in \mathbb{R} \;\middle|\; \text{such that } \lambda_i \geq 0,\ i = 1, 2, 3, 4,\ \sum_{i=1}^{4} \lambda_i = 1 \right\}.$$

Choose now $\lambda^* = \left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right)^T$. We immediately have that

$$\nabla^2_{xx} \mathcal{L}(x^*, \lambda^*) = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

By denoting $d = (d_x, d_y, d_z)$, we obtain that the QP (3.31) at (x^*, λ^*) is

$$\begin{array}{lrcl} \min_{d_x,d_y,d_z} & -\frac{1}{2}\left(d_x^2 + d_y^2\right) + d_z & & \\ \text{subject to} & -d_z & \leq & 0, \\ & d_x^2 + d_y^2 + d_z^2 & \leq & \gamma^2, \end{array} \tag{3.33}$$

since the gradients of all constraints, except the trust-region constraints, are equal to $(0, 0, -1)^T$. One of the multiple global solutions of this subproblem is $(\gamma, 0, 0)$. Therefore, although we are at a solution and we use the exact Hessian and Lagrange multipliers, $d = 0$ is not a solution of the above QP but only a stationary point. For this problem we have that any choice of Lagrange multiplier λ^* results in a Hessian of the Lagrangian that has negative curvature along some direction of $\mathcal{C}(x^*)$ (see [2]). Therefore any choice of λ^* will encounter the same phenomenon: $d = 0$ is not a solution of (3.31) but merely a stationary point.

Therefore, if (3.31) were used as a subproblem in an SQP algorithm under the assumptions considered here, then it could have stationary points at which the trust region is active, no matter how close x is to the solution x^*. This sits in contrast with the QCQP subproblem, which Theorem 3.1 ensures will have an inactive trust-region constraint near the solution. It thus seems that subproblem (3.31) is not a good local representation of (1.2).

It is thus possible that, close to the solution x^*, an algorithm that uses a global solution of subproblem (3.31) will actually tend to move away from the solution. This situation can be countered by the use of a good globalization strategy. For example, FilterSQP will likely reject such an iterate, but will have to shrink the trust region, which is an unwanted effect close to the solution.

When we have applied FilterSQP, for example (4.3), which is closely related to (3.32), we have not observed this behavior, probably because FilterSQP has locked onto one of the stationary points of the QP subproblem for which the trust region is inactive. However, it seems difficult to guarantee a priori that an algorithm will ignore a global solution of (3.31) and instead return one of the stationary points closer to 0.

4. Numerical examples. We present numerical runs on three examples with the algorithm presented in this work. We extend the scope of our method to include equality constraints.

To that end, we assume that the NLP to solve is

$$\begin{array}{rlcll} \min_x & f(x) & & & \\ \text{subject to} & g_i(x) & \leq & 0, & i = 1, 2, \ldots, m, \\ & h_j(x) & = & 0, & j = 1, 2, \ldots, p. \end{array} \tag{4.1}$$

The extension of MFCQ (1.7), the QG (1.1), and second-order sufficient conditions (1.12) to the case where equality constraints also hold is immediate [7, 22, 23, 10], as long as the equality constraints have linearly independent gradients at the solution x^*. For clarity, we have analyzed here only the inequality constraints case, though all results from the preceding sections extend fairly straightforwardly to the equality constrained case. The trust-region SQCQP algorithm for (4.1) is modified as follows.

- Choose a starting point x^k, $k = 0$.
- Let $x = x^k$ and determine d^k, a stationary point of

$$\begin{array}{rll} \min_{d \in \mathbb{R}^n} & \nabla_x f(x)^T d + \frac{1}{2} d^T \nabla^2_{xx} f(x) d & \\ \text{subject to} \quad g_i(x) + \nabla_x g_i(x)^T d + \frac{1}{2} d^T \nabla^2_{xx} g_i(x) d & \leq 0, & i = 1, 2, \ldots, m, \\ h_j(x) + \nabla_x h_j(x)^T d + \frac{1}{2} d^T \nabla^2_{xx} h_j(x) d & = 0, & j = 1, 2, \ldots, p, \\ d^T d & \leq \gamma^2. & \end{array} \tag{4.2}$$

- Take $x^{k+1} = x^k + d^k$ and $k = k + 1$ and restart.

We call one iteration of this algorithm an outer iteration. We determine a stationary point of QCQP subproblem (4.2) by using Powell's variable metric approach [3, pp. 287–289]. Powell's algorithm solves at every step l a convex QP whose constraints are a linearization of the constraints of (4.2) and whose objective matrix is obtained by a quasi-Newton update. If the sequence of matrices H^l produced in this fashion is uniformly positive definite and bounded, then Powell's algorithm will stop only at stationary points of (4.2), which is sufficient for our purposes, from Theorem 3.6. To enforce that, if the estimate of the condition number or the norm of H^l exceeds a certain value, then the matrix H^l is reset to the identity. Currently, there is no theory to guarantee that Powell's algorithm will converge superlinearly to a stationary point of (4.2) for degenerate problems like this one. Nevertheless, we use the variable metric approach instead of just choosing $H^l = I$ to provide a reasonable scaling. The resulting QP was solved with the subroutine *quadprog*, available from the Matlab Optimization Toolbox. The tolerance parameters for *quadprog* were set to 10^{-10}.

Therefore d^k is determined by solving a sequence of QPs, none of which requires any additional function or derivative evaluation of the original problem.

As a merit function we use the L_∞ exact penalty function [4]. We have not implemented a second-order correction, needed to avoid the Maratos effect [3, 4], since our interest lies primarily in the behavior of the outer iteration which produces the sequence x^k. Also, the second-order correction is theoretically founded in the limit only for the case in which the gradients of the active constraints are linearly dependent, which we do not expect to hold for our examples.

Since we want to demonstrate the local behavior of the SQCQP algorithm, we have not implemented a trust-region update, and the trust-region parameter was chosen to be the fixed value $\gamma = 10$. Also, no line-search or merit criteria was used for globalization of the outer iteration. The algorithm was implemented in Matlab using the ADMIT-1 toolbox [9] for computing the derivatives of the problem data.

It is clear that, for nondegenerate NLPs, classical nonlinear programming algorithms will be more efficient than the SQCQP algorithm since fast local convergence can be achieved at a cost of one QP per iteration [12, 15]. By contrast, SQCQP needs several QPs per iteration. Therefore we will present the behavior of the algorithm on three degenerate (or nearly degenerate) examples and compare it with FilterSQP [12] and SNOPT [15], two SQP algorithms. Both FilterSQP and SNOPT were run with the relevant tolerances set to 10^{-10}.

4.1. Degenerate examples. We consider three examples. In the first example, the problem has no locally convex augmented Lagrangian at the solution:

$$
\begin{aligned}
& \min_{(x,y,z)} z \\
\text{subject to} \quad & g_0(x,y,z) = x^2 + 3x^4 - 2y^2 - z && \le 0, \\
& g_1(x,y,z) = -\tfrac{1}{2}(x^2+y^2) + y^4 + 3xy - z && \le 0, \\
& g_2(x,y,z) = -2x^2 + y^2 - z && \le 0, \\
& g_3(x,y,z) = -\tfrac{1}{2}(x^2+y^2) + x^2y^2 - 3xy - z && \le 0.
\end{aligned} \tag{4.3}
$$

This example is closely related to example (3.32). The global solution of the problem is $x^* = [0,0,0]^T$. At x^*, all constraints are active and their gradients are equal to $[0,0,-1]^T$, which makes the problem degenerate. However, at x^* both MFCQ (1.7) and QG (1.1) are satisfied, and thus the results from this work apply. All algorithms are started from the point $x_0 = [1,1,1]^T$. The second example has a locally convex augmented Lagrangian for some of the Lagrange multipliers, but not

for others:

$$
\begin{array}{lll}
 & \min z & \\
\text{subject to} & g_1(x,y,z) = x^2 - 2y^2 + y^4 - z & \leq 0, \\
 & g_2(x,y,z) = -cx^2 + y^2 + x^2y^2 - z & \leq 0, \\
 & g_3(x,y,z) = -x^2 - y^2 - z & \leq 0.
\end{array} \tag{4.4}
$$

If $c < 0.5$, then the example has as an unique global solution of $x^* = [0,0,0]^T$. Again, at x^* all constraints are active, and their gradients are equal to $[0,0,-1]^T$, which makes the problem degenerate. Also, both MFCQ (1.7) and the QG (1.1) hold at the solution, which guarantees that the theoretical results presented in this work will hold.

The size of the set of Lagrange multipliers for which the corresponding augmented Lagrangian may be locally convex is controlled by the parameter c, a smaller c resulting in a larger such set. If such multipliers can be detected, then stabilization techniques can be used in a neighborhood of such multipliers to induce superlinear convergence to the primal-dual solution set [16, 29]. The type of degeneracy is thus less severe when compared to (4.3). In this work no attempt is made to detect such multipliers, and the SQCQP algorithm is directly applied for $c = 0.49$. All algorithms are started from the point $x_0 = [1,1,1]^T$.

The last example is *allinitc*, a member of the CUTE collection [8] which is a standard benchmark for nonlinear programming. The NLP has four variables and four convex constraints. All algorithms are started from $x_0 = [0,0,0,0]^T$. Our conclusion that the problem is nearly degenerate originates in the fact that SNOPT enters the elastic mode when applied to this problem, which is a sign that the Lagrange multiplier set is large and thus that the constraints are nearly degenerate. If the Lagrange multiplier set were unbounded, then MFCQ (1.7) would not hold at the solution. Although the assumptions under which we proved our results will hold only marginally for this example, or not at all, we still apply the SQCQP algorithm in order to compare its performance with SQP algorithms.

4.2. Results of the numerical runs. On the examples above, we have run our SQCQP implementation, as presented above, as well as FilterSQP and SNOPT, two SQP algorithms. FilterSQP uses second-order derivatives and solves nonconvex QPs as the main subproblem [12]. SNOPT uses only first-order information and solves convex QPs whose objective matrices are obtained by a quasi-Newton update [15]. FilterSQP and SNOPT examples were run using the AMPL interface on the NEOS server [24]. Because the examples were run on different platforms, the running times will not be relevant for a comparison. We therefore report performance metrics that are platform-independent, such as the number of function evaluations and QPs solved.

The results are presented in Table 4.1. "PB" represents the index of the problem, "DS" represents the distance to solution (when available), "FE" represents the number of function evaluations, while "QP" represents the number of quadratic programs. "Obj" and "Infeas" are the values of the objective function and of the maximum infeasibility of the constraints at the final iterate. For the third example, the exact solution is not available, and so the distance to the solution cannot be computed. In Table 4.2 we present the convergence behavior of the final iterations of SQCQP for the first two examples, for which exact solutions are available. A similar table cannot be constructed for SNOPT or FilterSQP, since we do not have access to the intermediate iterates x^k. In Table 4.2, "nrQP" represents the number of QPs solved at a particular iteration.

TABLE 4.1
Results of the numerical runs.

PB	Solver	Obj	Infeas	DS	FE	QP
Ex. 1	SQCQP	$7.875039e-19$	$1.12e-16$	$1.06e-8$	9	112
	SNOPT	$-2.857327e-14$	$3.60e-11$	$5.97e-6$	20	18
	FilterSQP	$-2.135276e-21$	$6.55e-11$	$7.85e-6$	19	18
Ex. 2	SQCQP	$1.176702e-17$	$6.06e-19$	$5.24e-8$	10	92
	SNOPT	$-2.049969e-13$	$0.00e-00$	$1.60e-5$	22	20
	FilterSQP	$-1.625107e-20$	$6.59e-12$	$8.34e-6$	18	19
Ex. 3	SQCQP	$3.049655e+01$	$2.22e-16$	*	5	288
	SNOPT	$3.049433e+01$	$6.30e-09$	*	95	47
	FilterSQP	$3.049652e+01$	$8.73e-12$	*	24	28

TABLE 4.2
Convergence behavior of SQCQP.

PB	Iteration	$\|\|x^k - x^*\|\|$	nrQP
Ex1	6	$1.65e-03$	12
	7	$1.15e-04$	15
	8	$8.54e-08$	4
	9	$1.06e-08$	1
Ex2	7	$9.00e-03$	20
	8	$7.02e-05$	12
	9	$6.78e-08$	8
	10	$5.24e-08$	1

We can see from Table 4.1 that, on the three examples presented here, SQCQP needs between 2 and 19 times fewer function evaluations and between 6 and 10 times more QPs to produce results of a quality that is comparable to or better than FilterSQP or SNOPT. Both results are to be expected, since the subproblem used by SQCQP is a more accurate, though harder to solve, representation of (4.1) than those used by either FilterSQP or SNOPT.

It can also be seen from Table 4.2 that for the first example we get a superlinearly convergent behavior before encountering rounding error effects. For the second example the sequence does not exhibit the classical superlinear convergence behavior, but rather a very fast linear convergence behavior, very likely because it appears by the time the rounding errors are encountered. Both SNOPT and FilterSQP exhibit a linear convergence behavior with a reasonable rate, though much larger than SQCQP.

The SQCQP algorithm seems promising, in that it needs fewer function evaluations to reach a near-optimal point, which is perhaps the most used metric for evaluating the performance of an algorithm. If the trend observed here were to replicate for the general class of nonlinear programming, then the question of overall efficiency would depend on the computational effort needed for evaluating the data and its derivatives, as opposed to the computational effort spent in solving the QPs. For problems for which the evaluation of data and its derivatives is very expensive, it may be worth using SQCQP in spite of the larger number of QPs required. For the cases presented here, which have quite simple problem data, the SQCQP algorithm would have very likely been outperformed by either of FilterSQP and SNOPT in terms of run time, since the largest portion of the cost comes from solving the QPs (though, in the case of FilterSQP, the QPs to be solved are nonconvex).

The key element in making SQCQP an efficient algorithm is to devise an efficient way of solving the QCQP subproblem. It should be pointed out that, for the experiments used in this work, we were limited by the techniques that were available

under Matlab or were reasonably easy to implement, and by the fact that there is no obvious way to take advantage of the solution of an instance of (4.2) for solving succeeding instances. Currently, we do not even take advantage of the knowledge of the active set between iterations. Looking at Table 4.1, it is conceivable that if we used an algorithm like FilterSQP directly on subproblem (4.2) when solving the third example, then we would need less than $5 \times 28 = 140$ quadratic programming solves. We observed for our examples that the final iterations of SQCQP take substantial steps towards the solution while using a moderate number of QPs, as can be seen from Table 4.1. A promising avenue would thus be to start with a classical algorithm and switch to SQCQP close to the solution, a strategy that needs further analysis and numerical evaluation.

5. Conclusions. We present an algorithm that achieves superlinear convergence of the iterates to a local minimum of NLP (1.2) at which MFCQ (1.7) and QG (1.1) are satisfied. The conditions we impose allow even situations for which no locally convex augmented Lagrangian exists, a case not accommodated by most previous results in the literature.

At each step we solve a subproblem generated by approximating the function and the constraints by the second-order Taylor series at the current iterate. We also add a trust-region constraint, which ensures that the problem is bounded. The algorithm therefore solves at each step a quadratically constrained quadratic program (QCQP) and we thus call it sequential quadratically constrained quadratic program (SQCQP).

The subproblem to be solved is not necessarily convex. However we prove that for a suitable, fixed size of the trust region the associated constraint is inactive at any stationary point of the QCQP. As a result, any stationary point of the QCQP induces superlinear convergence of the iterates, which obviates the need for finding the global optimum of the subproblem. In subsection 3.1 we showed that, in contrast to our SQCQP algorithm, SQP algorithms that solve QPs at each iteration with the exact Hessian in the objective function and a trust-region constraint may find their solution on the boundary of the trust region.

A subproblem that has quadratic constraints is more difficult to solve than a subproblem with linear constraints, the latter being the case of subproblems solved by SQP algorithms [25]. However, since SQCQP incorporates a more accurate model of the constraints than does SQP, it would be expected that a smaller number of exterior iterations and thus of function evaluations would be needed before completion. We demonstrate this point by solving three examples and showing that SQCQP needs fewer function evaluations to converge to a solution than do two widely used SQP algorithms. In the current implementation this comes with the cost of solving more QPs.

Devising methods to solve the QCQP subproblems efficiently, perhaps as a replacement for their QP counterparts in the latest stages of a classical SQP algorithm, will be the subject of future research.

REFERENCES

[1] K. Atkinson, *An Introduction to Numerical Analysis*, John Wiley & Sons, New York, 1988.
[2] M. Anitescu, *Degenerate nonlinear programming with a quadratic growth condition,* SIAM J. Optim., 10 (2000), pp. 1116–1135.
[3] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, New York, 1982.
[4] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, MA, 1995.

[5] J.F. Bonnans, *Local analysis of Newton-type methods for variational inequalities and nonlinear programming*, Appl. Math. Optim., 29 (1994), pp. 161–186.
[6] J. F. Bonnans and A. Ioffe, *Second-order sufficiency and quadratic growth for nonisolated minima*, Math. Oper. Res., 20 (1995), pp. 801–819.
[7] J. F. Bonnans and A. Shapiro, *Perturbation Analysis of Optimization Problems*, Springer-Verlag, New York, 2000.
[8] I. Bongartz, A. R. Conn, N. Gould, and Ph. L. Toint, *CUTE: Constrained and unconstrained testing environment*, ACM Trans. Math. Software, 21 (1995), pp. 123–160.
[9] T. Coleman and A. Verma, *ADMIT*-1: *Automatic Differentiation and MATLAB Interface Toolbox*, Technical report CS TR 98-1663, Department of Computer Science, Cornell University, Ithaca, NY, 1998.
[10] A. V. Fiacco, *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*, Academic Press, New York, 1983.
[11] R. Fletcher, *Practical Methods of Optimization*, John Wiley & Sons, Chichester, UK, 1987.
[12] R. Fletcher and S. Leyffer, *Nonlinear programming without a penalty function*, Math. Program., 91 (2002), pp. 239–270.
[13] J. Gauvin, *A necessary and sufficient regularity condition to have bounded multipliers in nonconvex programming*, Math. Programming, 12 (1977), pp. 136–138.
[14] J. Gauvin and J. W. Tolle, *Differential stability in nonlinear programming*, SIAM J. Control Optim., 15 (1977), pp. 294–311.
[15] P. E. Gill, W. Murray, and M. A. Saunders, *User's guide for SNOPT* 5.3: *A Fortran Package for Large-Scale Nonlinear Programming*, Report NA 97-5, Department of Mathematics, University of California, San Diego, CA, 1997.
[16] W. W. Hager, *Stabilized sequential quadratic programming*, Comput. Optim. Appl., 12 (1999), pp. 253–273.
[17] W. W Hager and M. S. Gowda, *Stability in the presence of degeneracy and error estimation*, Math. Program., 85 (1999), pp. 181–192.
[18] A. J. Hoffman, *On approximate solutions of systems of linear inequalities*, J. Res. National Bureau of Standards, 49 (1952), pp. 263–265.
[19] A. D. Ioffe, *Necessary and sufficient conditions for a local minimum.* 3: *Second order conditions and augmented duality*, SIAM J. Control Optim., 17 (1979), pp. 266–288.
[20] A. Ioffe, *On sensitivity analysis of nonlinear programs in Banach spaces: The approach via composite unconstrained optimization*, SIAM J. Optim., 4 (1994), pp. 1–43.
[21] S. Kruk and H. Wolkowicz, *Sequential, quadratic constrained, quadratic programming for general nonlinear programming*, in Handbook of Semidefinite Programming, H. Wolkowicz, R. Saigal, and L. Vandenberghe, eds., Kluwer Academic Publishers, Boston, 2000, pp. 563–575.
[22] O. L. Mangasarian, *Nonlinear Programming*, McGraw–Hill, New York, 1969.
[23] O. L. Mangasarian and S. Fromovitz, *The Fritz John necessary optimality conditions in the presence of equality constraints*, J. Math. Anal. Appl., 17 (1967), pp. 34–47.
[24] *The NEOS Guide,* available online at http://www.mcs.anl.gov/otc/Guide.
[25] E. Polak, *Optimization*, Springer-Verlag, New York, 1997.
[26] D. Ralph and S. J. Wright, *Superlinear Convergence of an Interior-Point Method Despite Dependent Constraints*, preprint ANL/MCS-P622-1196, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1996.
[27] S. M. Robinson, *Generalized equations and their solutions, Part* II: *Applications to nonlinear programming,* Math. Programming Study, 19 (1980), pp. 200–221.
[28] A. Shapiro, *Sensitivity analysis of nonlinear programs and differentiability properties of metric projections*, SIAM J. Control Optim., 26 (1988), pp. 628–645.
[29] S. J. Wright, *Superlinear convergence of a stabilized SQP method to a degenerate solution*, Comput. Optim. Appl., 11 (1998), pp. 253–275.
[30] S. J. Wright, *Modifying SQP for Degenerate Problems*, preprint ANL/MCS-P699-1097, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1997.

SIAM J. OPTIM.
Vol. 12, No. 4, pp. 979–1006

SNOPT: AN SQP ALGORITHM FOR LARGE-SCALE CONSTRAINED OPTIMIZATION*

PHILIP E. GILL†, WALTER MURRAY‡, AND MICHAEL A. SAUNDERS‡

Abstract. Sequential quadratic programming (SQP) methods have proved highly effective for solving constrained optimization problems with smooth nonlinear functions in the objective and constraints. Here we consider problems with general inequality constraints (linear and nonlinear). We assume that first derivatives are available and that the constraint gradients are sparse.

We discuss an SQP algorithm that uses a smooth augmented Lagrangian merit function and makes explicit provision for infeasibility in the original problem and the QP subproblems. SNOPT is a particular implementation that makes use of a semidefinite QP solver. It is based on a limited-memory quasi-Newton approximation to the Hessian of the Lagrangian and uses a reduced-Hessian algorithm (SQOPT) for solving the QP subproblems. It is designed for problems with many thousands of constraints and variables but a moderate number of degrees of freedom (say, up to 2000). An important application is to trajectory optimization in the aerospace industry. Numerical results are given for most problems in the CUTE and COPS test collections (about 900 examples).

Key words. large-scale optimization, nonlinear programming, nonlinear inequality constraints, sequential quadratic programming, quasi-Newton methods, limited-memory methods

AMS subject classifications. 49J20, 49J15, 49M37, 49D37, 65F05, 65K05, 90C30

PII. S1052623499350013

1. Introduction.

We present a sequential quadratic programming (SQP) method for large-scale optimization problems involving general linear and nonlinear constraints. SQP methods have proved reliable and efficient for many such problems. For example, under mild conditions the general-purpose solvers NLPQL [70], NPSOL [44, 47], and DONLP [73] typically find a (local) optimum from an arbitrary starting point, and they require relatively few evaluations of the problem functions and gradients compared to traditional solvers such as MINOS [58, 59, 60] and CONOPT [26].

1.1. The optimization problem.

The algorithm we describe applies to constrained optimization problems of the form

$$\text{(NP)} \qquad \underset{x\in\mathbb{R}^n}{\text{minimize}} \quad f(x)$$
$$\text{subject to} \quad l \le \begin{pmatrix} x \\ c(x) \\ Ax \end{pmatrix} \le u,$$

where $f(x)$ is a linear or nonlinear objective function, $c(x)$ is a vector of nonlinear constraint functions $c_i(x)$ with sparse derivatives, A is a sparse matrix, and l and u

*Received by the editors January 11, 1999; accepted for publication (in revised form) November 15, 2001; published electronically April 19, 2002.
http://www.siam.org/journals/siopt/12-4/35001.html

†Department of Mathematics, University of California, San Diego, La Jolla, CA 92093-0112 (pgill@ucsd.edu). The research of this author was supported by National Science Foundation grants DMI-9424639, CCR-9896198, and DMS-9973276.

‡Department of Management Science & Engineering, Stanford University, Stanford, CA 94305-4026 (walter@stanford.edu, saunders@stanford.edu). The research of this author was supported by National Science Foundation grants DMI-9500668 and CCR-9988205, and Office of Naval Research grant N00014-96-1-0274.

are vectors of lower and upper bounds.

We assume that the nonlinear functions are smooth and that their first derivatives are available (and possibly expensive to evaluate). For the present implementation we further assume that the number of active constraints at a solution is reasonably close to n. In other words, the number of degrees of freedom is not too large (say, less than 2000).

Important examples are control problems such as those arising in optimal trajectory calculations. For many years, the optimal trajectory system OTIS (Hargraves and Paris [51]) has been applied successfully within the aerospace industry, using NPSOL to solve the associated optimization problems. NPSOL is a transformed Hessian method that treats the Jacobian of the general constraints as a dense matrix and updates an explicit quasi-Newton approximation to $Q_k^T H_k Q_k$, the transformed Hessian of the Lagrangian, where Q_k is orthogonal. The QP (quadratic programming) subproblem is solved using a linearly constrained linear least-squares method that exploits the properties of the transformed Hessian.

Although NPSOL has solved OTIS examples with two thousand constraints and over a thousand variables, the need to handle increasingly large models has provided strong motivation for the development of new sparse SQP algorithms. Our aim is to describe a new SQP method that has the favorable theoretical properties of the NPSOL algorithm but is suitable for a broad class of large problems, including those arising in trajectory optimization. The implementation is called SNOPT (sparse nonlinear optimizer) [41]. Extensive numerical results are given in section 6.

The method of SNOPT exploits sparsity in the constraint Jacobian and maintains a limited-memory quasi-Newton approximation to H_k (not a full transformed Hessian $Q_k^T H_k Q_k$). A new method is used to update H_k in the presence of negative curvature. The QP subproblems are solved using an inertia-controlling reduced-Hessian active-set method that allows for variables to appear linearly in the objective and constraint functions. (The limited-memory Hessian is then semidefinite.) Other features include the treatment of infeasible nonlinear constraints using elastic programming, use of a well-conditioned nonorthogonal basis for the null-space of the QP working set, and early termination of the QP subproblems.

1.2. Infeasible constraints. SNOPT deals with infeasibility using ℓ_1 penalty functions. First, infeasible linear constraints are detected by solving a problem of the form

$$\text{(FLP)} \qquad \underset{x,v,w}{\text{minimize}} \quad e^T(v+w)$$
$$\text{subject to} \quad l \le \begin{pmatrix} x \\ Ax - v + w \end{pmatrix} \le u, \quad v \ge 0, \quad w \ge 0,$$

where e is a vector of ones and v and w are handled implicitly. This is equivalent to minimizing the one-norm of the general linear constraint violations subject to the simple bounds (often called *elastic programming* in the linear programming literature [11]). Elastic programming has long been a feature of the XS system of Brown and Graves [12]. Other algorithms based on minimizing one-norms of infeasibilities are given by Conn [21] and Bartels [1].

If the linear constraints are infeasible ($v \neq 0$ or $w \neq 0$), SNOPT terminates without computing the nonlinear functions. Otherwise, all subsequent iterates satisfy the linear constraints. (Sometimes this feature helps ensure that the functions and gradients are well defined; see section 5.2.)

SNOPT then proceeds to solve (NP) as given, using QP subproblems based on linearizations of the nonlinear constraints. If a QP subproblem proves to be infeasible or unbounded (or if the Lagrange multiplier estimates for the nonlinear constraints become large), SNOPT enters "nonlinear elastic" mode and solves the problem

$$\begin{array}{ll} (\mathrm{NP}(\gamma)) & \underset{x,v,w}{\text{minimize}} \quad f(x) + \gamma e^T(v+w) \\ & \text{subject to} \quad l \le \begin{pmatrix} x \\ c(x) - v + w \\ Ax \end{pmatrix} \le u, \quad v \ge 0, \quad w \ge 0, \end{array}$$

where $f(x) + \gamma e^T(v+w)$ is called a *composite objective*, and the penalty parameter γ ($\gamma \ge 0$) may take a finite sequence of increasing values. If (NP) has a feasible solution and γ is sufficiently large, the solutions to (NP) and (NP(γ)) are identical. If (NP) has no feasible solution, (NP(γ)) will tend to determine a "good" infeasible point if γ is again sufficiently large. (If γ were infinite, the nonlinear constraint violations would be minimized subject to the linear constraints and bounds.)

A similar ℓ_1 formulation of (NP) is used in the SQP method of Tone [76] and is fundamental to the Sℓ_1QP algorithm of Fletcher [30]. See also Conn [20] and Spellucci [72]. An attractive feature is that only linear terms are added to (NP), giving no increase in the expected degrees of freedom at each QP solution.

1.3. Other work on large-scale SQP. There has been considerable interest in extending SQP methods to the large-scale case (sometimes using exact second derivatives). Some of this work has focused on problems with nonlinear *equality* constraints. The method of Lalee, Nocedal, and Plantenga [53], related to the trust-region method of Byrd [15] and Omojokun [61], uses either the exact Lagrangian Hessian or a limited-memory quasi-Newton approximation defined by the method of Zhu et al. [79]. The method of Biegler, Nocedal, and Schmid [3] is in the class of *reduced-Hessian methods*, which maintain a dense approximation to the reduced Hessian, using quasi-Newton updates.

For large problems with general inequality constraints as in problem (NP), SQP methods have been proposed by Eldersveld [28], Tjoa and Biegler [75], Fletcher and Leyffer [32], and Betts and Frank [2]. The first three approaches are also reduced-Hessian methods. Eldersveld forms a full Hessian approximation from the reduced Hessian, and his implementation LSSQP solves the same class of problems as SNOPT. In Tjoa and Biegler's method, the QP subproblems are solved by eliminating variables using the (linearized) equality constraints, and the remaining variables are optimized using a dense QP solver. As bounds on the eliminated variables become dense constraints in the reduced QP, the method is best suited to problems with many nonlinear equality constraints but few bounds on the variables. The filter-SQP method of Fletcher and Leyffer uses a reduced Hessian QP-solver in conjunction with an exact Lagrangian Hessian. This method is also best suited for problems with few degrees of freedom. In contrast, the method of Betts and Frank employs an exact or finite-difference Lagrangian Hessian and a QP solver based on sparse KKT factorizations (see section 7). It is therefore applicable to problems with many degrees of freedom.

Several large-scale methods solve the QP subproblems by an interior method. They typically require an exact or finite-difference Lagrangian Hessian and can accommodate many degrees of freedom. Examples are Boggs, Kearsley, and Tolle [4, 5] and Sargent and Ding [69].

1.4. Other large-scale methods. MINOS and CONOPT are both reduced-Hessian methods. Like SNOPT, they use first derivatives and are designed for large problems with few degrees of freedom (again up to 2000, say, although MINOS can allow for any number; see section 7.1). For nonlinear constraints, MINOS uses a *linearly constrained Lagrangian* method, whose subproblems require frequent evaluation of the problem functions. CONOPT uses a *generalized reduced gradient* method, which maintains near-feasibility with respect to the nonlinear constraints, again at the expense of many function evaluations. SNOPT is likely to outperform MINOS and CONOPT when the functions (and their derivatives) are expensive to evaluate. Relative to MINOS, an added advantage is the existence of a merit function to ensure global convergence. This is especially important when the constraints are highly nonlinear.

LANCELOT Release A [22] is another widely used package in the area of large-scale constrained optimization. It uses a *bound constrained augmented Lagrangian* method. In general, LANCELOT is recommended for large problems with many degrees of freedom. It complements SNOPT and the other methods discussed above. A comparison between LANCELOT and MINOS has been made in [8, 9].

LOQO [78] and KNITRO [17, 16] are examples of large-scale optimization packages that treat inequality constraints by a primal-dual interior method. Both packages require second derivatives but can accommodate many degrees of freedom.

1.5. Notation. Some important quantities follow:

(x, π, s)	primal, dual and slack variables for problem (GNP) (see section 2.1),
(x^*, π^*, s^*)	optimal variables for problem (GNP),
(x_k, π_k, s_k)	the kth estimate of (x^*, π^*, s^*),
f_k, g_k, c_k, J_k	functions and gradients evaluated at x_k,
$(\widehat{x}_k, \widehat{\pi}_k, \widehat{s}_k)$	optimal variables for QP subproblem (GQP$_k$) (see section 2.4).

2. The SQP iteration. Here we discuss the main features of an SQP method for solving a generic nonlinear program. All features are readily specialized to the more general constraints in problem (NP).

2.1. The generic problem. In this section we take the problem to be

(GNP)	$\underset{x}{\text{minimize}} \quad f(x)$
	subject to $\quad c(x) \geq 0,$

where $x \in \mathbb{R}^n$, $c \in \mathbb{R}^m$, and the functions $f(x)$ and $c_i(x)$ have continuous second derivatives. The gradient of f is denoted by the vector $g(x)$, and the gradients of each element of c form the rows of the Jacobian matrix $J(x)$.

We assume that a KKT point (x^*, π^*) exists for (GNP), satisfying the first-order optimality conditions:

$$c(x^*) \geq 0, \qquad \pi^* \geq 0, \qquad c(x^*)^T\pi^* = 0, \qquad J(x^*)^T\pi^* = g(x^*). \tag{2.1}$$

2.2. Structure of the SQP method. An SQP method obtains search directions from a sequence of QP subproblems. Each QP subproblem minimizes a quadratic model of a certain Lagrangian function subject to linearized constraints. Some merit function is reduced along each search direction to ensure convergence from any starting point.

The basic structure of an SQP method involves *major* and *minor* iterations. The major iterations generate a sequence of iterates (x_k, π_k) that converge to (x^*, π^*). At each iterate a QP subproblem is used to generate a search direction towards the next iterate (x_{k+1}, π_{k+1}). Solving such a subproblem is itself an iterative procedure, with the *minor* iterations of an SQP method being the iterations of the QP method.

For an overview of SQP methods, see, for example, Boggs and Tolle [6], Fletcher [31], Gill, Murray, and Wright [48], Murray [56], and Powell [66].

2.3. The modified Lagrangian. Let x_k and π_k be estimates of x^* and π^*. For several reasons, our SQP algorithm is based on the *modified Lagrangian* associated with (GNP), namely,

$$\mathcal{L}(x, x_k, \pi_k) = f(x) - \pi_k^T d_L(x, x_k), \tag{2.2}$$

which is defined in terms of the *constraint linearization* and the *departure from linearity*:

$$\begin{aligned} c_L(x, x_k) &= c_k + J_k(x - x_k), \\ d_L(x, x_k) &= c(x) - c_L(x, x_k); \end{aligned}$$

see Robinson [68] and Van der Hoek [77]. The first and second derivatives of the modified Lagrangian with respect to x are

$$\begin{aligned} \nabla\mathcal{L}(x, x_k, \pi_k) &= g(x) - (J(x) - J_k)^T\pi_k, \\ \nabla^2\mathcal{L}(x, x_k, \pi_k) &= \nabla^2 f(x) - \sum_i (\pi_k)_i \nabla^2 c_i(x). \end{aligned}$$

Observe that $\nabla^2\mathcal{L}$ is independent of x_k (and is the same as the Hessian of the conventional Lagrangian). At $x = x_k$, the modified Lagrangian has the same function and gradient values as the objective: $\mathcal{L}(x_k, x_k, \pi_k) = f_k,\ \nabla\mathcal{L}(x_k, x_k, \pi_k) = g_k$.

2.4. The QP subproblem. Let the quadratic approximation to $\mathcal{L}$ at x_k be

$$\mathcal{L}_Q(x, x_k, \pi_k) = f_k + g_k^T(x - x_k) + \tfrac{1}{2}(x - x_k)^T\nabla^2\mathcal{L}(x_k, x_k, \pi_k)(x - x_k).$$

If $(x_k, \pi_k) = (x^*, \pi^*)$, optimality conditions for the QP

(GQP*)	$\underset{x}{\text{minimize}}$	$\mathcal{L}_Q(x, x_k, \pi_k)$	
	subject to	linearized constraints	$c_L(x, x_k) \geq 0$

are identical to those for the original problem (GNP). This suggests that if H_k is an approximation to $\nabla^2\mathcal{L}$ at the point (x_k, π_k), an improved estimate of the solution may be found from $(\widehat{x}_k, \widehat{\pi}_k)$, the solution of the following QP subproblem:

(GQP$_k$)	$\underset{x}{\text{minimize}}$	$f_k + g_k^T(x - x_k) + \frac{1}{2}(x - x_k)^T H_k(x - x_k)$
	subject to	$c_k + J_k(x - x_k) \geq 0.$

Optimality conditions for (GQP$_k$) may be written as

$$\begin{aligned} c_k + J_k(\widehat{x}_k - x_k) &= \widehat{s}_k, & \widehat{\pi}_k &\geq 0, \quad \widehat{s}_k \geq 0, \\ g_k + H_k(\widehat{x}_k - x_k) &= J_k^T\widehat{\pi}_k, & \widehat{\pi}_k^T\widehat{s}_k &= 0, \end{aligned}$$

where $\widehat{s}_k$ is a vector of slack variables for the linearized constraints. In this form, $(\widehat{x}_k, \widehat{\pi}_k, \widehat{s}_k)$ may be regarded as estimates of (x^*, π^*, s^*), where the slack variables s^* satisfy $c(x^*) - s^* = 0,\ s^* \geq 0$. The vector $\widehat{s}_k$ is needed explicitly for the line search (section 2.7).

2.5. The working-set matrix W_k. The *working set* is an important quantity for both the major and the minor iterations. It is the current estimate of the set of constraints that are binding at a solution. More precisely, suppose that (GQP_k) has just been solved. Although we try to regard the QP solver as a "black box," we expect it to return an independent set of constraints that are active at the QP solution (even if the QP constraints are degenerate). This is an optimal working set for subproblem (GQP_k).

The same constraint indices define a working set for (GNP) (and for subproblem (GQP_{k+1})). The corresponding gradients form the rows of the *working-set matrix* W_k, an $n_Y \times n$ full-rank submatrix of the Jacobian J_k.

2.6. The null-space matrix Z_k. Let Z_k be an $n \times n_Z$ full-rank matrix that spans the null space of W_k. (Thus, $n_Z = n - n_Y$, and $W_k Z_k = 0$.) The QP solver will often return Z_k as part of some matrix factorization. For example, in NPSOL it is part of an orthogonal factorization of W_k, while in LSSQP [28] (and in the current SNOPT) it is defined implicitly from a sparse LU factorization of part of W_k. In any event, Z_k is useful for theoretical discussions, and its column dimension has strong practical implications. Important quantities are the *reduced Hessian* $Z_k^T H_k Z_k$ and the *reduced gradient* $Z_k^T g$.

2.7. The merit function. Once the QP solution $(\widehat{x}_k, \widehat{\pi}_k, \widehat{s}_k)$ has been determined, new estimates of the (GNP) solution are computed using a line search on the augmented Lagrangian merit function

$$\mathcal{M}(x, \pi, s) = f(x) - \pi^T\big(c(x) - s\big) + \tfrac{1}{2}\big(c(x) - s\big)^T D\big(c(x) - s\big), \tag{2.3}$$

where D is a diagonal matrix of penalty parameters. If (x_k, π_k, s_k) are the current estimates of (x^*, π^*, s^*), the line search determines a step length α_k $(0 < \alpha_k \leq 1)$ such that the new point

$$\begin{pmatrix} x_{k+1} \\ \pi_{k+1} \\ s_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ \pi_k \\ s_k \end{pmatrix} + \alpha_k \begin{pmatrix} \widehat{x}_k - x_k \\ \widehat{\pi}_k - \pi_k \\ \widehat{s}_k - s_k \end{pmatrix} \tag{2.4}$$

gives a *sufficient decrease* in the merit function (2.3). Let $\varphi_k(\alpha)$ denote the merit function computed at the point $(x_k + \alpha(\widehat{x}_k - x_k), \pi_k + \alpha(\widehat{\pi}_k - \pi_k), s_k + \alpha(\widehat{s}_k - s_k))$; i.e., $\varphi_k(\alpha)$ defines $\mathcal{M}$ as a univariate function of the step length. Initially D is zero (for $k = 0$). When necessary, the penalties in D are increased by the minimum-norm perturbation that ensures *sufficient descent* for $\varphi_k(\alpha)$ [47]. (Note: As in NPSOL, s_{k+1} in (2.4) is redefined to minimize the merit function as a function of s, prior to the solution of (GQP_{k+1}). For more details, see [44, 28].)

In the line search, for some vector $b > 0$ the following condition is enforced:

$$c(x_k + \alpha_k p_k) \geq -b \qquad (p_k \equiv \widehat{x}_k - x_k). \tag{2.5}$$

We use $b_i = \tau_V \max\{1, -c_i(x_0)\}$, where τ_V is a specified constant, e.g., $\tau_V = 10$. This defines a region in which the objective is expected to be defined and bounded below. (A similar condition is used in [71].) Murray and Prieto [57] show that under certain conditions, convergence can be assured if the line search enforces (2.5). If the objective is bounded below in $\mathbb{R}^n$, then b may be any large positive vector.

If α_k is essentially zero (because $\|p_k\|$ is very large), the objective is considered "unbounded" in the expanded region. Elastic mode is entered (or continued) as described in section 4.7.

2.8. The approximate Hessian. As suggested by Powell [64], we maintain a positive-definite approximate Hessian H_k. On completion of the line search, let the change in x and the gradient of the modified Lagrangian be

$$\delta_k = x_{k+1} - x_k \quad \text{and} \quad y_k = \nabla\mathcal{L}(x_{k+1}, x_k, \pi) - \nabla\mathcal{L}(x_k, x_k, \pi),$$

for some vector π. An estimate of the curvature of the modified Lagrangian along δ_k is incorporated using the BFGS quasi-Newton update,

$$H_{k+1} = H_k + \theta_k y_k y_k^T - \phi_k q_k q_k^T,$$

where $q_k = H_k\delta_k$, $\theta_k = 1/y_k^T\delta_k$, and $\phi_k = 1/q_k^T\delta_k$. When H_k is positive-definite, H_{k+1} is positive-definite if and only if the approximate curvature $y_k^T\delta_k$ is positive. The consequences of a negative or small value of $y_k^T\delta_k$ are discussed in the next section.

There are several choices for π, including the QP multipliers $\widehat{\pi}_k$ and least-squares multipliers λ_k (see, e.g., [40]). Here we use the updated multipliers π_{k+1} from the line search, because they are responsive to short steps in the search and are available at no cost. The definition of $\mathcal{L}$ from (2.2) yields

$$\begin{aligned} y_k &= \nabla\mathcal{L}(x_{k+1}, x_k, \pi_{k+1}) - \nabla\mathcal{L}(x_k, x_k, \pi_{k+1}) \\ &= g_{k+1} - g_k - (J_{k+1} - J_k)^T\pi_{k+1}. \end{aligned}$$

2.9. Maintaining positive-definiteness. Since the Hessian of the modified Lagrangian need not be positive-definite at a local minimizer, the approximate curvature $y_k^T\delta_k$ can be negative or very small at points arbitrarily close to (x^*, π^*). The curvature is considered not sufficiently positive if

$$y_k^T\delta_k < \sigma_k, \qquad \sigma_k = \alpha_k(1-\eta)p_k^T H_k p_k, \tag{2.6}$$

where η is a preassigned constant $(0 < \eta < 1)$ and p_k is the search direction $\widehat{x}_k - x_k$ defined by the QP subproblem. In such cases, if there are nonlinear constraints, two attempts are made to modify the update: the first modifying δ_k and y_k, the second modifying only y_k. If neither modification provides sufficiently positive approximate curvature, no update is made.

First modification. The purpose of this modification is to exploit the properties of the reduced Hessian at a local minimizer of (GNP). We define a new point z_k and evaluate the nonlinear functions there to obtain new values for δ_k and y_k:

$$\delta_k = x_{k+1} - z_k, \qquad y_k = \nabla\mathcal{L}(x_{k+1}, x_k, \pi_{k+1}) - \nabla\mathcal{L}(z_k, x_k, \pi_{k+1}).$$

We choose z_k by recording $\bar{x}_k$, the first *feasible* iterate found for problem (GQP$_k$) (see section 4). The search direction may be regarded as

$$p_k = (\bar{x}_k - x_k) + (\widehat{x}_k - \bar{x}_k) \equiv p_R + p_N.$$

We set $z_k = x_k + \alpha_k p_R$, giving $\delta_k = \alpha_k p_N$ and

$$y_k^T\delta_k = \alpha_k y_k^T p_N \approx \alpha_k^2 p_N^T\nabla^2\mathcal{L}(x_k, x_k, \pi_k)p_N,$$

so that $y_k^T\delta_k$ approximates the curvature along p_N. If W_k, the final working set of problem (GQP$_k$), is also the working set at $\bar{x}_k$, then $W_k p_N = 0$, and it follows that

$y_k^T\delta_k$ approximates the curvature for the reduced Hessian, which must be positive semidefinite at a minimizer of (GNP).

The assumption that the QP working set does not change once z_k is known is always justified for problems with equality constraints. (See Byrd and Nocedal [18] for a similar scheme in this context.) With inequality constraints, we observe that $W_k p_N \approx 0$, particularly during later major iterations, when the working set has settled down.

This modification exploits the fact that SNOPT maintains feasibility with respect to any linear constraints in (GNP). Although an additional function evaluation is required at z_k, we have observed that even when the Hessian of the Lagrangian has negative eigenvalues at a solution, the modification is rarely needed more than a few times if used in conjunction with the augmented Lagrangian modification discussed next.

Second modification. If (x_k, π_k) is not close to (x^*, π^*), the modified approximate curvature $y_k^T\delta_k$ may not be sufficiently positive, and a second modification may be necessary. We choose Δy_k so that $(y_k + \Delta y_k)^T\delta_k = \sigma_k$ (if possible) and redefine y_k as $y_k + \Delta y_k$. This approach was suggested by Powell [65], who proposed redefining y_k as a linear combination of y_k and $H_k\delta_k$.

To obtain Δy_k, we consider the *augmented* modified Lagrangian [59]:

$$\mathcal{L}_A(x, x_k, \pi_k) = f(x) - \pi_k^T d_L(x, x_k) + \tfrac{1}{2} d_L(x, x_k)^T \Omega d_L(x, x_k), \tag{2.7}$$

where Ω is a matrix of parameters to be determined: $\Omega = \operatorname{diag}(\omega_i)$, $\omega_i \geq 0$, $i = 1{:}m$. The perturbation

$$\Delta y_k = (J_{k+1} - J_k)^T \Omega d_L(x_{k+1}, x_k)$$

is equivalent to redefining the gradient difference as

$$y_k = \nabla\mathcal{L}_A(x_{k+1}, x_k, \pi_{k+1}) - \nabla\mathcal{L}_A(x_k, x_k, \pi_{k+1}). \tag{2.8}$$

We choose the smallest (minimum two-norm) ω_i's that increase $y_k^T\delta_k$ to σ_k (see (2.6)). They are determined by the linearly constrained least-squares problem

$$\boxed{\begin{array}{ll} \text{(LSP)} & \underset{\omega}{\text{minimize}} \quad \|\omega\|^2 \\ & \text{subject to} \quad a^T\omega = \beta, \quad \omega \geq 0, \end{array}}$$

where $\beta = \sigma_k - y_k^T\delta_k$ and $a_i = v_i w_i$ $(i = 1{:}m)$, with $v = (J_{k+1} - J_k)\delta_k$ and $w = d_L(x_{k+1}, x_k)$. The optimal ω can be computed analytically [44, 28]. If no solution exists, or if $\|\omega\|$ is very large, no update is made.

The approach just described is related to the idea of updating an approximation of the Hessian of the augmented Lagrangian, as suggested by Han [50] and Tapia [74]. However, we emphasize that the second modification is not required in the neighborhood of a solution, because as $x \to x^*$, $\nabla^2\mathcal{L}_A$ converges to $\nabla^2\mathcal{L}$, and the first modification will already have been successful.

2.10. Convergence tests. A point (x, π) is regarded as a satisfactory solution if it satisfies the first-order optimality conditions (2.1) to within certain tolerances. Let τ_P and τ_D be specified small positive constants, and define $\tau_x = \tau_P(1 + \|x\|)$, $\tau_\pi = \tau_D(1 + \|\pi\|)$. The SQP algorithm terminates if

$$c_i(x) \geq -\tau_x, \quad \pi_i \geq -\tau_\pi, \quad c_i(x)\pi_i \leq \tau_\pi, \quad |d_j| \leq \tau_\pi, \tag{2.9}$$

where $d = g(x) - J(x)^T\pi$. These conditions cannot be satisfied if (GNP) is infeasible, but in that case the SQP algorithm will eventually enter elastic mode and satisfy analogous tests for a series of problems

$$\boxed{\begin{array}{ll} (\mathrm{GNP}(\gamma)) & \underset{x,v}{\text{minimize}} \quad f(x) + \gamma e^T v \\ & \text{subject to} \quad c(x) + v \geq 0, \quad v \geq 0, \end{array}}$$

with γ taking an increasing set of values $\{\gamma_\ell\}$ up to some maximum. The optimality conditions for (GNP(γ)) include

$$0 \leq \pi_i \leq \gamma, \quad (c_i(x) + v_i)\pi_i = 0, \quad v_i(\gamma - \pi_i) = 0.$$

The fact that $\|\pi^*\|_\infty \leq \gamma$ at a solution of (GNP(γ)) leads us to initiate elastic mode if $\|\pi_k\|$ exceeds some value γ_1 (or if (GQP$_k$) is infeasible). We use

$$\gamma_1 \equiv \gamma_0 \|g(x_{k1})\|, \qquad \gamma_\ell = 10^{\ell(\ell-1)/2}\gamma_1 \quad (\ell = 2, 3, \dots), \tag{2.10}$$

where γ_0 is a parameter (10^4 in our numerical results) and x_{k1} is the iterate at which γ is first needed.

3. Large-scale Hessians. In the large-scale case, we cannot treat H_k as an $n \times n$ dense matrix. Nor can we maintain dense triangular factors of a transformed Hessian $Q^T H_k Q = R^T R$ as in NPSOL. We discuss the alternatives implemented in SNOPT.

3.1. Linear variables. If only some of the variables occur nonlinearly in the objective and constraint functions, the Hessian of the Lagrangian has structure that can be exploited during the optimization. We assume that the nonlinear variables are the first $\bar{n}$ components of x. By induction, if H_0 is zero in its last $n - \bar{n}$ rows and columns, the last $n - \bar{n}$ components of the BFGS update vectors y_k and $H_k\delta_k$ are zero for all k, and every H_k has the form

$$H_k = \begin{pmatrix} \bar{H}_k & 0 \\ 0 & 0 \end{pmatrix}, \tag{3.1}$$

where $\bar{H}_k$ is $\bar{n} \times \bar{n}$. Simple modifications of the methods of section 2.9 can be used to keep $\bar{H}_k$ positive-definite. A QP subproblem with a Hessian of this form is either unbounded or has at least $n - \bar{n}$ constraints in the final working set. This implies that the reduced Hessian need never have dimension greater than $\bar{n}$.

Under the assumption that the objective function is bounded below in some expanded feasible region $c(x) \geq -b$ (see (2.5)), a sequence of positive-definite matrices $\bar{H}_k$ with uniformly bounded condition numbers is sufficient for the SQP convergence theory to hold. (This case is analogous to converting inequality constraints to equalities by adding slack variables—the Hessian is singular only in the space of the slack variables.) However, in order to treat semidefinite Hessians such as (3.1), the QP solver must include an *inertia controlling* working-set strategy, which ensures that the reduced Hessian has at most one zero eigenvalue. See sections 4.6–4.7.

3.2. Dense Hessians. The Hessian approximations $\bar{H}_k$ are matrices of order $\bar{n}$, the number of nonlinear variables. If $\bar{n}$ is not too large, it is efficient to treat each $\bar{H}_k$ as a dense matrix and apply the BFGS updates explicitly. The storage requirement is fixed, and the number of major iterations should prove to be moderate. (We can expect one-step Q-superlinear convergence.)

3.3. Limited-memory Hessians. To treat problems where the number of nonlinear variables $\bar{n}$ is very large, we use a limited-memory procedure to update an initial Hessian approximation H_r a limited number of times. The present implementation is quite simple and has an advantage in the SQP context when the constraints are linear: the reduced Hessian for the QP subproblem can be updated between major iterations (see section 5.4).

Initially, suppose $\bar{n} = n$. Let ℓ be preassigned (say $\ell = 20$), and let r and k denote two major iterations such that $r \le k \le r + \ell$. Up to ℓ updates to a positive-definite H_r are accumulated to represent the Hessian as

$$H_k = H_r + \sum_{j=r}^{k-1} \big(\theta_j y_j y_j^T - \phi_j q_j q_j^T\big), \tag{3.2}$$

where $q_j = H_j \delta_j$, $\theta_j = 1/y_j^T \delta_j$, and $\phi_j = 1/q_j^T \delta_j$. The quantities $(y_j, q_j, \theta_j, \phi_j)$ are stored for each j. During major iteration k, the QP solver accesses H_k by requesting products of the form $H_k v$. These are computed with work proportional to $k - r$:

$$H_k v = H_r v + \sum_{j=r}^{k-1} \big(\theta_j (y_j^T v) y_j - \phi_j (q_j^T v) q_j\big).$$

On completion of iteration $k = r + \ell$, the diagonals of H_k are computed from (3.2) and saved to form the next positive-definite H_r (with $r = k + 1$). Storage is then "reset" by discarding the previous updates. (Similar schemes are described by Buckley and LeNir [13, 14] and Gilbert and Lemaréchal [37]. More elaborate schemes are given by Liu and Nocedal [54], Byrd, Nocedal, and Schnabel [19], and Gill and Leonard [39], and some have been evaluated by Morales [55]. However, as already indicated, these schemes would require refactorization of the reduced Hessian in the linearly constrained case.)

If $\bar{n} < n$, H_k has the form (3.1), and the same procedure is applied to $\bar{H}_k$. Note that the vectors y_j and q_j have length $\bar{n}$—a benefit when $\bar{n} \ll n$. The modified Lagrangian $\mathcal{L}_A$ from (2.7) retains this property for the modified y_k in (2.8).

4. The QP solver SQOPT. Since SNOPT solves nonlinear programs of the form (NP), it requires solution of QP subproblems of the same form, with $f(x)$ replaced by a convex quadratic function, and $c(x)$ replaced by its current linearization:

$$\begin{array}{lll} (\mathrm{QP}_k) & \underset{x}{\text{minimize}} & f_k + g_k^T (x - x_k) + \tfrac{1}{2}(x - x_k)^T H_k (x - x_k) \\ & \text{subject to} & l \le \begin{pmatrix} x \\ c_k + J_k (x - x_k) \\ Ax \end{pmatrix} \le u. \end{array}$$

At present, (QP_k) is solved by the package SQOPT [42], which employs a two-phase active-set algorithm and implements elastic programming implicitly when necessary. The Hessian H_k may be positive-semidefinite and is defined by a routine for forming products $H_k v$.

4.1. Elastic bounds. SQOPT can treat any of the bounds in (QP_k) as elastic. Let x_j refer to the jth variable or slack. For each j, an input array specifies which of the bounds l_j, u_j is elastic (either, neither, or both). A parallel array maintains

the current state of each x_j. If the variable or slack is currently outside its bounds by more than the `Minor feasibility tolerance`, it is given a linear penalty term $\gamma \times$ *infeasibility* in the objective function. This is a much-simplified but useful form of piecewise linear programming (Fourer [33, 34, 35]).

SNOPT uses elastic bounds in three different ways:

- to solve problem (FLP) (section 1.2) if the linear constraints are infeasible,
- to solve problem (PP1) (section 5.1),
- to solve the QP subproblems associated with problem (NP(γ)) after nonlinear elastic mode is initiated.

4.2. The null-space method. SQOPT maintains a dense Cholesky factorization of the QP reduced Hessian:

$$Z^T H_k Z = R^T R, \tag{4.1}$$

where Z is the null-space matrix for the working sets W in the QP minor iterations. Normally, R is computed from (4.1) when the nonelastic constraints are first satisfied. It is then updated as the QP working set changes. For efficiency the dimension of R should not be excessive (say, $n_Z \leq 2000$). This is guaranteed if the number of nonlinear variables is moderate (because $n_Z \leq \bar{n}$ at a solution), but it is often true even if $\bar{n} = n$.

To review notation, Z is maintained in "reduced-gradient" form as in MINOS, using the package LUSOL [45] to maintain sparse LU factors of a square matrix B whose columns change as the working set W changes:

$$W = \begin{pmatrix} B & S & N \\ & & I \end{pmatrix} P, \qquad Z = P^T \begin{pmatrix} -B^{-1}S \\ I \\ 0 \end{pmatrix}, \tag{4.2}$$

where P is a permutation such that B is nonsingular. Variables associated with B and S are called basic and superbasic; the remainder are called nonbasic. The number of degrees of freedom is the number of superbasic variables (the column dimension of S). Products of the form Zv and $Z^T g$ are obtained by solving with B or B^T.

4.3. Threshold pivoting (TPP and TCP). Stability in LU factorization is achieved by bounding the off-diagonal elements of L or U. There are many ways to do this, especially in the sparse case. In LUSOL, L has unit diagonals, and each elimination step produces the next column of L and the next row of U. Let

$$\begin{aligned} \tau_L &= \text{the } LU \textit{ factor tolerance} \text{ such that } |L_{ij}| \leq \tau_L \\ &\quad (\text{where } 1 < \tau_L \leq 100, \text{ say}), \\ A_l &= \text{the remaining submatrix to be factored after } l \text{ steps} \\ &\quad (\text{updated by the first } l \text{ columns of } L). \end{aligned}$$

For most factorizations, LUSOL uses a *threshold partial pivoting* strategy (TPP) similar to that in LA05 [67] and MA28 [27]. To become the next diagonal of U, a nonzero in A_l must be sufficiently large compared to *other nonzeros in the same column of* A_l.

With $\tau_L \in [4, 25]$, TPP usually performs well in terms of balancing stability and sparsity, but is not especially good at rank-detection (revealing near-singularity and its cause). For example, a triangular matrix A gives $L = I$ and $U = A$ for all values of τ_L (a perfect L and maximum sparsity, but little hint of possible ill-conditioning).

For greater reliability, a *threshold complete pivoting* strategy (TCP) has been implemented recently in LUSOL [63], in which the next diagonal of U must be reasonably large compared to *all nonzeros in* A_l. The original aim was to improve rank-detection for the sparse matrices arising during the optimization of Markov decision chains [62]. Although reduced sparsity and speed are expected, TCP has proved valuable within SNOPT, as described below.

In general we use TPP where possible, with τ_L decreasing through a short sequence of values (currently 4, 2, $\sqrt{2}$, ... , 1.1) if various tests continue to indicate instability (e.g., large $\|b - Bx\|$ or $\|x\|$ when basic variables are recomputed from $Bx = b$). When necessary, a switch is made to TCP with another sequence of values (currently $\tau_L = 20$, 10, 5, 2.5, $\sqrt{2.5}$, ... , 1.1).

4.4. Basis repair (square or singular case). Whenever a basis is factored, LUSOL signals "singularity" if any diagonals of U are judged small, and indicates which unit vectors (corresponding to slack variables) should replace the associated columns of B. The modified B is then factored.

The process may need to be repeated if the factors of B are not sufficiently "rank-revealing." Extreme behavior of this kind was exhibited by one of the CUTE problems (section 6.2) when the first basis was factored with the normal partial pivoting options. Problem *drcavty2* is a large square system of nonlinear equations (10000 constraints and variables, 140000 Jacobian nonzeros). The first TPP factorization with $\tau_L = 4.0$ indicated 243 singularities. After slacks were inserted, the next factorization indicated 47 additional singularities, the next a further 25, then 18, 14, 10, and so on. Nearly 30 TPP factorizations and 460 new slacks were required before the basis was regarded as suitably nonsingular. Since L and U each had about a million nonzeros in all factorizations, the repeated failures were rather expensive.

In contrast, a single TCP factorization with $\tau_L = 2.5$ indicated 100 singularities, after which the modified B proved to be very well-conditioned. Although L and U were more dense (1.35 million nonzeros each) and much more expensive to compute, the subsequent optimization required significantly fewer major and minor iterations.

For such reasons, SQOPT includes a special "BR factorization" for estimating the rank of a given B, using the LUSOL options shown in Figure 1. P and Q are the row and column permutations that make L unit triangular and U upper triangular, with small elements in the bottom right if B is close to singular. To save storage, the factors are discarded as they are computed. A normal "B factorization" then follows.

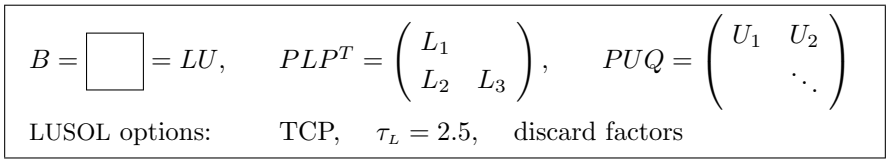


FIG. 1. *BR factorization (rank detection for square B).*

BR factorization is the primary recourse when unexpected growth occurs in $\|x\|$ following solution of $Bx = b$. It has proved valuable for some other CUTE problems arising from partial differential equations (namely, *porous1*, *porous2*, *bratu2d*, and *bratu3d*). A regular "marching pattern" is sometimes present in B, particularly in the first *triangular* basis following a cold start. With partial pivoting, the factors

display no small diagonals in U, yet the BR factors reveal a large number of dependent columns. Thus, although condition estimators are known that could tell us "this B is ill-conditioned" (e.g., [52]), we are using LUSOL's complete pivoting option to decide *which columns* are causing the poor condition.

4.5. Basis repair (rectangular case). When superbasic variables are present, the permutation P in (4.2) clearly affects the condition of B and Z. SQOPT therefore applies an occasional rectangular "BS factorization" to choose a new P, using the options shown in Figure 2.

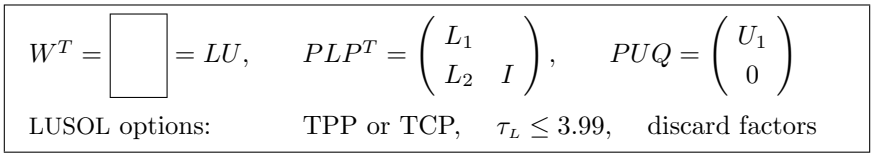


FIG. 2. *BS factorization (basis detection for rectangular W).*

For simplicity we assume that there are no nonbasic columns in W. A basis partition is given by

$$PW^T \equiv \begin{pmatrix} B^T \\ S^T \end{pmatrix} = \begin{pmatrix} L_1 \\ L_2 \end{pmatrix} U_1 Q^T,$$

and the required null-space matrix satisfying $WZ = 0$ is

$$Z \equiv P^T \begin{pmatrix} -B^{-1}S \\ I \end{pmatrix} = P^T \begin{pmatrix} -L_1^{-T} L_2^T \\ I \end{pmatrix}. \tag{4.3}$$

With $\tau_L \leq 3.99$, L and L_1 are likely to be well-conditioned, and $\zeta \equiv \|L_1^{-T} L_2^T\|$ is unlikely to be large. (It can be bounded by a polynomial function of τ_L.) The extreme singular values of Z are $\sigma_{\min} \geq 1$ and $\sigma_{\max} \approx 1 + \zeta$. It follows that Z should be well-conditioned *regardless of the condition of W*.

SQOPT applies this basis repair at the beginning of a warm start (when a potential B-S ordering is known). To prevent basis repair at *every* warm start—i.e., every major iteration of SNOPT—a normal $B = LU$ factorization is computed first with the current (usually larger) tolerance τ_L. If U appears to be more ill-conditioned than after the last repair, a new repair is invoked. The relevant test on the diagonals of U is tightened gradually to ensure that basis repair occurs periodically (even during a single major iteration if a QP subproblem requires many iterations).

Although the rectangular factors are discarded, we see from (4.3) that a normal factorization of B allows iterations to proceed with an equivalent Z. (A BR factorization may be needed to repair B first if W happens to be singular.)

4.6. Inertia control. If (NP) contains linear variables, H_k in (3.1) is positive semidefinite. In SQOPT, only the last diagonal of R (see (4.1)) is allowed to be zero. (See [46] for discussion of a similar strategy for indefinite QP.) If the initial R is singular, enough temporary constraints are added to the working set to give a nonsingular R. Thereafter, R can become singular only when a constraint is deleted from the working set (in which case no further constraints are deleted until R becomes

nonsingular). When R is singular at a nonoptimal point, it is used to define a direction d_Z such that

$$Z^T H_k Z d_Z = 0 \quad \text{and} \quad g^T Z d_Z < 0, \tag{4.4}$$

where $g = g(x_k) + H_k(x - x_k)$ is the gradient of the quadratic objective. The vector $d = Zd_Z$ is a direction of unbounded descent for the QP in the sense that the QP objective is linear and decreases without bound along d. Normally, a step along d reaches a new constraint, which is then added to the working set for the next iteration.

4.7. Unbounded QP subproblems. If the QP objective is unbounded along d, subproblem (QP_k) terminates. The final QP search direction $d = Zd_Z$ is also a direction of unbounded descent for the objective of (NP). To show this, we observe from (4.4) that if we choose $p = d$, then

$$H_k p = 0 \quad \text{and} \quad g_k^T p < 0.$$

The imposed nonsingularity of $\bar{H}_k$ (see (3.1)) implies that the nonlinear components of p are zero, and so the nonlinear terms of the objective and constraint functions are unaltered by steps of the form $x_k + \alpha p$. Since $g_k^T p < 0$, the objective of (NP) is unbounded along p, because it must include a term in the linear variables that decreases without bound along p.

In short, (NP) behaves like an unbounded linear program (LP) along p, with the nonlinear variables (and functions) frozen at their current values. Thus if x_k is feasible for (NP), unboundedness in (QP_k) implies that the objective $f(x)$ is unbounded for feasible points, and the problem is declared unbounded.

If x_k is infeasible, unboundedness in (QP_k) implies that $f(x)$ is unbounded for some expanded feasible region $c(x) \geq -b$ (see (2.5)). We enter or continue elastic mode (with an increased value of γ if it has not already reached its maximum permitted value). Eventually the QP subproblem will be bounded, or x_k will become feasible, or the iterations will converge to a point that approximately minimizes the one-norm of the constraint violations.

5. Algorithmic details. A practical SQP algorithm requires many features to achieve reliability and efficiency. We discuss some more of them here before summarizing the main algorithmic steps.

5.1. The initial point. To take advantage of a good starting point x_0, we apply SQOPT to one of the "proximal-point" problems

(PP1)	$\underset{x}{\text{minimize}} \quad \|\bar{x} - \bar{x}_0\|_1$
	subject to the linear constraints and bounds

or

(PP2)	$\underset{x}{\text{minimize}} \quad \|\bar{x} - \bar{x}_0\|_2^2$
	subject to the linear constraints and bounds,

where $\bar{x}$ and $\bar{x}_0$ correspond to the nonlinear variables in x and x_0. The solution defines a new starting point x_0 for the SQP iteration. The nonlinear functions are evaluated at this point, and a "crash" procedure is executed to find a working set W_0 for the linearized constraints.

In practice we prefer problem (PP1), as it is linear and can use SQOPT's implicit elastic bounds. (We temporarily set the bounds on the nonlinear variables to be $\bar{x}_0 \leq \bar{x} \leq \bar{x}_0$.) Note that problem (PP2) may be "more nonlinear" than the original problem (NP), in the sense that its exact solution may lie on fewer constraints (even though it is nonlinear in the same subset of variables, $\bar{x}$). To prevent the reduced Hessian from becoming excessively large with this option, we terminate SQOPT early by specifying a loose optimality tolerance.

5.2. Undefined functions. If the constraints in (PP1) or (PP2) prove to be infeasible, SNOPT solves problem (FLP) (see section 1.2) and terminates without computing the nonlinear functions. The problem was probably formulated incorrectly.

Otherwise, the linear constraints and bounds define a certain "linear feasible region" $\mathcal{R}_L$, and all iterates satisfy $x_k \in \mathcal{R}_L$ to within a feasibility tolerance (as with NPSOL). Although SQP algorithms might converge more rapidly sometimes if all constraints were treated equally, the aim is to help prevent function evaluations at obvious singularities.

In practice, the functions may not be defined everywhere within $\mathcal{R}_L$, and it may be an unbounded region. Hence, the function routines are permitted to return an "undefined function" signal. If the signal is received from the *first* function call (before any line search takes place), SNOPT terminates. Otherwise, the line search backtracks and tries again.

5.3. Early termination of QP subproblems. SQP theory usually assumes that the QP subproblems are solved to optimality. For large problems with a poor starting point and $H_0 = I$, many thousands of iterations may be needed for the first QP, building up many degrees of freedom (superbasic variables) that are promptly eliminated by more thousands of iterations in the second QP.

In general, it seems wasteful to expend much effort on any QP before updating H_k and the constraint linearization. Murray and Prieto [57] suggest one approach to terminating the QP solutions early, requiring that at least one QP stationary point be reached. The associated theory implies that any subsequent point $\widehat{x}_k$ generated by a QP solver is suitable, provided that $\|\widehat{x}_k - x_k\|$ is nonzero. In SNOPT we have implemented a method within this framework that has proved effective on many problems. Conceptually we could perform the following steps:

- Fix many variables at their current value.
- Perform one SQP major iteration on the reduced problem (solving a smaller QP to get a search direction for the nonfixed variables).
- Free the fixed variables, and complete the major iteration with a "full" search direction that happens to leave many variables unaltered.
- Repeat.

Normal merit-function theory should guarantee progress at each stage on the associated reduced *nonlinear* problem. We are simply suboptimizing.

In practice, we are not sure which variables to fix at each stage, the reduced QP could be infeasible, and degeneracy could produce a zero search direction. Instead, the choice of which variables to fix is made within the QP solver. The following steps are performed:

- Perform QP iterations on the full problem until a feasible point is found or elastic mode is entered.
- Continue iterating until certain limits are reached and not all steps have been degenerate.
- Freeze nonbasic variables that have not yet moved.

- Solve the reduced QP to optimality.

Rather arbitrary limits may be employed and perhaps combined. We have implemented the following as user options:

- `Minor iterations limit` (default 500) suggests termination if a reasonable number of QP iterations have been performed (beyond the first feasible point).
- `New superbasics limit` (default 99) suggests termination if the number of free variables has increased significantly (since the first feasible point).
- `Minor optimality tolerance` (default 10^{-6}) specifies an optimality tolerance for the final QPs.

Internally, SNOPT sets a loose but decreasing optimality tolerance for the early QPs (somewhat smaller than a measure of the current primal-dual infeasibility for (NP)). This "loose tolerance" strategy provides a dynamic balance between major and minor iterations in the manner of inexact Newton methods (Dembo, Eisenstat, and Steihaug [23]).

5.4. Linearly constrained problems. For problems with linear constraints only, the maximum step length is not necessarily one. Instead, it is the maximum feasible step along the search direction. If the line search is not restricted by the maximum step, the line search ensures that the approximate curvature is sufficiently positive and the BFGS update can always be applied. Otherwise, the update is skipped if the approximate curvature is not sufficiently positive.

For linear constraints, the working-set matrix W_k does not change at the new major iterate x_{k+1}, and the basis B need not be refactorized. If B is constant, then so is Z, and the only change to the reduced Hessian between major iterations comes from the rank-two BFGS update. This implies that the reduced Hessian need not be refactorized if the BFGS update is applied explicitly to the reduced Hessian. This obviates factorizing the reduced Hessian at the start of each QP, saving considerable computation.

Given *any* nonsingular matrix Q, the BFGS update to H_k implies the following update to $Q^T H_k Q$:

$$\bar{H}_Q = H_Q + \theta_k y_Q y_Q^T - \phi_k q_Q q_Q^T, \tag{5.1}$$

where $\bar{H}_Q = Q^T H_{k+1} Q$, $H_Q = Q^T H_k Q$, $y_Q = Q^T y_k$, $\delta_Q = Q^{-1}\delta_k$, $q_Q = H_Q \delta_Q$, $\theta_k = 1/y_Q^T \delta_Q$, and $\phi_k = 1/q_Q^T \delta_Q$. If Q is of the form $(\, Z \quad Y \,)$ for some matrix Y, the reduced Hessian is the leading principal submatrix of H_Q.

The Cholesky factor R of the reduced Hessian is simply the upper-left corner of the $\bar{n} \times n$ upper-trapezoidal matrix R_Q such that $H_Q = R_Q^T R_Q$. The update for R is derived from the rank-one update to R_Q implied by (5.1). Given δ_k and y_k, if we had all of the Cholesky factor R_Q, it could be updated directly as

$$R_Q + uv^T, \qquad w = R_Q \delta_Q, \quad u = w/\|w\|, \quad v = \sqrt{\theta_k}\, y_Q - R_Q^T u$$

(see Goldfarb [49], Dennis and Schnabel [24]). This rank-one modification of R_Q could be restored to upper-triangular form by applying two sequences of plane rotations from the left [38].

The same sequences of rotations can be generated even though not all of R_Q is present. Let v_Z be the first n_Z elements of v. The following algorithm determines the Cholesky factor $\bar{R}$ of the first n_Z rows and columns of $\bar{H}_Q$ from (5.1):

1. Compute $q = H_k \delta_k$ and $t = Z^T q$.
2. Define $\phi = \|w\|_2 = (\delta_k^T H_k \delta_k)^{1/2} = (q^T \delta_k)^{1/2}$.

3. Solve $R^T w_z = t$.
4. Define $u_z = w_z/\phi$ and $\sigma = (1 - \|u_z\|_2^2)^{1/2}$.
5. Apply a backward sweep of n_z rotations P_1 in the planes (n_z+1, i), $i = n_z{:}1$, to give an upper triangular $\widehat{R}$ and a "row spike" r^T:

$$P_1 \begin{pmatrix} R & u_z \\ & \sigma \end{pmatrix} = \begin{pmatrix} \widehat{R} & 0 \\ r^T & 1 \end{pmatrix}.$$

6. Apply a forward sweep of n_z rotations P_2 in the planes (i, n_z+1), $i = 1{:}n_z+1$, to restore the upper triangular form:

$$P_2 \begin{pmatrix} \widehat{R} \\ r^T + v_z^T \end{pmatrix} = \begin{pmatrix} \bar{R} \\ 0 \end{pmatrix}.$$

5.5. Summary of the SQP algorithm. The main steps of the SNOPT algorithm follow. We assume that a starting point $(x_0,\ \pi_0)$ is available, and that the reduced-Hessian QP solver SQOPT is being used. We describe elastic mode qualitatively. Specific values for γ are given in section 2.10.

0. Apply the QP solver to problem (PP1) or (PP2) to find a point close to x_0 satisfying the linear constraints. If the PP problem is infeasible, declare problem (NP) infeasible. Otherwise, a working-set matrix W_0 is returned. Set $k = 0$, and evaluate functions and gradients at x_0.
1. Factorize W_k.
2. Find $\bar{x}_k$, a feasible point for the QP subproblem. (This is an intermediate point for the QP solver, which also provides a working-set matrix $\bar{W}_k$ and its null-space matrix $\bar{Z}_k$.) If no feasible point exists, initiate elastic mode and restart the QP.
3. Form the reduced Hessian $\bar{Z}_k^T H_k \bar{Z}_k$, and compute its Cholesky factorization.
4. Continue solving the QP subproblem to find $(\widehat{x}_k, \widehat{\pi}_k)$, an optimal QP solution. (This provides a working-set matrix $\widehat{W}_k$ and its null-space matrix $\widehat{Z}_k$.)
   If elastic mode has not been initiated but $\|\widehat{\pi}_k\|_\infty$ is "large," enter elastic mode and restart the QP.
   If the QP is unbounded and x_k satisfies the nonlinear constraints, declare the problem unbounded (f is unbounded below in the feasible region). Otherwise (if the QP is unbounded), go to Step 6 (f is unbounded below in the feasible region if a feasible point exists).
5. If (x_k, π_k) satisfies the convergence tests for (NP) analogous to (2.9), declare the solution optimal. If similar convergence tests are satisfied for (NP(γ)), go to Step 6. Otherwise, go to Step 7.
6. If elastic mode has not been initiated, enter elastic mode and repeat Step 4. Otherwise, if γ has not reached its maximum value, increase γ and repeat Step 4. Otherwise, declare the problem infeasible.
7. Find a step length α_k that gives a sufficient reduction in the merit function. Set $x_{k+1} = x_k + \alpha_k(\widehat{x}_k - x_k)$ and $\pi_{k+1} = \pi_k + \alpha_k(\widehat{\pi}_k - \pi_k)$. In the process, evaluate functions and gradients at x_{k+1}.
8. Define $\delta_k = x_{k+1} - x_k$ and $y_k = \nabla\mathcal{L}(x_{k+1}, x_k, \pi_{k+1}) - \nabla\mathcal{L}(x_k, x_k, \pi_{k+1})$. If $y_k^T\delta_k < \sigma_k$ (see (2.6)), recompute δ_k and y_k, with x_k redefined as $x_k + \alpha_k(\bar{x}_k - x_k)$. (This requires an extra evaluation of the problem derivatives.) If necessary, increase $y_k^T\delta_k$ (if possible) by adding an augmented Lagrangian term to y_k.

9. If $y_k^T\delta_k \geq \sigma_k$, apply the BFGS update to H_k, using the pair $(H_k\delta_k, y_k)$.
10. Define W_{k+1} from $\widehat{W}_k$, set $k \leftarrow k+1$, and repeat from Step 1.

Apart from the function and gradient evaluations, most of the computational effort lies in Steps 1 and 3. Steps 2 and 4 may also involve significant work if the QP subproblem requires many minor iterations. Typically this will happen only during the early major iterations.

6. Numerical results. SNOPT and SQOPT implement all of the techniques described in sections 2–4. The Fortran 77 coding is compatible with Fortran 90 and 95 compilers and permits recursive calls, or re-entrant calls in a multithreaded environment, as well as translation into C via $f2c$ [29] (though these features are not used here).

We give the results of applying SNOPT 6.1 of May, 2001, to problems in the CUTE and COPS 2.0 test collections [10, 7, 25]. Function and gradient values were used throughout (but not second derivatives).

All runs were made on an SGI Octane workstation with 512MB of RAM and two 250MHz R10000 processors (only one being used for each problem solution). The f90 compiler was used with `-n32 -O` options specifying 32-bit addressing and full code optimization. The floating-point precision was 2.22×10^{-16}. Table 1 defines the notation used in the tables of results.

TABLE 1
Notation in tables of results.

n_Z	The number of degrees of freedom at a solution (columns in Z).
Mnr	The number of QP minor iterations.
Mjr	The number of major iterations required by the optimizer.
Fcn	The number of function and gradient evaluations.
cpu	The number of cpu seconds.
Obj	The final objective value (to help classify local solutions).
Con	The final constraint violation norm (to identify infeasible problems).
a	Almost optimal (within 10^{-2} of satisfying the convergence test).
c	Final nonoptimal point could not be improved.
s	User-defined superbasics limit exceeded.

6.1. Parameters for SNOPT. Figure 3 gives the SNOPT optional parameters used, most of which are default values. Linear constraints and variables are scaled (`Scale option 1`), and the first basis is essentially triangular (`Crash option 3`).

`Elastic weight` sets $\gamma_0 = 10^4$ in (2.10).

The `Major feasibility` and `optimality` tolerances set τ_P and τ_D in section 2.10 for problem (NP). The `Minor` tolerances are analogous parameters for SQOPT as it solves (QP_k). The `Minor feasibility tolerance` incidentally applies to the bound and linear constraints in (NP) as well as (QP_k).

`Violation limit` sets τ_V in section 2.7 to define an expanded feasible region in which the objective is expected to be bounded below.

For the Hessian approximations H_k, if the number of nonlinear variables is small enough ($\bar{n} \leq 75$), a full dense BFGS Hessian is used. Otherwise, a limited-memory BFGS Hessian is used, with H_k reset to the current Hessian diagonal every 20 major iterations.

6.2. Results on the CUTE test set. The CUTE distribution of 01/May/2001 contains 945 problems in standard interface format (SIF). A list of the CUTE problem

types and their frequency is given in Table 2. Although many problems allow for the number of variables and constraints to be adjusted in the SIF file, our tests used the dimensions set in the CUTE distribution. This gave problems ranging in size from *hs1*, with two variables and no constraints, to *cont5-qp*, with 40601 variables and 40201 constraints.

```
BEGIN SNOPT Problem
   Minimize
   Crash option                      3
   Derivative level                  3
   Elastic weight                1.0E+4
   Hessian updates                  20
   Superbasics limit              2000
   Iterations                    90000
   Major iterations               2000
   Minor iterations                500
   LU partial pivoting
   Major feasibility tolerance   1.0E-6
   Major optimality  tolerance   2.0E-6
   Minor feasibility tolerance   1.0E-6
   Minor optimality  tolerance   1.0E-6
   New superbasics                  99
   Line search tolerance           0.9
   Proximal point method             1
   Scale option                      1
   Step limit                      2.0
   Unbounded objective          1.0E+15
   Verify level                     -1
   Violation limit               1.0E+6
END SNOPT Problem
```

FIG. 3. *The SNOPT optional parameter file.*

TABLE 2
The 945 CUTE problems listed by type and frequency.

Frequency	Type	Characteristics
24	LP	Linear obj, linear constraints
116	QP	Quadratic obj, linear constraints
160	UC	Nonlinear obj, no constraints
125	BC	Nonlinear obj, bound constraints
70	LC	Nonlinear obj, linear constraints
375	NC	Nonlinear obj, nonlinear constraints
75	FP	No objective

From the complete set of 945 problems, 74 were omitted as follows:

- 6 nonsmooth problems (*bigbank*, *gridgena*, *hs87*, *net1*, *net2* and *net3*),
- 57 problems with more than 2000 degrees of freedom at the solution (*aug3d*, *aug3dc*, *aug3dcqp*, *dixmaanb*, *dtoc5*, *dtoc6*, *jannson3*, *jannson4*, *jimack*, *jnlbrng1*, *jnlbrng2*, *jnlbrnga*, *minsurfo*, *obstclae*, *obstclbm*, *odnamur*, *orthrdm2*, *orthrgdm*, *stcqp1*, *stnqp1*, *torsion6*, and the 36 *lukvli* and *lukvle1* problems),
- 9 problems with undefined variables or floating-point exceptions in the SIF file (*himmelbj*, *lhaifam*, *lin*, *pfit1*, *pfit3*, *recipe*, *robotarm*, *s365mod*, and *scon1dls*),

- 2 problems too large to decode (*qpband* and *qpnband*),
- 1 problem with excessively low accuracy in the objective gradients (*bleachng*). Requesting greater accuracy leads to excessive evaluation time.

SNOPT was applied to the remaining 870 problems, using the options listed in Figure 3. No special information was used in the case of LP, QP, and FP problems—i.e., each problem was assumed to have a general nonlinear objective. The results are summarized in Table 3.

TABLE 3
Summary: SNOPT on the smooth CUTE problems.

Problems attempted	870
Optimal	794
Unbounded	3
Infeasible	10
Optimal, low accuracy	11
Cannot be improved	7
False infeasibility	17
Terminated	28
Major iterations	108980
Minor iterations	678524
Function evaluations	153867
Cpu time (secs)	70864.7

Discussion. Problems *flosp2hh*, *flosp2hl*, *flosp2hm*, *ktmodel*, and *model* have infeasible linear constraints, but were included anyway. The objectives for *indef*, *mesh*, and *static3* are unbounded below in the feasible region. SNOPT correctly diagnosed the special features of these problems.

A total of 11 problems (*allinitc*, *eigmaxc*, *eigminc*, *hs268*, *mancino*, *marine*, *orthrds2*, *orthregd*, *penalty3*, *pinene*, and *s268*) were terminated at a point that satisfied either the feasibility *or* the optimality test and was within 10^{-2} of satisfying the other test. AMPL implementations of *marine* and *pinene* were solved successfully as part of the COPS 2.0 collection (see section 6.3).

SNOPT reported 22 problems (*argauss*, *bratu2dt*, *cont6-qq*, *drcavty2*, *eigenb*, *eigmaxb*, *fletcher*, *flosp2th*, *growth*, *hadamard*, *heart6*, *himmelbd*, *hs90*, *junkturn*, *lewispol*, *lootsma*, *lubrif*, *lubrifc*, *nystrom5*, *optcdeg3*, *powellsq*, *vanderm3*) with infeasible nonlinear constraints. Since SNOPT is not assured of finding a *global* minimizer of the sum of infeasibilities, failure to find a feasible point does not imply that none exists. Of these 22 problems, all but five cases must be counted as failures because they are known to have feasible points. The five exceptions, *flosp2th*, *junkturn*, *lewispol*, *lubrif*, and *nystrom5*, have no known feasible points. To gain further assurance that these problems are indeed infeasible, they were re-solved using SNOPT's `Feasible Point` option, in which the true objective is ignored but "elastic mode" is invoked (as usual) if the constraint linearizations prove to be infeasible (i.e., $f(x) = 0$ and $\gamma = 1$ in problem (NP(γ)) of section 1.1). In all five cases, the final sum of constraint violations was comparable to that obtained with the composite objective. We conjecture that these problems are infeasible.

Problems *fletcher* and *lootsma* have feasible solutions, but their initial points are infeasible and stationary for the sum of infeasibilities, and thus SNOPT terminated immediately. These problems are also listed as failures. Problem *drcavty2* is also

listed as a failure, although it is probably infeasible for the size of problem tested (196 variables, 101 general constraints). SNOPT ran successfully on the larger versions of the problem (the largest having 10816 variables and 10001 general constraints).

SNOPT was unable to solve 28 cases within the allotted 2000 major iterations (*biggsb1*, *bqpgauss*, *catena*, *chainwoo*, *chenhark*, *curly10*, *curly20*, *curly30*, *drcav1lq*, *drcav2lq*, *drcav3lq*, *eigenbls*, *eigencls*, *hydc20ls*, *noncvxu2*, *noncvxun*, *palmer5b*, *palmer5e*, *palmer7a*, *palmer7e*, *qr3dls*, *sbrybnd*, *scosine*, *scurly10*, *scurly20*, *scurly30*, *sparsine*, and *vibrbeam*). Another 7 problems could not be improved at a nonoptimal point: *brownbs*, *catena*, *glider*, *meyer3*, *nuffield*, *vanderm1*, and *vanderm2*. SNOPT essentially found the solution of the badly scaled problems *brownbs* and *meyer3* but was unable to declare optimality. An AMPL implementation of *glider* was solved successfully (see section 6.3)

If the infeasible LC problems, the unbounded problems, and the 5 (conjectured) infeasible problems are counted as successes, SNOPT solved a grand total of 807 of the 870 problems attempted. In another 11 cases, SNOPT found a point that was within a factor 10^{-2} of satisfying the convergence test. These results provide strong evidence of the robustness of first-derivative SQP methods when implemented with an augmented Lagrangian merit function and an elastic variable strategy for treating infeasibility.

6.3. Results on the COPS 2.0 test set. Tests on the 17 problems in the COPS 2.0 collection were made using the AMPL modeling system [36]. When necessary, the AMPL model and data files were modified to increase the problem size to be the largest considered in [7] (see Table 4).

TABLE 4
Dimensions of the AMPL versions of the COPS problems.

No.	Problem	Type	Variables	Constraints		
				Linear	Nonlinear	Total
1	*bearing*	BC	5000	0	0	0
2	*camshape*	NC	800	800	801	1601
3	*catmix*	NC	2401	1	1600	1601
4	*chain*	NC	800	401	1	402
5	*channel*	FP	3198	1598	1600	3198
6	*elec*	NC	600	1	200	201
7	*gasoil*	NC	4001	799	3200	3999
8	*glider*	NC	1999	1	1600	1601
9	*marine*	NC	4815	1593	3200	4793
10	*methanol*	NC	4802	1198	3600	4798
11	*minsurf*	BC	5000	0	0	0
12	*pinene*	NC	4000	996	3000	3996
13	*polygon*	NC	198	99	4950	5048
14	*robot*	NC	3599	2	2400	2402
15	*rocket*	NC	1601	0	1200	1201
16	*steering*	NC	2000	2	1600	1602
17	*torsion*	BC	5000	0	0	0

The bound constrained problems *bearing*, *minsurf*, and *torsion* have more than 2000 degrees of freedom at the solution, but were tested anyway. (SNOPT is not appropriate for problems with only bound constraints unless many of the bounds are active.) Table 5 gives results obtained by applying SNOPT with the options listed in Figure 3. The default AMPL options (including problem preprocessing) were used in each case.

TABLE 5
SNOPT on the COPS 2.0 problems.

No.	Problem	Mnr	Mjr	Fcn	Obj	Con	n_Z	cpu
1	*bearing*^{s}	2279	19	23	1.147002E+01	0.0E+00	2000	175.0
2	*camshape*	3019	9	18	4.222963E+00	9.4E-08	6	5.5
3	*catmix*	594	11	14	-4.796022E-02	2.8E-07	395	14.0
4	*chain*	839	40	44	5.068630E+00	4.2E-06	399	24.6
5	*channel*	2192	5	7	1.000000E+00	3.2E-05	0	18.8
6	*elec*	731	326	354	1.843890E+04	4.6E-10	400	194.9
7	*gasoil*	2607	21	25	5.236596E-03	7.2E-08	3	32.5
8	*glider*	33959	516	785	1.247974E+03	5.3E-09	359	891.7
9	*marine*	5437	71	132	1.974653E+07	1.1E-11	22	144.7
10	*methanol*	6250	1381	8280	9.022290E-03	9.0E-10	4	1170.2
11	*minsurf*^{s}	3029	19	26	2.516317E+00	0.0E+00	2000	1251.5
12	*pinene*	3090	41	63	1.987216E+01	4.0E-13	5	51.0
13	*polygon*	3490	64	66	7.850233E-01	1.1E-08	98	51.0
14	*robot*	5855	28	51	9.141018E+00	2.1E-06	0	279.3
15	*rocket*	2663	8	16	1.005422E+00	1.3E-07	66	16.7
16	*steering*	764	29	35	5.545734E-01	7.6E-07	398	30.2
17	*torsion*^{s}	3112	16	20	-4.004933E-01	0.0E+00	2000	171.4

Discussion. SNOPT solved every COPS problem that has fewer than 2000 degrees of freedom at the solution. The default `New superbasics` limit (99) often improves efficiency, but for *bearing*, *minsurf*, and *torsion*, a larger value would reduce the time and major iterations needed to terminate with excess superbasics.

It is not clear why the AMPL formulations of *glider* and *robot* (problem *robotarm* in the CUTE set) can be solved relatively easily, but not the CUTE versions. Repeating the runs with AMPL option `presolve 0` did not significantly increase the cpu time, which implies that preprocessing is not the reason for the difference in performance.

The COPS problems were also used to investigate the effect of the number ℓ of limited-memory updates on the performance of SNOPT. Table 6 gives times for the 14 nonlinearly constrained problems when solved with different choices for ℓ. In the case of the BC problems *bearing*, *minsurf*, and *torsion*, the principal effect of increasing ℓ is to increase the cost of the Hessian/vector products in the minor iterations needed to expand the reduced Hessian to its maximum size.

The results are typical of the performance of SNOPT in practical situations.

- Small values of ℓ can give low computation times but may adversely affect robustness on more challenging problems. For example, $\ell = 5$ gave the one run in which the AMPL formulation of *glider* could not be solved.
- As ℓ is increased, the number of major iterations tends to decrease. However, numerical performance remains relatively stable. (For example, the same local solution was always found for the highly nonlinear problem *polygon*.)
- As ℓ is increased, the solution time often decreases initially, but then increases as the cost of the products $H_k v$ increases. This would be reflected in the total computation time for Table 6 if it were not for *methanol*, whose time improves dramatically because of a better Hessian approximation.

The choice of default value $\ell = 20$ is intended to provide robustness without a significant computational penalty.

7. Extensions. Where possible, we have defined the SQP algorithm to be independent of the QP solver. Of course, implicit elastic bounds and certain "warm start" features are highly desirable. For example, SQOPT can use a given starting point and

TABLE 6
Number of LM updates vs. cpu time.

Problem	Limited-memory updates					
	5	10	15	20	25	30
camshape	5.6	5.4	5.6	5.5	5.4	5.4
catmix	7.7	14.3	14.5	14.0	15.2	15.0
chain	14.2	13.2	18.8	24.6	17.6	22.5
channel	19.1	18.8	19.0	18.8	19.0	18.7
elec	221.3	216.3	127.3	194.9	217.6	241.0
gasoil	34.1	32.8	32.0	32.5	32.2	31.8
glider	254.0[c]	845.5	429.2	891.7	369.6	595.0
marine	155.2	139.3	157.3[a]	144.7	163.4[a]	166.6
methanol	398.2	390.5	1218.2	1170.2	1253.0	501.1
pinene	42.4[a]	48.7	50.2	51.0	43.8	45.6
polygon	120.3	74.8	87.3	51.0	56.5	63.0
robot	215.7	248.4	275.9	279.3	277.2	274.9
rocket	16.4	16.2	16.5	16.7	15.8	16.0
steering	43.8	26.0	27.6	30.2	31.4	30.7
	1548.2	2090.2	2479.4	2924.7	2517.7	2027.4

TABLE 7
Number of LM updates vs. major iterations.

Problem	Limited-memory updates					
	5	10	15	20	25	30
camshape	9	9	9	9	9	9
catmix	7	11	11	11	11	11
chain	29	25	33	40	27	32
channel	5	5	5	5	5	5
elec	459	399	227	326	340	361
gasoil	26	23	20	21	21	21
glider	50[c]	513	224	516	173	275
marine	83	71	90[a]	71	84[a]	88
methanol	479	245	1224	1381	1469	604
pinene	30[a]	37	39	41	29	30
polygon	243	123	158	64	81	94
robot	22	23	28	28	28	28
rocket	8	8	8	8	8	8
steering	38	28	28	29	26	26
	1488	1520	2104	2550	2311	1592

working set, and for linearly constrained problems (section 5.4) it can accept a known Cholesky factor R for the reduced Hessian.

Here we discuss other "black-box" QP solvers that could be used in future implementations of SNOPT. Recall that active-set methods solve KKT systems of the form

$$\begin{pmatrix} H_k & W^T \\ W & \end{pmatrix} \begin{pmatrix} p \\ q \end{pmatrix} = \begin{pmatrix} g \\ h \end{pmatrix} \tag{7.1}$$

at each minor iteration, where W is the current working-set matrix. Reduced-Hessian methods such as SQOPT are efficient if W is nearly square and products $H_k x$ can be formed efficiently, but our aim is to accommodate many degrees of freedom.

7.1. Approximate reduced Hessians. As the major iterations converge, the QP subproblems require fewer changes to their working set, and with warm starts they eventually solve in one minor iteration. Hence, the work required by SQOPT becomes dominated by the computation of the reduced Hessian $Z^T H_k Z$ and its factor R from (4.1), especially if there are many degrees of freedom.

For such cases, MINOS could be useful as the QP solver because it has two ways of *approximating* the reduced Hessian in the form $Z^T H_k Z \approx R^T R$:

- R may be input from the previous major iteration and maintained using quasi-Newton updates during the QP minor iterations.
- If R is very large, it is maintained in the form

$$R = \begin{pmatrix} R_r & 0 \\ & D \end{pmatrix},$$

where R_r is a dense triangle of specified size and D is diagonal. This structure partitions the superbasic variables into two sets. After a few minor iterations involving all superbasics (with quasi-Newton updates to R_r and D), the variables associated with D are temporarily frozen. Iterations proceed with updates to R_r only, and superlinear convergence can be expected within that subspace. A frozen superbasic variable is then interchanged with one from R_r, and the process is repeated.

Both of these features could be implemented in a future version of SQOPT. Thus, SNOPT with MINOS or an enhanced SQOPT as the QP solver would provide a viable SQP algorithm for optimization problems of arbitrary dimension. The cost per minor iteration is controllable, and the only unpredictable quantity is the total number of minor iterations.

Note that the SQP updates to H_k could be applied to R between major iterations as for the linear-constraint case (section 5.4). However, the quasi-Newton updates during the first few minor iterations of each QP should achieve a similar effect.

7.2. Range-space methods. If all variables appear nonlinearly, H_k is positive-definite. A "range-space" approach could then be used to solve systems (7.1) as W changes. This amounts to maintaining factors of H_k's Schur complement, $S = W H_k^{-1} W^T$. It would be efficient if W did not have many rows, so that S could be treated as a dense matrix.

7.3. Schur-complement methods. For limited-memory Hessians of the form $H_k = H_0 + V D V^T$, where H_0 is some convenient Hessian approximation, $D = \operatorname{diag}(I, -I) = D^{-1}$, and V contains the BFGS update vectors, equation (7.1) is equivalent to

$$\begin{pmatrix} H_0 & W^T & V \\ W & & \\ V^T & & -D \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} = \begin{pmatrix} g \\ h \\ 0 \end{pmatrix}.$$

Following [43, section 3.6.2], if we define

$$K_0 = \begin{pmatrix} H_0 & W^T \\ W & \end{pmatrix},$$

it would be efficient to work with a sparse factorization of K_0 and dense factors of its Schur complement S. (For a given QP subproblem, V is constant, but changes to W would be handled by appropriate updates to S.)

This approach has been explored by Betts and Frank [2, section 5] with $H_0 = I$ (or possibly a sparse finite-difference Hessian approximation). As part of an SQP algorithm, its practical success depends greatly on the definition of H_0 and on the BFGS updates that define V. Our experience with SNOPT emphasizes the importance of updating H_k even in the presence of negative curvature; hence the precautions of section 2.9.

If H_0 were defined as in section 3, the major iterates would be identical to those currently obtained with SQOPT.

8. Summary and conclusions. We have presented theoretical and practical details about an SQP algorithm for solving nonlinear programs with large numbers of constraints and variables, where the nonlinear functions are smooth and first derivatives are available.

As with interior-point methods, the most promising way to achieve efficiency with the linear algebra is to work with sparse second derivatives (i.e., an exact Hessian of the Lagrangian, or a sparse finite-difference approximation). However, indefinite QP subproblems raise many practical questions, and alternatives are needed when second derivatives are not available.

The present implementation, SNOPT, uses a positive-definite quasi-Newton Hessian approximation H_k. If the number of nonlinear variables is moderate, H_k is stored as a dense matrix. Otherwise, limited-memory BFGS updates are employed, with resets to the current diagonal at a specified frequency (typically every 20 major iterations). An augmented Lagrangian merit function (the same as in NPSOL) ensures convergence from arbitrary starting points.

The present QP solver, SQOPT, maintains a dense reduced-Hessian factorization $Z^T H_k Z = R^T R$, where Z is obtained from a sparse LU factorization of part of the Jacobian. Efficiency improves with the number of constraints active at a solution; i.e., the number of degrees of freedom n_z should not be excessive. For the numerical tests we set a limit of 2000. This is adequate for many problem classes, such as control problems when the number of control variables is not excessive.

The numerical results of section 6 show that SNOPT is effective on most of the problems in the CUTE and COPS 2.0 test sets. Separate comparisons with MINOS have shown greater reliability as a result of the merit function and the "elastic variables" treatment of infeasibility, and much greater efficiency when function evaluations are expensive. Reliability has also improved relative to NPSOL, and the sparse-matrix techniques have permitted production runs on increasingly large trajectory problems.

Future work will include the use of second derivatives (when available) and alternative QP solvers to allow for indefiniteness of the QP Hessian and many degrees of freedom.

Acknowledgments. We extend sincere thanks to our colleagues Dan Young and Rocky Nelson of the Boeing Company (formerly McDonnell Douglas Space Systems, Huntington Beach, CA) for their constant support and feedback during the development of SNOPT. We also appreciate many suggestions from the referees and Associate Editor Jorge Nocedal.

REFERENCES

[1] R. H. Bartels, *A penalty linear programming method using reduced-gradient basis-exchange techniques*, Linear Algebra Appl., 29 (1980), pp. 17–32.

[2] J. T. Betts and P. D. Frank, *A sparse nonlinear optimization algorithm*, J. Optim. Theory Appl., 82 (1994), pp. 519–541.
[3] L. T. Biegler, J. Nocedal, and C. Schmid, *A reduced Hessian method for large-scale constrained optimization*, SIAM J. Optim., 5 (1995), pp. 314–347.
[4] P. T. Boggs, A. J. Kearsley, and J. W. Tolle, *A global convergence analysis of an algorithm for large-scale nonlinear optimization problems*, SIAM J. Optim., 9 (1999), pp. 833–862.
[5] P. T. Boggs, A. J. Kearsley, and J. W. Tolle, *A practical algorithm for general large scale nonlinear optimization problems*, SIAM J. Optim., 9 (1999), pp. 755–778.
[6] P. T. Boggs and J. W. Tolle, *Sequential quadratic programming*, in Acta Numerica, Cambridge University Press, Cambridge, England, 1995, pp. 1–51.
[7] A. Bondarenko, D. Bortz, and J. J. Moré, *COPS: Large-Scale Nonlinearly Constrained Optimization Problems*, Technical report ANL/MCS-TM-237, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1998; revised, 1999.
[8] I. Bongartz, A. R. Conn, N. I. M. Gould, M. A. Saunders, and P. L. Toint, *A Numerical Comparison Between the LANCELOT and MINOS Packages for Large-Scale Constrained Optimization*, Report 97/13, Département de Mathématique, Facultés Universitaires de Namur, Namur, Belgium, 1997.
[9] I. Bongartz, A. R. Conn, N. I. M. Gould, M. A. Saunders, and P. L. Toint, *A Numerical Comparison Between the LANCELOT and MINOS Packages for Large-Scale Constrained Optimization: The Complete Numerical Results*, Report 97/14, Département de Mathématique, Facultés Universitaires de Namur, Namur, Belgium, 1997.
[10] I. Bongartz, A. R. Conn, N. I. M. Gould, and P. L. Toint, *CUTE: Constrained and unconstrained testing environment*, ACM Trans. Math. Software, 21 (1995), pp. 123–160.
[11] G. G. Brown and G. W. Graves, *Elastic Programming: A New Approach to Large-Scale Mixed-Integer Optimization*, presented at the ORSA/TIMS meeting, Las Vegas, NV, 1975.
[12] G. G. Brown and G. W. Graves, *The XS Mathematical Programming System*, working paper, Department of Operations Research, Naval Postgraduate School, Monterey, CA, 1975.
[13] A. Buckley and A. LeNir, *QN-like variable storage conjugate gradients*, Math. Programming, 27 (1983), pp. 155–175.
[14] A. Buckley and A. LeNir, *BBVSCG—A variable storage algorithm for function minimization*, ACM Trans. Math. Software, 11 (1985), pp. 103–119.
[15] R. H. Byrd, *Robust Trust-Region Methods for Contstrained Optimization*, presented at the SIAM Conference on Optimization, Houston, TX, 1987.
[16] R. H. Byrd, J. C. Gilbert, and J. Nocedal, *A trust region method based on interior point techniques for nonlinear programming*, Math. Program., 89 (2000), pp. 149–185.
[17] R. H. Byrd, M. E. Hribar, and J. Nocedal, *An interior point algorithm for large-scale nonlinear programming*, SIAM J. Optim., 9 (1999), pp. 877–900.
[18] R. H. Byrd and J. Nocedal, *An analysis of reduced Hessian methods for constrained optimization*, Math. Programming, 49 (1991), pp. 285–323.
[19] R. H. Byrd, J. Nocedal, and R. B. Schnabel, *Representations of quasi-Newton matrices and their use in limited-memory methods*, Math. Programming, 63 (1994), pp. 129–156.
[20] A. R. Conn, *Constrained optimization using a nondifferentiable penalty function*, SIAM J. Numer. Anal., 10 (1973), pp. 760–784.
[21] A. R. Conn, *Linear programming via a nondifferentiable penalty function*, SIAM J. Numer. Anal., 13 (1976), pp. 145–154.
[22] A. R. Conn, N. I. M. Gould, and P. L. Toint, *LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A)*, Lecture Notes in Comput. Math. 17, Springer-Verlag, Berlin, Heidelberg, New York, London, Paris, Tokyo, 1992.
[23] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.
[24] J. E. Dennis, Jr., and R. B. Schnabel, *A new derivation of symmetric positive definite secant updates*, in Nonlinear Programming, 4 (1980), Academic Press, New York, 1981, pp. 167–199.
[25] E. Dolan and J. J. Moré, *Benchmarking optimization software with COPS*, Math. Program., 91 (2002), pp. 201–213.
[26] A. Drud, *CONOPT: A GRG code for large sparse dynamic nonlinear optimization problems*, Math. Programming, 31 (1985), pp. 153–191.
[27] I. S. Duff, *MA28—A set of Fortran Subroutines for Sparse Unsymmetric Linear Equations*, Report AERE R8730, Atomic Energy Research Establishment, Harwell, England, 1977.
[28] S. K. Eldersveld, *Large-Scale Sequential Quadratic Programming Algorithms*, Ph.D. thesis, Department of Operations Research, Stanford University, Stanford, CA, 1991.
[29] S. I. Feldman, D. M. Gay, M. W. Maimone, and N. L. Schryer, *A Fortran-to-C Converter*,

Computing Science Technical report 149, AT&T Bell Laboratories, Murray Hill, NJ, 1990.
[30] R. Fletcher, *An ℓ_1 penalty method for nonlinear constraints*, in Numerical Optimization, P. T. Boggs, R. H. Byrd, and R. B. Schnabel, eds., SIAM, Philadelphia, 1985, pp. 26–40.
[31] R. Fletcher, *Optimization*, 2nd ed., John Wiley and Sons, Chichester, New York, Brisbane, Toronto, Singapore, 1987.
[32] R. Fletcher and S. Leyffer, *Nonlinear programming without a penalty function*, Math. Program., 91 (2002), pp. 239–269.
[33] R. Fourer, *A simplex algorithm for piecewise-linear programming.* I. *Derivation and proof*, Math. Programming, 33 (1985), pp. 204–233.
[34] R. Fourer, *A simplex algorithm for piecewise-linear programming.* II. *Finiteness, feasibility and degeneracy*, Math. Programming, 41 (1988), pp. 281–315.
[35] R. Fourer, *A simplex algorithm for piecewise-linear programming.* III. *Computational analysis and applications*, Math. Programming, 53 (1992), pp. 213–235.
[36] R. Fourer, D. M. Gay, and B. W. Kernighan, *AMPL: A Modeling Language for Mathematical Programming*, The Scientific Press, San Francisco, 1993.
[37] J. C. Gilbert and C. Lemaréchal, *Some numerical experiments with variable-storage quasi-Newton algorithms*, Math. Programming, (1989), pp. 407–435.
[38] P. E. Gill, G. H. Golub, W. Murray, and M. A. Saunders, *Methods for modifying matrix factorizations*, Math. Comput., 28 (1974), pp. 505–535.
[39] P. E. Gill and M. W. Leonard, *Limited-Memory Reduced-Hessian Methods for Unconstrained Optimization*, Numerical Analysis Report NA 97-1, University of California, San Diego, La Jolla, CA, 1997.
[40] P. E. Gill and W. Murray, *The computation of Lagrange multiplier estimates for constrained minimization*, Math. Programming, 17 (1979), pp. 32–60.
[41] P. E. Gill, W. Murray, and M. A. Saunders, *User's Guide for SNOPT* 5.3*: A Fortran Package for Large-Scale Nonlinear Programming*, Numerical Analysis Report 97-5, Department of Mathematics, University of California, San Diego, La Jolla, CA, 1997.
[42] P. E. Gill, W. Murray, and M. A. Saunders, *User's Guide for SQOPT* 5.3*: A Fortran Package for Large-Scale Linear and Quadratic Programming*, Numerical Analysis Report 97-4, Department of Mathematics, University of California, San Diego, La Jolla, CA, 1997.
[43] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *Sparse matrix methods in optimization*, SIAM J. Sci. Statist. Comput., 5 (1984), pp. 562–589.
[44] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *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, 1986.
[45] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *Maintaining LU factors of a general sparse matrix*, Linear Algebra Appl., 88/89 (1987), pp. 239–270.
[46] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *Inertia-controlling methods for general quadratic programming*, SIAM Rev., 33 (1991), pp. 1–36.
[47] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *Some theoretical properties of an augmented Lagrangian merit function*, in Advances in Optimization and Parallel Computing, P. M. Pardalos, ed., North–Holland, Amsterdam, 1992, pp. 101–128.
[48] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, London, New York, 1981.
[49] D. Goldfarb, *Factorized variable metric methods for unconstrained optimization*, Math. Comput., 30 (1976), pp. 796–811.
[50] S. P. Han, *Superlinearly convergent variable metric algorithms for general nonlinear programming problems*, Math. Programming, 11 (1976), pp. 263–282.
[51] C. R. Hargraves and S. W. Paris, *Direct trajectory optimization using nonlinear programming and collocation*, J. Guidance, Control, and Dynamics, 10 (1987), pp. 338–348.
[52] N. Higham, *FORTRAN codes for estimating the one-norm of a real or complex matrix, with applications to condition estimation*, ACM Trans. Math. Software, 14 (1988), pp. 381–396.
[53] M. Lalee, J. Nocedal, and T. Plantenga, *On the implementation of an algorithm for large-scale equality constrained optimization*, SIAM J. Optim., 8 (1998), pp. 682–706.
[54] D. C. Liu and J. Nocedal, *On the limited memory BFGS method for large scale optimization*, Math. Programming, 45 (1989), pp. 503–528.
[55] J. L. Morales, *A numerical study of limited memory BFGS methods*, Appl. Math. Lett., to appear.
[56] W. Murray, *Sequential quadratic programming methods for large-scale problems*, J. Comput. Optim. Appl., 7 (1997), pp. 127–142.
[57] W. Murray and F. J. Prieto, *A sequential quadratic programming algorithm using an incomplete solution of the subproblem*, SIAM J. Optim., 5 (1995), pp. 590–640.

[58] B. A. Murtagh and M. A. Saunders, *Large-scale linearly constrained optimization*, Math. Programming, 14 (1978), pp. 41–72.
[59] B. A. Murtagh and M. A. Saunders, *A projected Lagrangian algorithm and its implementation for sparse nonlinear constraints*, Math. Prog. Study, 16 (1982), pp. 84–117.
[60] B. A. Murtagh and M. A. Saunders, *MINOS* 5.4 *User's Guide*, Report SOL 83-20R, Department of Operations Research, Stanford University, Stanford, CA, 1995.
[61] E. O. Omojokun, *Trust region algorithms for nonlinear equality and inequality constraints*, Ph.D. thesis, Department of Computer Science, University of Colorado, Boulder, CO, 1989.
[62] M. O'Sullivan, *New Methods for Dynamic Programming over an Infinite Time Horizon*, Ph.D. thesis, Department of Management Science and Engineering, Stanford University, Stanford, CA, 2001.
[63] M. O'Sullivan and M. A. Saunders, *Sparse LU Factorization with Threshold Complete Pivoting*, SOL Report, Department of Management Science and Engineering, Stanford University, Stanford, CA, to appear.
[64] M. J. D. Powell, *Algorithms for nonlinear constraints that use Lagrangian functions*, Math. Programming, 14 (1978), pp. 224–248.
[65] M. J. D. Powell, *The convergence of variable metric methods for nonlinearly constrained optimization calculations*, in Nonlinear Programming, 3 (Proceedings of the Symposium, Special Interest Group Mathematical Programming, University of Wisconsin, Madison, WI, 1977), Academic Press, New York, 1978, pp. 27–63.
[66] M. J. D. Powell, *Variable metric methods for constrained optimization*, in Mathematical Programming: The State of the Art (Bonn, 1982), A. Bachem, M. Grötschel, and B. Korte, eds., Springer, Berlin, 1983, pp. 288–311.
[67] J. K. Reid, *Fortran Subroutines for Handling Sparse Linear Programming Bases*, Report AERE R8269, Atomic Energy Research Establishment, Harwell, England, 1976.
[68] S. M. Robinson, *A quadratically-convergent algorithm for general nonlinear programming problems*, Math. Programming, 3 (1972), pp. 145–156.
[69] R. W. H. Sargent and M. Ding, *A new SQP algorithm for large-scale nonlinear programming*, SIAM J. Optim., 11 (2000), pp. 716–747.
[70] K. Schittkowski, *NLPQL: A Fortran subroutine for solving constrained nonlinear programming problems*, Ann. Oper. Res., 11 (1985/1986), pp. 485–500.
[71] P. Spellucci, *Han's method without solving QP*, in Optimization and Optimal Control (Proceedings of the Conf. Math. Res. Inst., Oberwolfach, Oberwolfache-Walke, Germany, 1980), Springer, Berlin, 1981, pp. 123–141.
[72] P. Spellucci, *A new technique for inconsistent QP problems in the SQP method*, Math. Methods Oper. Res., 47 (1998), pp. 355–400.
[73] P. Spellucci, *An SQP method for general nonlinear programs using only equality constrained subproblems*, Math. Prog. Ser. A, 82 (1998), pp. 413–448.
[74] R. A. Tapia, *A stable approach to Newton's method for general mathematical programming problems in* $\mathbb{R}^n$, J. Optim. Theory Appl., 14 (1974), pp. 453–476.
[75] I.-B. Tjoa and L. T. Biegler, *Simultaneous solution and optimization strategies for parameter estimation of differential algebraic equation systems*, Ind. Eng. Chem. Res., 30 (1991), pp. 376–385.
[76] K. Tone, *Revisions of constraint approximations in the successive QP method for nonlinear programming problems*, Math. Programming, 26 (1983), pp. 144–152.
[77] G. Van der Hoek, *Asymptotic properties of reduction methods applying linearly equality constrained reduced problems*, Math. Prog. Study, 16 (1982), pp. 162–189.
[78] R. J. Vanderbei and D. F. Shanno, *An interior-point algorithm for nonconvex nonlinear programming*, Comput. Optim. Appl., 13 (1999), pp. 231–252.
[79] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, *Algorithm* 778*: L-BFGS-B—Fortran subroutines for large-scale bound constrained optimization*, ACM Trans. Math. Software, 23 (1997), pp. 550–560.

SIAM J. OPTIM.
Vol. 12, No. 4, pp. 1007–1031

LAGRANGIAN DUAL INTERIOR-POINT METHODS FOR SEMIDEFINITE PROGRAMS*

MITUHIRO FUKUDA†, MASAKAZU KOJIMA†, AND MASAYUKI SHIDA‡

Abstract. This paper proposes a new predictor-corrector interior-point method for a class of semidefinite programs, which numerically traces the central trajectory in a space of Lagrange multipliers. The distinguishing features of the method are full use of the BFGS quasi-Newton method in the corrector procedure and an application of the conjugate gradient method with an effective preconditioning matrix induced from the BFGS quasi-Newton method in the predictor procedure. Some preliminary numerical results are reported.

Key words. semidefinite program, primal-dual interior-point method, predictor-corrector method, Lagrangian dual, BFGS quasi-Newton method, conjugate gradient method

AMS subject classifications. 90C22, 90C51, 90C53, 65F10, 49N15, 49M29

PII. S1052623401387349

1. Introduction.

Consider the following equality standard form semidefinite program (SDP) and its dual:

$$\text{(1.1)}\quad \left\{ \begin{array}{lll} \text{Primal SDP:} & \text{maximize} & \boldsymbol{C} \bullet \boldsymbol{X} \\ & \text{subject to} & \boldsymbol{A}_p \bullet \boldsymbol{X} = a_p \ (p = 1, 2, \ldots, m),\ \boldsymbol{X} \in \mathbb{S}^n_+, \end{array} \right.$$

$$\text{(1.2)}\quad \left\{ \begin{array}{lll} \text{Dual SDP:} & \text{minimize} & \displaystyle\sum_{p=1}^{m} a_p y_p \\ & \text{subject to} & \displaystyle\sum_{p=1}^{m} \boldsymbol{A}_p y_p - \boldsymbol{S} = \boldsymbol{C},\ \boldsymbol{S} \in \mathbb{S}^n_+, \end{array} \right.$$

where $\mathbb{S}^n_+$ denotes the cone of positive semidefinite matrices in the space $\mathbb{S}^n$ of $n \times n$ real symmetric matrices; $\boldsymbol{C}, \boldsymbol{A}_p \in \mathbb{S}^n$ $(p = 1, 2, \ldots, m)$ are given matrices; $a_p \in \mathbb{R}$ $(p = 1, 2, \ldots, m)$ are given real numbers; and $\boldsymbol{A} \bullet \boldsymbol{X}$ is the inner product of $\boldsymbol{A} \in \mathbb{S}^n$ and $\boldsymbol{X} \in \mathbb{S}^n$ (i.e., $\boldsymbol{A} \bullet \boldsymbol{X} = \text{Trace } \boldsymbol{A}^T \boldsymbol{X} = \sum_{i=1}^n \sum_{j=1}^n A_{ij} X_{ij}$). We will also use the notation $\mathbb{R}_{++} \subseteq \mathbb{R}$ and $\mathbb{S}^n_{++} \subseteq \mathbb{S}^n_+$ for the set of positive numbers and the cone of positive definite symmetric matrices, respectively.

Primal-dual interior-point methods (see, e.g., [1, 14, 17, 19, 26]) for SDPs have been getting popular, and now several software packages (see, e.g., [3, 9, 25, 28]) based on them are available through the Internet. However, these software packages are not powerful enough to solve large scale general SDPs, e.g., SDPs with m and/or n larger than several thousand. Serious difficulties arise when we solve the key linear equation $\boldsymbol{M}\boldsymbol{dy} = \boldsymbol{r}$ with a fully dense $m \times m$ matrix $\boldsymbol{M}$, which is often called the Schur complement equation, to obtain a search direction $(\boldsymbol{dX}, \boldsymbol{dy}, \boldsymbol{dS}) \in \mathbb{S}^n \times \mathbb{R}^m \times \mathbb{S}^n$. As m gets larger, solving the equation $\boldsymbol{M}\boldsymbol{dy} = \boldsymbol{r}$ by direct methods such as the

*Received by the editors April 4, 2001; accepted for publication (in revised form) November 9, 2001; published electronically April 19, 2002.
http://www.siam.org/journals/siopt/12-4/38734.html

†Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, 2-12-1 Oh-Okayama, Meguro-ku, Tokyo 152-8552, Japan (mituhiro@is.titech.ac.jp, kojima@is.titech.ac.jp). The first author was supported by The Ministry of Education, Culture, Sports, Science and Technology of Japan.

‡Department of Mathematics, National Defense Academy of Japan, 1-10-20 Hashirimizu, Yokosuka, Kanagawa 239-8686, Japan (shida@cc.nda.ac.jp).

Cholesky factorization also becomes more expensive and more difficult. When m is larger than ten thousand, it is impossible even to store the entire coefficient matrix $\boldsymbol{M}$ in standard workstations; hence we are forced to use iterative methods such as the CG (conjugate gradient) method and the CR (conjugate residual) method to solve the equation $\boldsymbol{M}d\boldsymbol{y} = \boldsymbol{r}$. However, the condition number of the coefficient matrix $\boldsymbol{M}$ gets worse rapidly as the iterate $(\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{S})$ gets closer to an optimal solution in the primal-dual space. Under such circumstances, we need to improve the condition number by applying an appropriate preconditioning to $\boldsymbol{M}$ because, otherwise, even a low accuracy solution of $\boldsymbol{M}d\boldsymbol{y} = \boldsymbol{r}$ would require more and more CG or CR iterations. One of the current important issues is how to obtain an effective preconditioning for the coefficient matrix $\boldsymbol{M}$ without the need to store $\boldsymbol{M}$. See [6, 18, 22, 27, 33, 34] for more details on applications of iterative methods, preconditioning, and numerical experiments on some large scale SDPs.

Another inefficiency of using primal-dual interior-point methods is that the $n \times n$ primal matrix variable $\boldsymbol{X}$ is fully dense in general, even when all the data matrices $\boldsymbol{C}, \boldsymbol{A}_p \in \mathbb{S}^n$ $(p = 1, 2, \ldots, m)$ are sparse. This is a disadvantage of primal-dual interior-point methods compared to the dual scaling method [2], which generates iterates only in the dual space; note that the dual matrix variable $\boldsymbol{S} = \sum_{p=1}^m \boldsymbol{A}_p y_p - \boldsymbol{C}$ inherits the sparsity of the data matrices $\boldsymbol{C}, \boldsymbol{A}_p \in \mathbb{S}^n$ $(p = 1, 2, \ldots, m)$. To overcome this disadvantage, Fukuda et al. [10] and Nakata et al. [21] recently proposed methods based on the positive definite matrix completion for exploiting the aggregate sparsity pattern over the data matrices. Besides interior-point methods, some other computational methods have also been proposed and intensively studied for solving large scale SDPs: the spectral bundle method [13] and nonlinear programming reformulations of SDPs [4, 5, 29].

Numerical results on large scale SDPs have been reported. These include (i) SDP relaxations of the max-cut problem and the graph bisection problem solved by the spectral bundle method [12, 13], the dual-scaling method [2], and a nonlinear programming reformulation [4] and (ii) an SDP relaxation of the max clique problem solved by the primal-dual interior-point method with the use of the CG method [22] and the CR method [27]. However, thus far successful numerical results on large scale SDPs have been restricted to a few types of such SDPs (arising from SDP relaxation of combinatorial optimization problems on graphs) that do not require highly accurate solutions.

Aiming to resolve many of the difficulties mentioned so far, this paper proposes a new computational method, a Lagrangian dual predictor-corrector path-following interior-point method (abbreviated as LDIPM) for solving a class of SDPs. We may regard the LDIPM as a variant of the simple dual predictor-corrector path-following interior-point method (abbreviated as DIPM) with the use of the standard dual logarithmic barrier function $\tilde{g}(\cdot; \mu) : \mathcal{Y}_{++} \to \mathbb{R}$ $(\mu \in \mathbb{R}_{++})$ defined by

$$\tilde{g}(\boldsymbol{y}; \mu) = \sum_{p=1}^m a_p y_p - \mu \log\det\left(\sum_{p=1}^m \boldsymbol{A}_p y_p - \boldsymbol{C}\right) \quad \text{for every } (\boldsymbol{y}, \mu) \in \mathcal{Y}_{++} \times \mathbb{R}_{++}, \tag{1.3}$$

where $\mathcal{Y}_{++} \equiv \{\boldsymbol{y} \in \mathbb{R}^m : \sum_{p=1}^m \boldsymbol{A}_p y_p - \boldsymbol{C} \in \mathbb{S}^n_{++}\}$. In the DIPM, we replace the dual SDP (1.2) by a family of strictly convex minimization problems

$$\text{minimize } \tilde{g}(\boldsymbol{y}; \mu) \text{ subject to } \boldsymbol{y} \in \mathcal{Y}_{++} \qquad (\mu \in \mathbb{R}_{++}). \tag{1.4}$$

The DIPM is a predictor-corrector method that numerically traces the central trajectory $\{(\boldsymbol{y}(\mu), \mu) : \mu \in \mathbb{R}_{++}\}$ in the dual space, where $\boldsymbol{y}(\mu)$ corresponds to the unique minimizer of problem (1.4). Although the idea of the DIPM is rather classical and stemmed from the SUMT (sequential unconstrained minimization technique) of Fiacco–McCormick [8], it provides us with some significant features to overcome the difficulties involved in the primal-dual interior-point method for SDPs. We will outline below the DIPM together with its major advantages and disadvantages.

Let $\bar{\mu} \in \mathbb{R}_{++}$ be fixed. The role of the corrector procedure is to approximate the minimizer $\boldsymbol{y}(\bar{\mu})$ of the function $\tilde{g}(\cdot; \bar{\mu})$ over $\mathcal{Y}_{++}$, starting from a $\hat{\boldsymbol{y}} \in \mathcal{Y}_{++}$. Here we assume that the point $(\hat{\boldsymbol{y}}, \bar{\mu}) \in \mathcal{Y}_{++} \times \mathbb{R}_{++}$ has been generated by the predictor procedure of the previous iteration or given initially in the first iteration. When we generate a sequence $\hat{\boldsymbol{y}} = \boldsymbol{y}^0, \boldsymbol{y}^1, \ldots, \boldsymbol{y}^k, \ldots \in \mathcal{Y}_{++}$ in the corrector procedure,

(a) we can fully utilize various unconstrained optimization methods such as quasi-Newton methods with the use of first derivatives.

In connection with our predictor procedure, it is convenient for us to use the BFGS quasi-Newton method, which updates an approximation $\boldsymbol{H}^k$ of the inverse of the Hessian matrix $\boldsymbol{\nabla}^2 \tilde{g}(\boldsymbol{y}^k; \bar{\mu})$ of the function $\tilde{g}(\cdot; \bar{\mu})$ at $\boldsymbol{y} = \boldsymbol{y}^k$.

Throughout the iterations of the DIPM,

(b) the variables $(\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{S}) \in \mathbb{S}^n_{++} \times \mathbb{R}^m \times \mathbb{S}^n_{++}$ of the primal-dual pair of SDPs (1.1) and (1.2) are evaluated only where $(\boldsymbol{y}, \boldsymbol{S})$ is an interior feasible solution of (1.2), and the relation $\boldsymbol{X}\boldsymbol{S} = \mu \boldsymbol{I}$ holds for some $\mu \in \mathbb{R}_{++}$.

This implies that the dual matrix variable $\boldsymbol{S}$ inherits the sparsity of the data matrices $\boldsymbol{C}, \boldsymbol{A}_p$ $(p = 1, 2, \ldots, m)$ as in the existing primal-dual interior-point methods, and also that the inverse $\boldsymbol{X}^{-1} = \boldsymbol{S}/\mu$ of the primal matrix variable $\boldsymbol{X}$ shares exactly the same sparsity with $\boldsymbol{S}$. Furthermore, we can explicitly avoid computing and maintaining the primal matrix variable $\boldsymbol{X}$ if we compute and maintain a (sparse) Cholesky factorization of $\boldsymbol{S}$. It is noteworthy that even the positive definite matrix completion [10, 21] is unnecessary for retrieving the primal matrix variable.

Now, suppose that the rth iterate $\boldsymbol{y}^r$ of the corrector procedure attains an approximation of the minimizer $\boldsymbol{y}(\bar{\mu})$ of $\tilde{g}(\cdot; \bar{\mu})$ over $\mathcal{Y}_{++}$. Then the point $(\boldsymbol{y}^r, \bar{\mu})$ lies approximately on the trajectory $\{(\boldsymbol{y}(\mu), \mu) : \mu \in \mathbb{R}_{++}\}$ or $\boldsymbol{y}^r \approx \boldsymbol{y}(\bar{\mu})$. We then perform the predictor procedure based on the first-order approximation $\boldsymbol{y}(\mu^+) \approx \boldsymbol{y}(\bar{\mu}) + (\mu^+ - \bar{\mu})\dot{\boldsymbol{y}}(\bar{\mu})$ of a point $(\boldsymbol{y}(\mu^+), \mu^+)$, for some positive μ^+ less than the current $\bar{\mu}$, on the trajectory $\{(\boldsymbol{y}(\mu), \mu) : \mu \in \mathbb{R}_{++}\}$. Here $\dot{\boldsymbol{y}}(\bar{\mu})$ denotes the first derivative of $\boldsymbol{y}(\mu)$ evaluated at $\mu = \bar{\mu}$. In order to compute the derivative $\dot{\boldsymbol{y}}(\bar{\mu})$, we apply the CG method to a system of linear equations with coefficient matrix equal to the Hessian matrix $\boldsymbol{\nabla}^2 \tilde{g}(\boldsymbol{y}(\bar{\mu}); \bar{\mu})$ of $\tilde{g}(\cdot; \bar{\mu})$ at $\boldsymbol{y} = \boldsymbol{y}(\bar{\mu})$. It should be emphasized here that

(c) the BFGS quasi-Newton matrix $\boldsymbol{H}^r$ used in the previous corrector procedure serves as a powerful preconditioning matrix.

This preconditioning technique is essential to making our predictor procedure more effective.

In spite of the nice features (a), (b), and (c) mentioned above,

(d) there is a major worry that as $\boldsymbol{y}(\mu)$ approaches to the boundary of $\mathcal{Y}_{++}$ we may encounter numerical difficulties in approximating the minimizer $\boldsymbol{y}(\mu)$ since the condition number of the Hessian matrix $\boldsymbol{\nabla}^2 \tilde{g}(\cdot; \mu)$ gets worse rapidly.

To offset this disadvantage, we will use a "logarithmic barrier function" defined on the entire m-dimensional space $\mathbb{R}^m$ without any boundary in our LDIPM. We are

concerned with the following SDP and its dual:

$$\text{(1.5)}\qquad \begin{cases} \text{Primal SDP:} & \text{maximize} & \boldsymbol{C} \bullet \boldsymbol{X} \\ & \text{subject to} & \boldsymbol{A}_p \bullet \boldsymbol{X} = a_p \ (p = 1, 2, \ldots, m), \\ & & \boldsymbol{I} \bullet \boldsymbol{X} = b, \ \boldsymbol{X} \in \mathbb{S}^n_+, \end{cases}$$

$$\text{(1.6)}\qquad \begin{cases} \text{Dual SDP:} & \text{minimize} & \displaystyle\sum_{p=1}^{m} a_p y_p + bw \\ & \text{subject to} & \displaystyle\sum_{p=1}^{m} \boldsymbol{A}_p y_p + \boldsymbol{I} w - \boldsymbol{S} = \boldsymbol{C}, \ \boldsymbol{S} \in \mathbb{S}^n_+. \end{cases}$$

Here $\boldsymbol{I}$ denotes the $n \times n$ identity matrix, and b a positive number. Throughout the paper we assume the following.

CONDITION 1.1.

(i) *There is an interior feasible solution* $\boldsymbol{X}^0$ *of* (1.5)*, i.e., an* $\boldsymbol{X}^0 \in \mathbb{S}^n_{++}$ *which satisfies the constraints of* (1.5)*.*

(ii) *The data matrices* $\boldsymbol{A}_p$ $(p = 1, 2, \ldots, m)$ *and* $\boldsymbol{I}$*, which appear in the equality constraints of* (1.5)*, are linearly independent.*

Note that the primal SDP (1.5) involves a "simplex constraint" $\boldsymbol{X} \in \Omega_+ \equiv \{\boldsymbol{X} \in \mathbb{S}^n_+ : \boldsymbol{I} \bullet \boldsymbol{X} = b\}$. Although this constraint is restrictive, the primal SDP (1.5) covers various important SDPs such as SDP relaxations of combinatorial optimization problems. We also note that if the feasible region of a given SDP (1.1) without the simplex constraint is bounded and a bound is known in advance, then we can transform it into the primal SDP (1.5) above. For any given Lagrangian multiplier (dual variable) vector $\boldsymbol{y} \in \mathbb{R}^m$, we can easily find a $(w, \boldsymbol{S}) \in \mathbb{R} \times \mathbb{S}^n_{++}$ such that $(\boldsymbol{y}, w, \boldsymbol{S}) \in \mathbb{R}^{m+1} \times \mathbb{S}^n_{++}$ becomes an interior feasible solution of the dual SDP (1.6). This is another important feature of the above primal-dual pair of SDPs (1.5) and (1.6). Helmberg and Rendl [13] dealt with this type of SDP, for which they presented their spectral bundle method.

Based on the Lagrangian duality theory, we will construct a family of strictly convex smooth functions $g(\cdot; \mu) : \mathbb{R}^m \to \mathbb{R}$ $(\mu \in \mathbb{R}_{++})$, whose minimizers over the entire space $\mathbb{R}^m$ form the central trajectory in the space $\mathbb{R}^m$ of the Lagrange multiplier vector $\boldsymbol{y}$. For this purpose, we first introduce the dual logarithmic barrier function $f(\boldsymbol{y}, w, \boldsymbol{S}; \mu) = \sum_{p=1}^{m} a_p y_p + bw - \mu \log\det \boldsymbol{S}$ for every $(\boldsymbol{y}, w, \boldsymbol{S}, \mu) \in \mathbb{R}^{m+1} \times \mathbb{S}^n_{++} \times \mathbb{R}_{++}$ and then replace the objective function of (1.6) by $f(\boldsymbol{y}, w, \boldsymbol{S}; \mu)$:

$$\text{(1.7)}\qquad \begin{cases} \text{minimize} & f(\boldsymbol{y}, w, \boldsymbol{S}; \mu) \\ \text{subject to} & \boldsymbol{I} w - \boldsymbol{S} = \boldsymbol{C} - \displaystyle\sum_{p=1}^{m} \boldsymbol{A}_p y_p, \ \boldsymbol{S} \in \mathbb{S}^n_{++}. \end{cases}$$

The variable vector $\boldsymbol{y} \in \mathbb{R}^m$ of the dual SDP (1.6) is now a parameter vector given from outside. For every $(\boldsymbol{y}, \mu) \in \mathbb{R}^m \times \mathbb{R}_{++}$, problem (1.7) has a unique optimal solution, which we will denote by $(w(\boldsymbol{y}; \mu), \boldsymbol{S}(\boldsymbol{y}; \mu))$. More precisely, $(w, \boldsymbol{S}) \in \mathbb{R} \times \mathbb{S}^n_{++}$ is an optimal solution of problem (1.7) if and only if (1.8) holds for some $\boldsymbol{X} \in \mathbb{S}^n_{++}$.

$$\text{(1.8)}\quad \boldsymbol{I} \bullet \boldsymbol{X} = b, \ \boldsymbol{I} w - \boldsymbol{S} = \boldsymbol{C} - \sum_{p=1}^{m} \boldsymbol{A}_p y_p, \ \boldsymbol{X}\boldsymbol{S} = \mu \boldsymbol{I}, \quad \boldsymbol{X} \in \mathbb{S}^n_{++}, \ \boldsymbol{S} \in \mathbb{S}^n_{++}.$$

Now we define the function $g(\cdot; \mu) : \mathbb{R}^m \to \mathbb{R}$ as the optimal value of problem

(1.7); for every $(\boldsymbol{y}, \mu) \in \mathbb{R}^m \times \mathbb{R}_{++}$,

$$g(\boldsymbol{y};\mu) = \min \left\{ f(\boldsymbol{y}, w, \boldsymbol{S};\mu) : \ \boldsymbol{I}w - \boldsymbol{S} = \boldsymbol{C} - \sum_{p=1}^{m} \boldsymbol{A}_p y_p, \ \boldsymbol{S} \in \mathbb{S}^n_{++} \right\}$$

$$= \sum_{p=1}^{m} a_p y_p + bw(\boldsymbol{y};\mu) - \mu \log\det \boldsymbol{S}(\boldsymbol{y};\mu). \tag{1.9}$$

For every $\mu \in \mathbb{R}_{++}$, $g(\cdot;\mu) : \mathbb{R}^m \to \mathbb{R}$ turns out to be a smooth strictly convex function having a unique minimizer $\boldsymbol{y} = \boldsymbol{y}(\mu)$ over $\mathbb{R}^m$, and the set $\{(\boldsymbol{y}(\mu), \mu) : \mu \in \mathbb{R}_{++}\}$ forms a trajectory in the space $\mathbb{R}^m$ of the Lagrange multiplier vector $\boldsymbol{y}$.

Utilizing the function $g(\cdot;\mu)$ defined above on the entire m-dimensional Euclidean space $\mathbb{R}^m$ instead of the function $\tilde{g}(\cdot;\mu)$ restricted on $\mathcal{Y}_{++}$, our LDIPM numerically traces the trajectory $\{(\boldsymbol{y}(\mu), \mu) : \mu \in \mathbb{R}_{++}\}$ in a similar way as in the DIPM. In particular, the LDIPM shares the distinguishing features (a), (b), and (c) with the DIPM and is expected to offset the disadvantage (d) of the DIPM to a certain extent.

One crucial feature of using the function $g(\cdot;\mu)$ in the LDIPM is that

(e) for each $(\boldsymbol{y}, \mu) \in \mathbb{R}^m \times \mathbb{R}_{++}$, the evaluations of the function value $g(\boldsymbol{y};\mu)$, the gradient vector $\nabla g(\boldsymbol{y};\mu)$, and the Hessian matrix $\nabla^2 g(\boldsymbol{y};\mu)$ are done by solving the system of nonlinear equations (1.8) in $(\boldsymbol{X}, w, \boldsymbol{S}) \in \mathbb{S}^n_{++} \times \mathbb{R} \times \mathbb{S}^n_{++}$, which generally requires more than one Cholesky factorization in the set $\{\boldsymbol{S} = \sum_{p=1}^m \boldsymbol{A}_p y_p + \boldsymbol{I}w - \boldsymbol{C} \in \mathbb{S}^n_{++} : y_p \in \mathbb{R} \ (p = 1, 2, \ldots, m), \ w \in \mathbb{R}\}$ of dual matrix variable matrices, while the function $\tilde{g}(\cdot;\mu)$ is defined in terms of an explicit formula (1.3), and its evaluation requires only one Cholesky factorization.

The cost of the Cholesky factorization of $\boldsymbol{S} = \sum_{p=1}^m \boldsymbol{A}_p y_p + \boldsymbol{I}w - \boldsymbol{C} \in \mathbb{S}^n_{++}$ strongly depends on the data matrices $\boldsymbol{C}, \boldsymbol{A}_p$ $(p = 1, 2, \ldots, m)$ [10, 21]. In the development of our method, we implicitly assume that either the size of $\boldsymbol{S}$ is not large compared to the number of equality constraints m in the primal SDP (1.1), or the data matrices $\boldsymbol{C}, \boldsymbol{A}_p$ $(p = 1, 2, \ldots, m)$ enjoy a nice aggregated sparsity pattern that allows a cheap Cholesky factorization of $\boldsymbol{S}$. For detailed discussions on sparsity, see section 4.4.

We present more technical details on the LDIPM in section 2, a prototype algorithm of the LDIPM and its variants in section 3, and some additional techniques which enhance the effectiveness and efficiency of the LDIPM in section 4. In particular, we explain how we offset the disadvantage (e) of the LDIPM in section 4.1 and how we exploit the sparsity in the DIPM and the LDIPM in section 4.4. We report some preliminary numerical results on the DIPM and the LDIPM in section 5. We will also confirm there that the LDIPM works more efficiently than the DIPM.

Remark 1.2. The basic idea of using Lagrangian duals in interior-point methods was originally proposed for LPs in the working papers [15, 16]. However, neither of the papers was published because the method would be unlikely to compete with the primal-dual interior-point method for LPs, which had already become a powerful computational method for solving large scale LPs at that time, and also because some proofs of the main theorem of [16] were incomplete. This was pointed out by Gongyun Zhao. He later proposed an implementable version [30] of interior-point methods based on Lagrangian duals of linear programs and proved its polynomial-time computational complexity. See also [31, 32]. Finally, Shida [24] extended the Lagrangian dual interior-point method to linear optimization problems over pointed closed convex cones.

2. Basic analysis.

2.1. Characterization in the primal and the dual spaces. For every $\boldsymbol{y} \in \mathbb{R}^m$, let us consider a Lagrangian relaxation of SDP (1.5):

$$\text{(2.1)} \qquad \begin{cases} \text{maximize} & \left(\boldsymbol{C} - \sum_{p=1}^{m} \boldsymbol{A}_p y_p\right) \bullet \boldsymbol{X} + \sum_{p=1}^{m} a_p y_p \\ \text{subject to} & \boldsymbol{X} \in \Omega_+ = \{\boldsymbol{X} \in \mathbb{S}^n_+ : \boldsymbol{I} \bullet \boldsymbol{X} = b\} \end{cases}$$

and its dual

$$\text{(2.2)} \qquad \begin{cases} \text{minimize} & \sum_{p=1}^{m} a_p y_p + bw \\ \text{subject to} & \boldsymbol{I}w - \boldsymbol{S} = \boldsymbol{C} - \sum_{p=1}^{m} \boldsymbol{A}_p y_p,\ \boldsymbol{S} \in \mathbb{S}^n_+. \end{cases}$$

It should be noted that, for any $\boldsymbol{y} \in \mathbb{R}^m$, both SDPs (2.1) and (2.2) have interior feasible solutions; hence they both have optimal solutions, and their optimal values coincide with each other. This nice feature is due to the simplex constraint, which has not been incorporated into the Lagrangian relaxation and is maintained in (2.1).

The function $f(\cdot,\cdot,\cdot;\mu) : \mathbb{R}^{m+1} \times \mathbb{S}^n_{++} \to \mathbb{R}$ $(\mu \in \mathbb{R}_{++})$ given in the introduction corresponds to the logarithmic barrier function of the dual SDP (2.2), where we replaced the objective function $\sum_{p=1}^{m} a_p y_p + bw$ of (2.2) by $f(\boldsymbol{y}, w, \boldsymbol{S}; \mu)$ to obtain problem (1.7). We now consider the primal barrier function $f^p(\cdot,\cdot;\mu) : \mathbb{R}^m \times \mathbb{S}^n_{++} \to \mathbb{R}$ $(\mu \in \mathbb{R}_{++})$:

$$f^p(\boldsymbol{y}, \boldsymbol{X}; \mu) = \left(\boldsymbol{C} - \sum_{p=1}^{m} \boldsymbol{A}_p y_p\right) \bullet \boldsymbol{X} + \sum_{p=1}^{m} a_p y_p + \mu \log\det \boldsymbol{X}$$

for every $(\boldsymbol{y}, \boldsymbol{X}, \mu) \in \mathbb{R}^m \times \mathbb{S}^n_{++} \times \mathbb{R}_{++}$. Replacing the objective function $(\boldsymbol{C} - \sum_{p=1}^{m} \boldsymbol{A}_p y_p) \bullet \boldsymbol{X} + \sum_{p=1}^{m} a_p y_p$ by $f^p(\boldsymbol{y}, \boldsymbol{X}; \mu)$ in (2.1), we obtain the problem

$$\text{(2.3)} \qquad \begin{cases} \text{maximize} & f^p(\boldsymbol{y}, \boldsymbol{X}; \mu) \\ \text{subject to} & \boldsymbol{X} \in \Omega_{++} = \{\boldsymbol{X} \in \mathbb{S}^n_{++} : \boldsymbol{I} \bullet \boldsymbol{X} = b\}. \end{cases}$$

For every $(\boldsymbol{y}, \mu) \in \mathbb{R}^m \times \mathbb{R}_{++}$, problem (2.3) has a unique optimal solution, which we will denote by $\boldsymbol{X}(\boldsymbol{y}; \mu)$. Hence we can write the optimal value function $g^p(\cdot;\mu) : \mathbb{R}^m \to \mathbb{R}$ $(\mu \in \mathbb{R}_{++})$ of (2.3) as $g^p(\boldsymbol{y};\mu) \equiv \max\{f^p(\boldsymbol{y}, \boldsymbol{X}; \mu) : \boldsymbol{X} \in \Omega_{++}\} = f^p(\boldsymbol{y}, \boldsymbol{X}(\boldsymbol{y};\mu); \mu)$ for every $(\boldsymbol{y};\mu) \in \mathbb{R}^m \times \mathbb{R}_{++}$.

For each $(\boldsymbol{y}, \mu) \in \mathbb{R}^m \times \mathbb{R}_{++}$, problems (2.3) and (1.7) form a primal-dual pair. Among others, they share the common optimality condition (1.8); $\boldsymbol{X}$ is an optimal solution of the primal problem (2.3), and $(w, \boldsymbol{S})$ is an optimal solution of the dual problem (1.7) if and only if (1.8) holds. We will denote the unique solution of (1.8) by $(\boldsymbol{X}, w, \boldsymbol{S}) = (\boldsymbol{X}(\boldsymbol{y};\mu), w(\boldsymbol{y};\mu), \boldsymbol{S}(\boldsymbol{y};\mu))$. We further observe from (1.8) that the optimal value functions $g^p(\cdot;\mu)$ and $g(\cdot;\mu)$ satisfy

$$\begin{aligned} g(\boldsymbol{y};\mu) &= \sum_{p=1}^{m} a_p y_p + bw(\boldsymbol{y};\mu) - \mu \log\det \boldsymbol{S}(\boldsymbol{y};\mu) \\ &= \sum_{p=1}^{m} a_p y_p + \left(\boldsymbol{C} - \sum_{p=1}^{m} \boldsymbol{A}_p y_p + \boldsymbol{S}\right) \bullet \boldsymbol{X} - \mu \log\det\left(\mu \boldsymbol{X}^{-1}(\boldsymbol{y};\mu)\right) \\ &= g^p(\boldsymbol{y};\mu) + n\mu - n\mu \log \mu \end{aligned}$$

for every $(\boldsymbol{y}, \mu) \in \mathbb{R}^m \times \mathbb{R}_{++}$; hence the difference in the function values between $g(\boldsymbol{y}; \mu)$ and $g^p(\boldsymbol{y}; \mu)$ is constant independent of $\boldsymbol{y} \in \mathbb{R}^m$ as long as $\mu \in \mathbb{R}_{++}$ is fixed. Therefore, for each fixed $\mu \in \mathbb{R}_{++}$, the minimization of $g(\cdot; \mu)$ over $\mathbb{R}^m$ and the minimization of $g^p(\cdot; \mu)$ over $\mathbb{R}^m$ are equivalent.

Suppose that $\boldsymbol{X} \in \mathbb{S}^n_{++}$ and $\boldsymbol{S} \in \mathbb{S}^n_{++}$. Then we can describe the coefficient matrix $\widehat{\boldsymbol{M}}$ (or $\boldsymbol{M}$) of the so-called Schur complement equation for the HRVW/KSH/M search direction [14, 17, 19] applied to the primal-dual pair of SDPs (1.5) and (1.6) (or applied to the primal-dual pair of SDPs (1.1) and (1.2)) as follows:

$$\boldsymbol{M} \equiv \begin{pmatrix} M_{11} & M_{12} & \cdots & M_{1m} \\ M_{21} & M_{22} & \cdots & M_{2m} \\ \cdots & \cdots & \ddots & \cdots \\ M_{m1} & M_{m2} & \cdots & M_{mm} \end{pmatrix} \in \mathbb{S}^m,$$

$$\widehat{\boldsymbol{M}} \equiv \begin{pmatrix} \boldsymbol{M} & \boldsymbol{h} \\ \boldsymbol{h}^T & h_{m+1} \end{pmatrix} \in \mathbb{S}^{m+1},$$

where

$$\begin{aligned} M_{qr} &\equiv \boldsymbol{A}_q \bullet \boldsymbol{X}\boldsymbol{A}_r\boldsymbol{S}^{-1} \quad (q = 1, 2, \ldots, m,\ r = 1, 2, \ldots, m), \\ \boldsymbol{h} &\equiv (\boldsymbol{A}_1 \bullet \boldsymbol{X}\boldsymbol{I}\boldsymbol{S}^{-1}, \boldsymbol{A}_2 \bullet \boldsymbol{X}\boldsymbol{I}\boldsymbol{S}^{-1}, \ldots, \boldsymbol{A}_m \bullet \boldsymbol{X}\boldsymbol{I}\boldsymbol{S}^{-1})^T, \\ h_{m+1} &\equiv \boldsymbol{I} \bullet \boldsymbol{X}\boldsymbol{I}\boldsymbol{S}^{-1}. \end{aligned}$$

The matrix $\widehat{\boldsymbol{M}}$ (or $\boldsymbol{M}$) is known to be symmetric and positive definite. When the primal-dual pair of matrix variables $\boldsymbol{X} \in \mathbb{S}^n_{++}$ and $\boldsymbol{S} \in \mathbb{S}^n_{++}$ satisfies $\boldsymbol{X}\boldsymbol{S} = \mu\boldsymbol{I}$ for some $\mu \in \mathbb{R}_{++}$ as in the succeeding discussions, we have

$$\begin{aligned} M_{qr} &= \mu\boldsymbol{A}_q \bullet \boldsymbol{S}^{-1}\boldsymbol{A}_r\boldsymbol{S}^{-1} \quad (q = 1, 2, \ldots, m,\ r = 1, 2, \ldots, m), \\ \boldsymbol{h} &= (\mu\boldsymbol{A}_1 \bullet \boldsymbol{S}^{-2}, \mu\boldsymbol{A}_2 \bullet \boldsymbol{S}^{-2}, \ldots, \mu\boldsymbol{A}_m \bullet \boldsymbol{S}^{-2})^T, \\ h_{m+1} &= \mu\boldsymbol{I} \bullet \boldsymbol{S}^{-2}. \end{aligned} \tag{2.4}$$

In this case, the matrix $\widehat{\boldsymbol{M}}$ (or $\boldsymbol{M}$) is also identical to the coefficient matrix of the Schur complement equation for other well-known search directions, including the AHO direction [1] and the NT direction [26].

2.2. Computation of the function value, the gradient vector, and the Hessian matrix of $g(\cdot; \mu)$. Let $(\boldsymbol{y}, \mu) \in \mathbb{R}^m \times \mathbb{R}_{++}$. In order to compute the function value, the gradient vector, and the Hessian matrix of $g(\boldsymbol{y}; \mu)$, it suffices to solve the system of equations (1.8). In fact, if $(\boldsymbol{X}, w, \boldsymbol{S}) = (\boldsymbol{X}(\boldsymbol{y}; \mu), w(\boldsymbol{y}; \mu), \boldsymbol{S}(\boldsymbol{y}; \mu)) \in \mathbb{S}^n_{++} \times \mathbb{R} \times \mathbb{S}^n_{++}$ is a solution of (1.8), then the function value is computed as

$$\begin{aligned} g(\boldsymbol{y}; \mu) &= f(\boldsymbol{y}, w(\boldsymbol{y}; \mu), \boldsymbol{S}(\boldsymbol{y}; \mu); \mu) \\ &= \sum_{p=1}^m a_p y_p + bw(\boldsymbol{y}; \mu) - \mu \log\det \boldsymbol{S}(\boldsymbol{y}; \mu), \end{aligned}$$

and the gradient vector and the Hessian matrix are given by the following lemma.

LEMMA 2.1.

(i) $\nabla g(\boldsymbol{y}; \mu) = (a_1 - \boldsymbol{A}_1 \bullet \boldsymbol{X}(\boldsymbol{y}; \mu), a_2 - \boldsymbol{A}_2 \bullet \boldsymbol{X}(\boldsymbol{y}; \mu), \ldots, a_m - \boldsymbol{A}_m \bullet \boldsymbol{X}(\boldsymbol{y}; \mu))^T$.

(ii) $\nabla^2 g(\boldsymbol{y}; \mu) = (\boldsymbol{M} - \boldsymbol{h}\boldsymbol{h}^T/h_{m+1})$. *Here* $\boldsymbol{M} \in \mathbb{S}^m$, $\boldsymbol{h} \in \mathbb{R}^m$, *and* $h_{m+1} \in \mathbb{R}$ *are given by* (2.4) *with* $\boldsymbol{S} = \boldsymbol{S}(\boldsymbol{y}; \mu)$.

(iii) *Moreover, the Hessian matrix* $\nabla^2 g(\boldsymbol{y};\mu)$ *is positive definite.*

Proof. This is a special case for the cone of positive semidefinite matrices [24]. □

We can reduce the system of equations (1.8) to a single equation

$$\mu \boldsymbol{I} \bullet \left(\sum_{p=1}^{m} \boldsymbol{A}_p y_p + \boldsymbol{I} w - \boldsymbol{C} \right)^{-1} = b$$

in the single variable w with an additional positive definite condition

$$\left(\sum_{p=1}^{m} \boldsymbol{A}_p y_p + \boldsymbol{I} w - \boldsymbol{C} \right) \in \mathbb{S}^n_{++}. \tag{2.5}$$

Then the other parts $\boldsymbol{X} = \boldsymbol{X}(\boldsymbol{y};\mu)$ and $\boldsymbol{S} = \boldsymbol{S}(\boldsymbol{y};\mu)$ of the solution of (1.8) are computed by

$$\boldsymbol{S} = \left(\sum_{p=1}^{m} \boldsymbol{A}_p y_p + \boldsymbol{I} w - \boldsymbol{C} \right) \quad \text{and} \quad \boldsymbol{X} = \mu \boldsymbol{S}^{-1}. \tag{2.6}$$

Let $\mu \in \mathbb{R}_{++}$ and $\boldsymbol{y} \in \mathbb{R}^m$ be fixed arbitrary. Define

$$\phi(w;\boldsymbol{y},\mu) = \mu \boldsymbol{I} \bullet \left(\sum_{p=1}^{m} \boldsymbol{A}_p y_p + \boldsymbol{I} w - \boldsymbol{C} \right)^{-1}$$

for every w satisfying (2.5). We want to solve $\phi(w;\boldsymbol{y},\mu) = b$. Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the eigenvalues of the matrix $\boldsymbol{C} - \sum_{p=1}^{m} \boldsymbol{A}_p y_p$ involved in the definition of $\phi(\cdot;\boldsymbol{y},\mu)$. Let $\lambda_{\max}$ denote the maximum eigenvalue among $\lambda_1, \lambda_2, \ldots, \lambda_n$. Then the positive definite condition (2.5) turns out to be $w > \lambda_{\max}$, and

$$\begin{aligned} \phi(w;\boldsymbol{y},\mu) &= \mu \sum_{i=1}^{n} \frac{1}{w - \lambda_i}, \\ \frac{d\phi(w;\boldsymbol{y},\mu)}{dw} &= -\mu \sum_{i=1}^{n} \frac{1}{(w-\lambda_i)^2} = -\mu \boldsymbol{I} \bullet \left(\sum_{p=1}^{m} \boldsymbol{A}_p y_p + \boldsymbol{I} w - \boldsymbol{C} \right)^{-2}. \end{aligned} \tag{2.7}$$

We also see that $\phi(\cdot;\boldsymbol{y},\mu)$ is a strictly convex and strictly decreasing function on $(\lambda_{\max}, +\infty)$ and that it satisfies $\phi(w;\boldsymbol{y},\mu) \to +\infty$ as $w \to \lambda^+_{\max}$, and $\phi(w;\boldsymbol{y},\mu) \to 0$ as $w \to +\infty$. Therefore the equation $\phi(w;\boldsymbol{y},\mu) = b$ has the unique solution w^* in the interval $[\lambda_{\max} + \mu/b, \lambda_{\max} + n\mu/b]$.

To solve $\phi(w;\boldsymbol{y},\mu) = b$, we will utilize a method similar to the one often employed in trust region methods for nonlinear unconstrained optimization [7]. Let $\bar{w} \in [\lambda_{\max} + \mu/b, \lambda_{\max} + n\mu/b]$ be a current iterate. We approximate the function $\phi(\cdot;\boldsymbol{y},\mu)$ by a function of the form $\psi(w) = \beta/(w-\alpha)$. Here, the two real parameters α and β are determined by

$$0 < \bar{w} - \alpha, \quad \psi(\bar{w}) = \phi(\bar{w};\boldsymbol{y},\mu), \quad \text{and} \quad \frac{d\psi(\bar{w})}{dw} = \frac{d\phi(\bar{w};\boldsymbol{y},\mu)}{dw}.$$

We choose the next iterate $w^+ = \beta/b + \alpha$ by solving the equation $\psi(w) = b$.

At each iteration of this algorithm, we need to evaluate

$$\phi(w;\boldsymbol{y},\mu) = \mu \boldsymbol{I} \bullet \boldsymbol{S}^{-1} \quad \text{and} \quad \frac{d\phi(w;\boldsymbol{y},\mu)}{dw} = -\mu \boldsymbol{S}^{-1} \bullet \boldsymbol{S}^{-1},$$

where $\boldsymbol{S} = \sum_{p=1}^{m} \boldsymbol{A}_p y_p + \boldsymbol{I} w - \boldsymbol{C}$. In general, this requires $O(mn^2 + n^3)$ arithmetic operations. Or, if we compute all the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ of the matrix $\boldsymbol{C} - \sum_{p=1}^{m} \boldsymbol{A}_p y_p$ in advance, we can easily compute the value and the first derivative of the function $\phi(\cdot;\boldsymbol{y},\mu)$ using the relations (2.7). When all the data matrices $\boldsymbol{C}, \boldsymbol{A}_p$ $(p = 1, 2, \ldots, m)$ are sparse, the former method is cheaper than the latter unless the number of iterations gets larger. This will be discussed again in section 4.4.

2.3. Approximation of the central trajectory in the *y*-space. This section will provide theoretical foundations of the predictor procedure of the LDIPM.

Let $\mu \in \mathbb{R}_{++}$. Assume that $\boldsymbol{y} = \boldsymbol{y}(\mu) \in \mathbb{R}^m$ is the unique minimizer of the function $g(\cdot;\mu)$ over $\mathbb{R}^m$. In view of Lemma 2.1, $\boldsymbol{y} = \boldsymbol{y}(\mu)$ satisfies $\boldsymbol{A}_p \bullet \boldsymbol{X}(\boldsymbol{y};\mu) = a_p$ $(p = 1, 2, \ldots, m)$. Recall also that $\boldsymbol{X}(\boldsymbol{y};\mu)$ is characterized by the condition (1.8). Therefore $\boldsymbol{y} = \boldsymbol{y}(\mu)$ satisfies

$$\begin{aligned} &\boldsymbol{A}_p \bullet \boldsymbol{X} = a_p \ (p = 1, 2, \ldots, m), \quad \boldsymbol{I} \bullet \boldsymbol{X} = b, \\ &\sum_{p=1}^{m} \boldsymbol{A}_p y_p + \boldsymbol{I} w - \boldsymbol{S} = \boldsymbol{C}, \quad \boldsymbol{X}\boldsymbol{S} = \mu \boldsymbol{I}, \quad \boldsymbol{X} \in \mathbb{S}^n_{++}, \ \boldsymbol{S} \in \mathbb{S}^n_{++}, \end{aligned} \tag{2.8}$$

for some $(\boldsymbol{X}, w, \boldsymbol{S}) = (\boldsymbol{X}(\boldsymbol{y};\mu), w(\boldsymbol{y};\mu), \boldsymbol{S}(\boldsymbol{y};\mu)) \in \mathbb{S}^n_{++} \times \mathbb{R} \times \mathbb{S}^n_{++}$. This is exactly the condition that characterizes the central trajectory of the primal-dual SDPs (1.5) and (1.6). Thus, we may regard $\{(\boldsymbol{y}(\mu), \mu) : \mu \in \mathbb{R}_{++}\}$ as the projection of the central trajectory on the space $\mathbb{R}^m$ of the Lagrange multiplier vector $\boldsymbol{y}$. In the remainder of this section, we derive systems of linear equations that determine the first derivative $\dot{\boldsymbol{y}}(\mu)$ and the second derivative $\ddot{\boldsymbol{y}}(\mu)$. With these derivatives, we can approximate the trajectory $\{(\boldsymbol{y}(\mu), \mu) : \mu \in \mathbb{R}_{++}\}$ by either

$$\{(\boldsymbol{y}(\bar{\mu}) + (\mu - \bar{\mu})\dot{\boldsymbol{y}}(\bar{\mu}), \mu) : \mu \in \mathbb{R}_{++}\}$$

(the first-order approximation) or

$$\left\{\left(\boldsymbol{y}(\bar{\mu}) + (\mu - \bar{\mu})\dot{\boldsymbol{y}}(\bar{\mu}) + \frac{(\mu - \bar{\mu})^2}{2}\ddot{\boldsymbol{y}}(\bar{\mu}), \mu\right) : \mu \in \mathbb{R}_{++}\right\}$$

(the second-order approximation) in a neighborhood of each $\bar{\mu} \in \mathbb{R}_{++}$.

For simplicity of notation, we write $(\boldsymbol{X}, w, \boldsymbol{S}) = (\boldsymbol{X}(\boldsymbol{y}(\mu);\mu), w(\boldsymbol{y}(\mu);\mu), \boldsymbol{S}(\boldsymbol{y}(\mu);\mu))$. Then $(\boldsymbol{X}, \boldsymbol{y}, w, \boldsymbol{S})$ satisfies all the identities in (2.8). Differentiating those identities once and twice in μ, we obtain

$$\begin{aligned} &\boldsymbol{A}_p \bullet \dot{\boldsymbol{X}} = 0 \ (p = 1, 2, \ldots, m), \quad \boldsymbol{I} \bullet \dot{\boldsymbol{X}} = 0, \\ &\sum_{p=1}^{m} \boldsymbol{A}_p \dot{y}_p + \boldsymbol{I}\dot{w} - \dot{\boldsymbol{S}} = \boldsymbol{O}, \quad \dot{\boldsymbol{X}}\boldsymbol{S} + \boldsymbol{X}\dot{\boldsymbol{S}} = \boldsymbol{I} \end{aligned}$$

and

$$\begin{aligned} &\boldsymbol{A}_p \bullet \ddot{\boldsymbol{X}} = 0 \ (p = 1, 2, \ldots, m), \quad \boldsymbol{I} \bullet \ddot{\boldsymbol{X}} = 0, \\ &\sum_{p=1}^{m} \boldsymbol{A}_p \ddot{y}_p + \boldsymbol{I}\ddot{w} - \ddot{\boldsymbol{S}} = \boldsymbol{O}, \quad \ddot{\boldsymbol{X}}\boldsymbol{S} + \boldsymbol{X}\ddot{\boldsymbol{S}} = -2\dot{\boldsymbol{X}}\dot{\boldsymbol{S}}, \end{aligned}$$

respectively. Here $(\dot{\boldsymbol{X}}, \dot{\boldsymbol{y}}, \dot{w}, \dot{\boldsymbol{S}})$ and $(\ddot{\boldsymbol{X}}, \ddot{\boldsymbol{y}}, \ddot{w}, \ddot{\boldsymbol{S}})$ denote the first and second derivatives of $(\boldsymbol{X}, \boldsymbol{y}, w, \boldsymbol{S})$ in relation to μ, respectively. From these equations and Lemma 2.1(ii), we obtain

$$\begin{aligned} \nabla^2 g(\boldsymbol{y};\mu)\dot{\boldsymbol{y}} &= \bar{\boldsymbol{a}} \equiv \frac{\boldsymbol{a} - b\boldsymbol{h}/h_{m+1}}{\mu}, \qquad \dot{w} = \frac{b/\mu - \boldsymbol{h}^T\dot{\boldsymbol{y}}}{h_{m+1}}, \\ \dot{\boldsymbol{S}} &= \sum_{p=1}^{m} \boldsymbol{A}_p \dot{y}_p + \boldsymbol{I}\dot{w}, \\ \nabla^2 g(\boldsymbol{y};\mu)\ddot{\boldsymbol{y}} &= \bar{\boldsymbol{r}} \equiv \frac{\boldsymbol{r} - r_{m+1}\boldsymbol{h}/h_{m+1}}{\mu}, \qquad \ddot{w} = \frac{r_{m+1}/\mu - \boldsymbol{h}^T\ddot{\boldsymbol{y}}}{h_{m+1}}, \end{aligned} \tag{2.9}$$

where h_{m+1} and $\boldsymbol{h}$ are given by (2.4), and r_q $(q = 1, 2, \ldots, m)$, r_{m+1} are given by

$$\begin{aligned} r_q &= -2\mu \boldsymbol{A}_q \bullet (\boldsymbol{S}^{-1} - \mu \boldsymbol{S}^{-1}\dot{\boldsymbol{S}}\boldsymbol{S}^{-1})\dot{\boldsymbol{S}}\boldsymbol{S}^{-1} \in \mathbb{R} \ (q = 1, 2, \ldots, m), \\ \boldsymbol{r} &= (r_1, r_2, \ldots, r_m)^T \in \mathbb{R}^m, \qquad \boldsymbol{a} = (a_1, a_2, \ldots, a_m)^T \in \mathbb{R}^m, \\ r_{m+1} &= -2\mu \boldsymbol{I} \bullet (\boldsymbol{S}^{-1} - \mu \boldsymbol{S}^{-1}\dot{\boldsymbol{S}}\boldsymbol{S}^{-1})\dot{\boldsymbol{S}}\boldsymbol{S}^{-1} \in \mathbb{R}. \end{aligned} \tag{2.10}$$

In view of Lemma 2.1(iii), we know that the common coefficient matrix $\nabla^2 g(\boldsymbol{y};\mu) = \boldsymbol{M} - \boldsymbol{h}\boldsymbol{h}^T/h_{m+1}$ of the equations in (2.9) is positive definite, so that we can utilize its Cholesky factorization or iterative methods such as the CG method and the CR method to solve them.

3. Predictor-corrector path-following algorithms in the space $\mathbb{R}^m$ of the Lagrange multiplier vector $\boldsymbol{y}$.

3.1. A prototype algorithm using the Newton method. Among various candidates for neighborhoods of the trajectory $\{(\boldsymbol{y}(\mu), \mu) \in \mathbb{R}^m \times \mathbb{R}_{++} : \mu \in \mathbb{R}_{++}\}$, we employ the one based on the self-concordant theory [23]. In the LP case, Kojima et al. [15] showed that $\{g(\cdot;\mu) : \mu \in \mathbb{R}_{++}\}$ forms a self-concordant family. Shida [24] extended this fact to a general class of linear optimization problems over pointed closed convex cones, which includes our SDP case. For every $\epsilon \in \mathbb{R}_{++}$, define

$$N(\epsilon) = \left\{(\boldsymbol{y}, \mu) \in \mathbb{R}^m \times \mathbb{R}_{++} : \nabla g(\boldsymbol{y};\mu)^T \nabla^2 g(\boldsymbol{y};\mu)^{-1} \nabla g(\boldsymbol{y};\mu) \le \mu\epsilon\right\}.$$

ALGORITHM 3.1.

Step 0: *Let* $0 < \epsilon_c < \epsilon_p$, $0 < \gamma < 1$. *Choose a* $\mu^0 \in \mathbb{R}_{++}$ *and a* $\bar{\boldsymbol{y}}^0 \in \mathbb{R}^m$. *Let* $k = 0$.

Step 1-*N (Corrector procedure using the Newton method): Let* $\boldsymbol{z} = \bar{\boldsymbol{y}}^k$. *To approximately solve the problem of minimizing the strictly convex smooth function* $g(\cdot;\mu^k)$ *over* $\mathbb{R}^m$, *repeat the damped Newton iteration*

- *solve* $\nabla^2 g(\boldsymbol{z};\mu^k)\boldsymbol{d} = -\nabla g(\boldsymbol{z};\mu^k)$ *in the search direction* $\boldsymbol{d}$,
- *choose a step length* $\alpha \in (0, 1]$ *such that* $\boldsymbol{z}^+ = \boldsymbol{z} + \alpha\boldsymbol{d} \in \mathbb{R}^m$; *for example, use Armijo's line search rule, Wolfe's line search rule, or a quadratic approximation of* $g(\boldsymbol{z} + \alpha\boldsymbol{d};\mu)$ *(see section* 4.2 *for more details),*
- *replace* $\boldsymbol{z}^+$ *by* $\boldsymbol{z}$

until $(\boldsymbol{z}, \mu^k) \in N(\epsilon_c)$. *Let* $\boldsymbol{y}^k = \boldsymbol{z}$. *For the computation of the function value* $g(\boldsymbol{z};\mu^k)$, *the gradient vector* $\nabla g(\boldsymbol{z};\mu^k)$, *and the Hessian matrix* $\nabla^2 g(\boldsymbol{z};\mu^k)$, *see section* 2.2.

Step 2-*N (Predictor procedure using the Newton method): Let* $(\boldsymbol{X}^k, w^k, \boldsymbol{S}^k) \in \mathbb{S}^n_{++} \times \mathbb{R} \times \mathbb{S}^n_{++}$ *be an approximate solution of* (2.8) *with* $(\boldsymbol{X}, \boldsymbol{y}, w, \boldsymbol{S}, \mu) = (\boldsymbol{X}^k, \boldsymbol{y}^k, w^k, \boldsymbol{S}^k, \mu^k)(\boldsymbol{A}_p \bullet \boldsymbol{X}^k \approx a_p (p = 1, 2, \ldots, m), \boldsymbol{I} \bullet \boldsymbol{X}^k \approx b)$. *Compute an approximation* $\dot{\boldsymbol{y}} = \dot{\boldsymbol{y}}^k \in \mathbb{R}^m$ *of the first derivative* $\dot{\boldsymbol{y}}(\mu^k)$ *and an approximation*

$\ddot{\boldsymbol{y}} = \ddot{\boldsymbol{y}}^k \in \mathbb{R}^m$ *of the second derivative* $\ddot{\boldsymbol{y}}(\mu^k)$ *by solving the systems of linear equations* (2.9) *with* $(\boldsymbol{X}, \boldsymbol{y}, w, \boldsymbol{S}, \mu) = (\boldsymbol{X}^k, \boldsymbol{y}^k, w^k, \boldsymbol{S}^k, \mu^k)$. *Let* $\delta = 1$ *and* $\gamma \in (0,1)$. *Repeat*

- $\delta = \gamma\delta$, $\bar{\mu} = (1-\delta)\mu^k$ *and*
- $\bar{\boldsymbol{y}} = \boldsymbol{y}^k + (\bar{\mu} - \mu^k)\dot{\boldsymbol{y}}^k + \frac{(\bar{\mu}-\mu^k)^2}{2}\ddot{\boldsymbol{y}}^k$ *(the second-order predictor procedure) until* $(\bar{\boldsymbol{y}}, \bar{\mu}) \in N(\epsilon_p)$. *Let* $\mu^{k+1} = \bar{\mu}$ *and* $\bar{\boldsymbol{y}}^{k+1} = \bar{\boldsymbol{y}}$.

Step 3: *Replace* $k+1$ *by* k *and return to Step* 1-N.

In Step 2-N, we may replace the second-order predictor procedure by the first-order predictor procedure $\bar{\boldsymbol{y}} = \boldsymbol{y}^k + (\bar{\mu} - \mu^k)\dot{\boldsymbol{y}}^k$. An approximation $\ddot{\boldsymbol{y}} = \ddot{\boldsymbol{y}}^k \in \mathbb{R}^m$ of the second derivative $\ddot{\boldsymbol{y}}(\mu^k)$ is unnecessary in this case.

To design efficient algorithms for large scale SDPs based on the prototype algorithm described above, we need to incorporate various practical techniques which have been developed in the field of unconstrained optimization. In particular, we can employ the BFGS quasi-Newton method for the minimization of the function $g(\cdot;\mu^k)$ over the entire Euclidean space $\mathbb{R}^m$. This will be discussed in section 3.2. Another important technique is an application of iterative methods such as the CG method and the CR method for solving the systems of linear equations (2.9). This will be discussed in section 3.3.

3.2. Corrector procedure using the quasi-Newton BFGS method. In Step 0-BFGS, we initialize $\boldsymbol{H} = \boldsymbol{I} \in \mathbb{S}^n_{++}$, which we will update to approximate the inverse of the Hessian matrix $\boldsymbol{\nabla}^2 g(\boldsymbol{z};\mu^k)$ in Step 1-BFGS below.

Step 1-*BFGS (Corrector procedure using the BFGS quasi-Newton method): Let* $\boldsymbol{z} = \bar{\boldsymbol{y}}^k$. *To approximately solve the problem of minimizing the strictly convex smooth function* $g(\cdot;\mu^k)$ *over* $\mathbb{R}^m$, *repeat the BFGS quasi-Newton iteration*

- $\boldsymbol{d} = -\boldsymbol{H}\boldsymbol{\nabla} g(\boldsymbol{z};\mu^k)$ *(note that* $\boldsymbol{H}$ *corresponds to an approximation of* $(\boldsymbol{\nabla}^2 g(\boldsymbol{z};\mu^k))^{-1}$*),*
- *choose a step length* $\alpha \in (0,1]$ *such that* $\boldsymbol{z}^+ = \boldsymbol{z} + \alpha\boldsymbol{d} \in \mathbb{R}^m$*; for example, use Armijo's line search rule, Wolfe's line search rule, or a quadratic approximation of* $g(\boldsymbol{z} + \alpha\boldsymbol{d};\mu)$ *(see section* 4.2 *for more details),*
- *let* $\boldsymbol{H}^+ = \boldsymbol{H} - \frac{\boldsymbol{H}\boldsymbol{\eta}\boldsymbol{\sigma}^T + \boldsymbol{\sigma}(\boldsymbol{H}\boldsymbol{\eta})^T}{\boldsymbol{\sigma}^T\boldsymbol{\eta}} + \left(1 + \frac{\boldsymbol{\eta}^T\boldsymbol{H}\boldsymbol{\eta}}{\boldsymbol{\sigma}^T\boldsymbol{\eta}}\right)\frac{\boldsymbol{\sigma}\boldsymbol{\sigma}^T}{\boldsymbol{\sigma}^T\boldsymbol{\eta}}$, *where* $\boldsymbol{\sigma} = \boldsymbol{z}^+ - \boldsymbol{z}$ *and* $\boldsymbol{\eta} = \boldsymbol{\nabla} g(\boldsymbol{z}^+;\mu^k) - \boldsymbol{\nabla} g(\boldsymbol{z};\mu^k)$ *(a BFGS quasi-Newton update),*
- *replace* $\boldsymbol{z}^+$ *by* $\boldsymbol{z}$, *and* $\boldsymbol{H}^+$ *by* $\boldsymbol{H}$

until

$$\boldsymbol{\nabla} g(\boldsymbol{z};\mu^k)^T \boldsymbol{H} \boldsymbol{\nabla} g(\boldsymbol{z};\mu^k) \le \mu\epsilon_c. \tag{3.1}$$

Let $\boldsymbol{y}^k = \boldsymbol{z}$.

It should be noted that the stopping criterion $(\boldsymbol{z},\mu^k) \in N(\epsilon_c)$, which also requires us to solve $\boldsymbol{\nabla}^2 g(\boldsymbol{z};\mu^k)\boldsymbol{d} = -\boldsymbol{\nabla} g(\boldsymbol{z};\mu^k)$ in $\boldsymbol{d}$, for the Newton iteration in Step 1-N of Algorithm 3.1 has been now replaced by (3.1).

3.3. Predictor procedure using preconditioned iterative methods. In the predictor procedure, Step 2-N of Algorithm 3.1, we need to compute an approximation $\dot{\boldsymbol{y}} = \dot{\boldsymbol{y}}^k \in \mathbb{R}^m$ of the first derivative $\dot{\boldsymbol{y}}(\mu^k)$, and an approximation $\ddot{\boldsymbol{y}} = \ddot{\boldsymbol{y}}^k \in \mathbb{R}^m$ of the second derivative $\ddot{\boldsymbol{y}}(\mu^k)$ by solving the systems of linear equations (2.9) with $(\boldsymbol{X}, \boldsymbol{y}, w, \boldsymbol{S}, \mu) = (\boldsymbol{X}^k, \boldsymbol{y}^k, w^k, \boldsymbol{S}^k, \mu^k)$. Since the common coefficient matrix $\boldsymbol{\nabla}^2 g(\boldsymbol{y};\mu)$ of the equations on $\dot{\boldsymbol{y}}$ and $\ddot{\boldsymbol{y}}$ in (2.9) is positive definite by Lemma 2.1(iii), we can utilize its Cholesky factorization to solve the first and the second equations exactly. But the coefficient matrix is fully dense in general, and solving the equations

exactly by the Cholesky factorization gets more and more expensive when the dimension m of the equations gets larger. We propose below to use iterative methods such as the CG method and the CR method.

Let $\nabla^2 g(\boldsymbol{y};\mu)\boldsymbol{u} = \boldsymbol{v}$ represent either of the equations in $\dot{\boldsymbol{y}}$ and $\ddot{\boldsymbol{y}}$ in system (2.9). Let $\boldsymbol{H}^k$ denote the BFGS quasi-Newton matrix, which we have computed in the corrector procedure of the kth iteration. We will use this matrix as a left preconditioner when we apply the CG method (or the CR method) to $\nabla^2 g(\boldsymbol{y};\mu)\boldsymbol{u} = \boldsymbol{v}$. Specifically, multiplying $\nabla^2 g(\boldsymbol{y};\mu)\boldsymbol{u} = \boldsymbol{v}$ by $\boldsymbol{H}^k$ from the left side, we transform it into the equivalent system of linear equations $\boldsymbol{H}^k\nabla^2 g(\boldsymbol{y}^k;\mu^k)\boldsymbol{u} = \boldsymbol{H}^k\boldsymbol{v}$ and then apply the CG method (or the CR method) to the resulting system from an initial point $\boldsymbol{u}^0 = \boldsymbol{H}^k\boldsymbol{v}$. If $\boldsymbol{H}^k$ approximates $(\nabla^2 g(\boldsymbol{y}^k;\mu^k))^{-1}$ with a reasonable accuracy, or at least, if $\boldsymbol{H}^k$ captures the eigenvalue structure of $(\nabla^2 g(\boldsymbol{y}^k;\mu^k))^{-1}$, this preconditioning technique considerably improves the condition number of the original coefficient matrix $\nabla^2 g(\boldsymbol{y}^k;\mu^k)$.

We use an approximated scaled norm, which we have used in Step 1-BFGS (3.1), as a stopping criteria for the CG method; more precisely, we stop the CG iteration when

$$(\nabla^2 g(\boldsymbol{y}^k;\mu^k)\boldsymbol{u} - \boldsymbol{v})^T \boldsymbol{H}^k (\nabla^2 g(\boldsymbol{y}^k;\mu^k)\boldsymbol{u} - \boldsymbol{v}) < \mu^k \epsilon_{cg}. \tag{3.2}$$

Here $\boldsymbol{u}$ denotes an iterate of the CG method.

At each iteration of the CG method, we need to compute a vector $\nabla^2 g(\boldsymbol{z};\mu)\boldsymbol{u} \in \mathbb{R}^m$ for some $\boldsymbol{u} \in \mathbb{R}^m$. In general, the computation of the vector $\nabla^2 g(\boldsymbol{z};\mu)\boldsymbol{u}$ is much cheaper than the computation of the entire Hessian matrix $\nabla^2 g(\boldsymbol{z};\mu)$.

3.4. A note on the simple DIPM. In the introduction we introduced the logarithmic barrier function $\tilde{g}(\cdot;\mu) : \mathcal{Y}_{++} \to \mathbb{R}$ and the associated family (1.4) of strictly convex minimization problems. After that, we presented a simple dual predictor-corrector interior-point method, the DIPM, for SDP (1.2). Recall that the LDIPM is a variant of the DIPM. In this section, we show that all of the discussions in sections 3.1, 3.2, and 3.3 can be easily simplified to adapt them to the DIPM.

We first replace Condition 1.1 by the following.

CONDITION 3.2.

(i) *There is an interior feasible solution* $\boldsymbol{X}^0$ *of the primal SDP* (1.1).

(ii) *An interior feasible solution* $(\boldsymbol{y}^0, \boldsymbol{S}^0) \in \mathbb{R}^m \times \mathbb{S}^n_{++}$ *of the dual SDP* (1.2) *is known in advance.*

(iii) *The data matrices* $\boldsymbol{A}_p$ $(p = 1, 2, \ldots, m)$ *are linearly independent.*

For each $(\boldsymbol{y}, \mu) \in \mathcal{Y}_{++} \times \mathbb{R}_{++}$, let $\boldsymbol{S}(\boldsymbol{y};\mu) = \sum_{p=1}^m \boldsymbol{A}_p y_p - \boldsymbol{C}$ and $\boldsymbol{X}(\boldsymbol{y};\mu) = \mu \boldsymbol{S}(\boldsymbol{y};\mu)^{-1}$. Then the gradient and the Hessian matrix of $\tilde{g}(\cdot;\mu)$ at each $(\boldsymbol{y}, \mu) \in \mathcal{Y}_{++} \times \mathbb{R}_{++}$ are given by

$$\nabla \tilde{g}(\boldsymbol{y};\mu) = (a_1 - \boldsymbol{A}_1 \bullet \boldsymbol{X}(\boldsymbol{y};\mu), a_2 - \boldsymbol{A}_2 \bullet \boldsymbol{X}(\boldsymbol{y};\mu), \ldots, a_m - \boldsymbol{A}_m \bullet \boldsymbol{X}(\boldsymbol{y};\mu))^T$$

and

$$\nabla^2 \tilde{g}(\boldsymbol{y};\mu) = \boldsymbol{M},$$

respectively, where $\boldsymbol{M} \in \mathbb{S}^m$ is given by (2.4) with $\boldsymbol{S} = \boldsymbol{S}(\boldsymbol{y};\mu)$.

Now we are ready to relate the DIPM to the LDIPM. All the discussions in sections 3.1, 3.2, and 3.3 remain valid if we replace

- $\mathbb{R}^m$, on which $g(\cdot;\mu)$ $(\mu \in \mathbb{R}_{++})$ is defined, by the set $\mathcal{Y}_{++}$, on which $\tilde{g}(\cdot;\mu)$ $(\mu \in \mathbb{R}_{++})$ is defined,

- the systems (2.9) of linear equations in the first derivative $\dot{\boldsymbol{y}}$ and the second derivative $\ddot{\boldsymbol{y}}$ by

$$\boldsymbol{\nabla}^2\tilde{g}(\boldsymbol{y};\mu)\dot{\boldsymbol{y}} = \frac{\boldsymbol{a}}{\mu}, \quad \dot{\boldsymbol{S}} = \sum_{p=1}^{m} \boldsymbol{A}_p\dot{y}_p, \quad \boldsymbol{\nabla}^2\tilde{g}(\boldsymbol{y};\mu)\ddot{\boldsymbol{y}} = \frac{\boldsymbol{r}}{\mu},$$

  where $\boldsymbol{a}, \boldsymbol{r} \in \mathbb{R}^m$ are given by (2.10).

Also we can easily adapt all the discussions on the LDIPM in sections 4.2, 4.3, 4.4, and 4.5 to the DIPM.

Major differences between the DIPM and the LDIPM are the following:

(i) The explicit formula of $\boldsymbol{X}(\boldsymbol{y};\mu)$ is available in the DIPM, while $\boldsymbol{X}(\boldsymbol{y};\mu)$ in the LDIPM is computed through an iterative method (section 2.2).

(ii) The Lagrange multiplier vector $\boldsymbol{y}$ of the DIPM is restricted to $\mathcal{Y}_{++}$, while $\boldsymbol{y}$ of the LDIPM can vary in the entire space $\mathbb{R}^m$.

Although the DIPM is more attractive than the LDIPM due to feature (i), numerical results, which we will report in section 5, support that the LDIPM is more efficient than the DIPM. We have not discovered the exact reason, but feature (ii) is probably a critical weak point of the DIPM, because when $\boldsymbol{y} \in \mathcal{Y}_{++}$ is near to the boundary of $\mathcal{Y}_{++}$, a little perturbation to $(\boldsymbol{y}, \mu)$ may cause a drastic change in $\boldsymbol{X}(\boldsymbol{y};\mu)$ and $g(\boldsymbol{y};\mu)$.

4. Additional techniques.

4.1. Dynamical adjustment of b. In many applications of the SDP of the form (1.5), it is not necessary that the equality constraint $\boldsymbol{I} \bullet \boldsymbol{X} = b$ hold strictly. For example, suppose that the feasible region of the equality standard form SDP (1.1) is bounded and that a positive number b satisfying $\boldsymbol{I} \bullet \boldsymbol{X} + 1 \leq b$ for every feasible solution $\boldsymbol{X}$ of (1.1) is available. Then, adding artificial redundant constraints $\boldsymbol{I}\bullet\boldsymbol{X}+X_{n+1,n+1} = b$ and $X_{n+1,n+1} \geq 0$ to (1.1), we can transform (1.1) into an SDP of the form (1.5). In this case, we may replace the equality constraint $\boldsymbol{I}\bullet\boldsymbol{X}+X_{n+1,n+1} = b$ by $\underline{\chi} \leq \boldsymbol{I}\bullet\boldsymbol{X}+X_{n+1,n+1} \leq \overline{\chi}$, where $b-1 < \underline{\chi} \leq \overline{\chi} < \infty$. Thus the equality constraint $\boldsymbol{I} \bullet \boldsymbol{X} + X_{n+1,n+1} = b$ is allowed to be satisfied loosely.

In general, we may regard SDP (1.5) itself as a special case of the equality standard form SDP (1.1), in which its feasible region is bounded and a positive number b' satisfying $\boldsymbol{I} \bullet \boldsymbol{X} + 1 \leq b'$ for every feasible solution $\boldsymbol{X}$ of (1.5) is available; just take $b' = b+1$. Therefore we can apply the transformation mentioned above to SDP (1.5).

We will give another good example.

LMI (Linear matrix inequality): Let $\boldsymbol{F}_p \in \mathbb{S}^\ell$ $(p = 0, 1, \ldots, m)$. Consider an LMI: $\sum_{p=1}^m \boldsymbol{F}_p y_p - \boldsymbol{F}_0 \in \mathbb{S}^\ell_+$. The LMI has a solution $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)^T \in \mathbb{R}^m$ if and only if the following SDP has a nonpositive optimal value:

$$\text{(4.1)} \qquad \begin{cases} \text{minimize} & w \\ \text{subject to} & \displaystyle\sum_{p=1}^{m} \boldsymbol{F}_p y_p + \boldsymbol{I}w - \boldsymbol{S} = \boldsymbol{F}_0, \qquad w \geq -1,\ \boldsymbol{S} \in \mathbb{S}^\ell_+. \end{cases}$$

Let $n = \ell + 1$, $a_p = 0$ $(p = 1, 2, \ldots, m)$, $b = 1$,

$$\boldsymbol{C} = \begin{pmatrix} \boldsymbol{F}_0 & \boldsymbol{0} \\ \boldsymbol{0}^T & -1 \end{pmatrix} \in \mathbb{S}^n, \quad \boldsymbol{A}_p = \begin{pmatrix} \boldsymbol{F}_p & \boldsymbol{0} \\ \boldsymbol{0}^T & 0 \end{pmatrix} \in \mathbb{S}^n \ (p = 1, 2, \ldots, m).$$

Then, we can rewrite (4.1) as our dual form SDP (1.6). In this case, we may replace the objective function $bw = w$ of the resulting SDP by $b''w$, with any $b'' > 0$; hence

the primal equality constraint $\boldsymbol{I} \bullet \boldsymbol{X} = b$ needs to be satisfied loosely in the sense $\underline{\chi} \le \boldsymbol{I} \bullet \boldsymbol{X} \le \overline{\chi}$, where $0 < \underline{\chi} \le \overline{\chi} < \infty$.

The looseness of the equality constraint $\boldsymbol{I} \bullet \boldsymbol{X} = b$ will provide our method with lots of flexibility. Theoretically, we consider a class of central trajectories $\mathcal{C}(b)$ ($b \in [\underline{\chi}, \overline{\chi}]$) characterized by (2.8). For each fixed $b \in [\underline{\chi}, \overline{\chi}]$, we know that the central trajectory $\mathcal{C}(b)$ leads to a primal-dual optimal solution of SDP (1.5), so that we may perform the predictor procedure described in section 2.3 at any point $(\boldsymbol{X}, w, \boldsymbol{S}) = (\boldsymbol{X}(\boldsymbol{y};\mu), w(\boldsymbol{y};\mu), \boldsymbol{S}(\boldsymbol{y};\mu))$ that satisfies (2.8) for any $b \in [\underline{\chi}, \overline{\chi}]$. This then allows us to loosely solve the equation $\phi(w;\boldsymbol{y},\mu) = b$, discussed in section 2.2, such that $\underline{\chi} \le \phi(w;\boldsymbol{y},\mu) \le \overline{\chi}$. This technique saves much computation time.

On the other hand, the definition of the function $g(\cdot;\mu)$ involves b, so that the function value, the gradient vector, and the Hessian matrix of $g(\cdot;\mu)$ with a fixed μ are affected by perturbations of b. Therefore, we need to fix b, or at least not perturb b much, to consistently perform the minimization of the function $g(\cdot;\mu)$ with a fixed $\mu > 0$ in our corrector procedure. In particular, a little perturbation to b may destroy the convergence of the corrector procedure. This would become more serious as μ becomes smaller. As a compromise, we take the following strategy in our numerical experiment reported in section 5. Let $\chi : \mathbb{R}_{++} \to \mathbb{R}_{++}$ be a nondecreasing continuous function; we use $\chi(\mu) = 10^{-7} + \min\{10^{-2}, 10\mu\}$ for every $\mu \in \mathbb{R}_{++}$ in our numerical experiment. We stop the iteration described in section 2.2 to compute a solution $w = w(\boldsymbol{y};\mu)$ of the equation $\phi(w;\boldsymbol{y},\mu) = b$ with a fixed $\mu > 0$ when $|\phi(w;\boldsymbol{y},\mu) - b| < \chi(\mu)$. Now suppose that the corrector procedure utilizing the Newton iteration or the BFGS iteration has successfully terminated, satisfying the stopping criterion $(\boldsymbol{y},\mu) \in N(\epsilon_c)$ or $\boldsymbol{\nabla} g(\boldsymbol{y};\mu)^T \boldsymbol{H} \boldsymbol{\nabla} g(\boldsymbol{y};\mu) \le \mu\epsilon_c$, respectively. Then we must have

$$|\boldsymbol{I} \bullet \boldsymbol{X} - b| \le \chi(\mu), \quad \boldsymbol{I}w - \boldsymbol{S} = \boldsymbol{C} - \sum_{p=1}^{m} \boldsymbol{A}_p y_p, \quad \boldsymbol{X}\boldsymbol{S} = \mu\boldsymbol{I}, \qquad \boldsymbol{X} \in \mathbb{S}^n_{++},\ \boldsymbol{S} \in \mathbb{S}^n_{++},$$
$$\boldsymbol{\nabla} g(\boldsymbol{y};\mu)^T \boldsymbol{\nabla}^2 g(\boldsymbol{y};\mu)^{-1} \boldsymbol{\nabla} g(\boldsymbol{y};\mu) \le \mu\epsilon_c \quad \text{or} \quad \boldsymbol{\nabla} g(\boldsymbol{y};\mu)^T \boldsymbol{H} \boldsymbol{\nabla} g(\boldsymbol{y};\mu) \le \mu\epsilon_c.$$

Let $b' = \boldsymbol{I} \bullet \boldsymbol{X}$. Then the resulting point $(\boldsymbol{X}, \boldsymbol{y}, w, \boldsymbol{S})$ lies (approximately) on the central trajectory $\mathcal{C}(b')$, and we can perform Step 2-N (the predictor procedure using the Newton method) or its variant (the predictor procedure using a preconditioned iterative method) described in section 3.3.

4.2. A step length based on a quadratic approximation of $g(\boldsymbol{z} + \alpha\boldsymbol{d};\mu)$ in α. For determining a step length $\alpha \in (0, 1]$ in Step 1-N or Step 1-BFGS, we propose to employ a quadratic approximation of $g(\boldsymbol{z} + \alpha\boldsymbol{d};\mu)$ in α:

$$g(\boldsymbol{z} + \alpha\boldsymbol{d};\mu) \approx g(\boldsymbol{z}) + \alpha \boldsymbol{\nabla} g(\boldsymbol{z};\mu)^T \boldsymbol{d} + \frac{\alpha^2}{2} \boldsymbol{d}^T \boldsymbol{\nabla}^2 g(\boldsymbol{z};\mu) \boldsymbol{d}. \tag{4.2}$$

Let $\alpha_{\min} = -\boldsymbol{\nabla} g(\boldsymbol{z};\mu)^T \boldsymbol{d} / \boldsymbol{d}^T \boldsymbol{\nabla}^2 g(\boldsymbol{z};\mu) \boldsymbol{d}$ be the minimizer of the quadratic function. Then choose a step length $\alpha = \min\{\alpha_{\min}, 1.0\}$. In the case of the DIPM method, we need to shorten the step length α when $\boldsymbol{z} + \alpha\boldsymbol{d} \notin \mathcal{Y}_{++}$. In our numerical experiment, which we will report in section 5, we multiply the step length α by a constant ratio $\tau = 0.8$ iteratively until $\boldsymbol{z} + \alpha\boldsymbol{d} \in \mathcal{Y}_{++}$ holds. τ was chosen empirically.

We tried Armijo's and Wolfe's line search rules in our numerical experiments. We found, however, that either of the rules often leads us into a jam or makes the BFGS quasi-Newton method not converge. One major reason might be that inaccuracies occur in the computation of the value of the merit function $g(\cdot;\mu)$. In particular, when

we employed a loose solution w of the equation $\phi(w; \boldsymbol{y}, \mu) = b$ as described in section 4.1, they did not work effectively at all. On the other hand, the step length based on the one-step quadratic approximation method without any line search, mentioned above, worked well in both the DIPM and the LDIPM.

4.3. One additional primal-dual interior-point method iteration to increase the accuracy. Suppose that the LDIPM using the BFGS quasi-Newton method as the corrector procedure results in approximate optimal solutions $\boldsymbol{X}^\ell \in \mathbb{S}^n_{++}$ of the primal SDP (1.5), and $(\boldsymbol{y}^\ell, w^\ell, \boldsymbol{S}^\ell) \in \mathbb{R}^{m+1} \times \mathbb{S}^n_{++}$ of the dual SDP (1.6), at the ℓth iteration. We will show how we perform one iteration of the primal-dual interior-point method using the HRVW/KSH/M search direction [14, 17, 19] to get approximate optimal solutions with a higher accuracy. We compute a search direction $(\boldsymbol{dX}, \boldsymbol{dy}, dw, \boldsymbol{dS})$ by solving

$$\begin{aligned}
&\boldsymbol{A}_p \bullet (\boldsymbol{X}^\ell + \boldsymbol{dX}) = a_p \ (p = 1, 2, \ldots, m), \qquad \boldsymbol{I} \bullet (\boldsymbol{X}^\ell + \boldsymbol{dX}) = b,\\
&\sum_{p=1}^m \boldsymbol{A}_p(y_p^\ell + dy_p) + \boldsymbol{I}(w^\ell + dw) - (\boldsymbol{S}^\ell + \boldsymbol{dS}) = \boldsymbol{C},\\
&\boldsymbol{X}^\ell \boldsymbol{dS} + \boldsymbol{dX}\boldsymbol{S}^\ell = \beta\mu^\ell \boldsymbol{I} - \boldsymbol{X}^\ell \boldsymbol{S}^\ell,
\end{aligned}$$

where $\beta \in [0, 1]$ denotes a centering parameter. Since $(\boldsymbol{y}^\ell, w^\ell, \boldsymbol{S}^\ell)$ is an interior feasible solution of SDP (1.6) and $\boldsymbol{X}^\ell \boldsymbol{S}^\ell = \mu^\ell \boldsymbol{I}$ holds for some $\mu^\ell \in \mathbb{R}_{++}$, we obtain

$$\begin{aligned}
\boldsymbol{dS} &= \sum_{p=1}^m \boldsymbol{A}_p dy_p + \boldsymbol{I} dw,\\
\boldsymbol{dX} &= (\beta - 1)\mu^\ell (\boldsymbol{S}^\ell)^{-1} - \mu^\ell (\boldsymbol{S}^\ell)^{-1} \left(\sum_{p=1}^m \boldsymbol{A}_p dy_p + \boldsymbol{I} dw \right) (\boldsymbol{S}^\ell)^{-1},
\end{aligned}$$

$$\begin{aligned}
\nabla^2 g(\boldsymbol{y}^\ell; \mu^\ell)\boldsymbol{dy} &= \left(\boldsymbol{M} - \frac{\boldsymbol{h}\boldsymbol{h}^T}{h_{m+1}} \right) \boldsymbol{dy} = \boldsymbol{\rho} - \frac{\rho_{m+1}}{h_{m+1}} \boldsymbol{h},\\
h_{m+1} dw &= \rho_{m+1} - \boldsymbol{h}^T \boldsymbol{dy}.
\end{aligned} \tag{4.3}$$

Here $\boldsymbol{M} \in \mathbb{S}^m_{++}$, $\boldsymbol{h} \in \mathbb{R}^m$, and $h_{m+1} \in \mathbb{R}$ are given by (2.4) with $\boldsymbol{S} = \boldsymbol{S}^\ell$, and

$$\begin{aligned}
\rho_p &= \beta\mu^\ell \boldsymbol{A}_p \bullet (\boldsymbol{S}^\ell)^{-1} - a_p \in \mathbb{R} \ \ (p = 1, 2, \ldots, m),\\
\rho_{m+1} &= \beta\mu^\ell \boldsymbol{I} \bullet (\boldsymbol{S}^\ell)^{-1} - b \in \mathbb{R}, \quad \boldsymbol{\rho} = (\rho_1, \rho_2, \ldots, \rho_m)^T \in \mathbb{R}^m.
\end{aligned}$$

We apply the CG method to system (4.3), utilizing the BFGS quasi-Newton matrix $\boldsymbol{H}^\ell$ obtained at the ℓth iteration as an effective preconditioning matrix. We stop the CG iteration when an approximate solution $\boldsymbol{dy}$ satisfying

$$\left(\nabla^2 g(\boldsymbol{y}^\ell; \mu^\ell)\boldsymbol{dy} - \boldsymbol{\rho} + \frac{\rho_{m+1}}{h_{m+1}} \boldsymbol{h} \right)^T \boldsymbol{H}^\ell \left(\nabla^2 g(\boldsymbol{y}^\ell; \mu^\ell)\boldsymbol{dy} - \boldsymbol{\rho} + \frac{\rho_{m+1}}{h_{m+1}} \boldsymbol{h} \right) < \mu\epsilon_{pd} \tag{4.4}$$

is obtained. Here $\epsilon_{pd} > 0$. Note that this stopping criterion is compatible with the ones we have employed so far for the BFGS quasi-Newton iteration in the corrector procedure and for the CG method to compute the predictor directions.

The dual step length $\alpha_d = \max\{\alpha \in [0, 1] : \ \boldsymbol{S}^\ell + \alpha \boldsymbol{dS} \in \mathbb{S}^n_+\}$, the primal step length $\alpha_p = \max\{\alpha \in [0, 1] : \ \hat{\boldsymbol{X}} = \boldsymbol{X}^\ell + \alpha \boldsymbol{dX} \in \mathbb{S}^n_+\}$, and the primal objective value $\boldsymbol{C} \bullet \hat{\boldsymbol{X}}$ can be computed from the sparse matrices $\boldsymbol{S}^\ell$ and $\boldsymbol{dS}$. Details are omitted here.

4.4. Exploiting sparsity. There are many places in which we can exploit the sparsity of data matrices $\boldsymbol{C}, \boldsymbol{A}_p$ $(p = 1, 2, \ldots, m)$ in the DIPM and the LDIPM. Among others, we mention the following.

(I) *A sparse Cholesky factorization* $\boldsymbol{NN^T}$ *of the dual matrix variable* $\boldsymbol{S}$. We use a Cholesky factorization $\boldsymbol{NN}^T$ of the dual matrix variable $\boldsymbol{S}$ when we evaluate the value of the function $\phi(\cdot; \boldsymbol{y}, \mu)$ and its derivative to compute an approximate solution of (1.8). We can apply various existing heuristic methods, such as the minimum degree ordering for less fill-in, the (nested) dissection ordering for less fill-in, and the reverse Cuthill–McKee ordering for reducing bandwidth [11]. More generally, we can handle a case in which $\boldsymbol{S}$ is the sum of low rank (dense) matrices in $\mathbb{S}^n$ and a matrix that allows a sparse Cholesky factorization. Such a case appears in the SDP relaxation of the graph equibisection problem. Suppose that $\boldsymbol{S} \in \mathbb{S}_{++}$ has "a sparse and rank-ℓ-dense structure":

$$\boldsymbol{S} = \boldsymbol{S}_0 + \sum_{i=1}^{\ell} \boldsymbol{q}_i \boldsymbol{t}_i^T \in \mathbb{S}^n_{++} \tag{4.5}$$

for some ℓ, $\boldsymbol{q}_j$, $\boldsymbol{t}_j \in \mathbb{R}^n$ $(j = 1, 2, \ldots, \ell)$, and some sparse matrix $\boldsymbol{S}_0$ that allows a sparse Cholesky factorization $\boldsymbol{N}_0 \boldsymbol{N}_0^T$. Define $\boldsymbol{S}_j = \boldsymbol{S}_0 + \sum_{i=1}^{j} \boldsymbol{q}_i \boldsymbol{t}_i^T$ $(j = 1, 2, \ldots, \ell)$. Then, applying the Sherman–Morrison formula recursively to each $\boldsymbol{S}_j$ $(j = 1, 2, \ldots, \ell)$, we see that

$$\boldsymbol{S}_{j+1}^{-1} = \boldsymbol{S}_j^{-1} - \frac{\boldsymbol{S}_j^{-1} \boldsymbol{q}_{j+1} \boldsymbol{t}_{j+1}^T \boldsymbol{S}_j^{-1}}{1 + \boldsymbol{t}_{j+1}^T \boldsymbol{S}_j^{-1} \boldsymbol{q}_{j+1}} = \boldsymbol{S}_j^{-1} + \boldsymbol{u}_{j+1} \boldsymbol{v}_{j+1}^T,$$

where

$$\boldsymbol{u}_{j+1} = -\frac{\boldsymbol{S}_j^{-1} \boldsymbol{q}_{j+1}}{1 + \boldsymbol{t}_{j+1}^T \boldsymbol{S}_j^{-1} \boldsymbol{q}_{j+1}} \quad \text{and} \quad \boldsymbol{v}_{j+1} = \boldsymbol{S}_j^{-1} \boldsymbol{t}_{j+1}.$$

Consequently, we obtain that

$$\boldsymbol{S}^{-1} = \boldsymbol{N}_0^{-T} \boldsymbol{N}_0^{-1} + \sum_{i=1}^{\ell} \boldsymbol{u}_i \boldsymbol{v}_i^T. \tag{4.6}$$

We will call the formula (4.6) a *Cholesky and rank-ℓ-factorization* of $\boldsymbol{S}^{-1}$. Note that a multiplication of $\boldsymbol{S}^{-1}$ by a vector $\boldsymbol{\omega} \in \mathbb{R}^n$ is now reduced to the following procedure.

- Let $\boldsymbol{z}_1 = \sum_{i=1}^{\ell} \boldsymbol{u}_i \boldsymbol{v}_i^T \boldsymbol{\omega}$.
- Solve $\boldsymbol{N}_0 \boldsymbol{z}_2 = \boldsymbol{\omega}$ in $\boldsymbol{z}_2 \in \mathbb{R}^n$.
- Solve $\boldsymbol{N}_0^T \boldsymbol{z}_3 = \boldsymbol{z}_2$ in $\boldsymbol{z}_3 \in \mathbb{R}^n$.
- Let $\boldsymbol{S}^{-1} \boldsymbol{\omega} = \boldsymbol{z}_1 + \boldsymbol{z}_3$.

(II) *Use of the Cholesky and rank-ℓ-factorization* (4.6) *of* $\boldsymbol{S}^{-1}$. Suppose that a Cholesky and rank-ℓ-factorization (4.6) of the inverse $\boldsymbol{S}^{-1}$ of a dual matrix variable $\boldsymbol{S}$ of the form (4.5) is available at some iteration. Recall that the identity $\boldsymbol{XS} = \mu \boldsymbol{I}$ holds. Therefore, if we maintain the vectors $\boldsymbol{q}_j, \boldsymbol{t}_j, \boldsymbol{u}_j, \boldsymbol{v}_j \in \mathbb{R}^n$ $(j = 1, 2, \ldots, \ell)$ and a sparse lower triangular matrix $\boldsymbol{N}_0$, not only $\boldsymbol{S}$ but also $\boldsymbol{X} = \mu \boldsymbol{S}^{-1}$ is restored at any time, saving a significant amount of memory. We should also mention that all of the computations in our method can be carried out without restoring the dense matrix $\boldsymbol{X}$ from those vectors and matrices explicitly. We have mentioned above a procedure

for computing $\boldsymbol{S}^{-1}\boldsymbol{\omega}$ for a given $\boldsymbol{\omega} \in \mathbb{R}^n$. Given $\boldsymbol{F}_j \in \mathbb{S}^n$ $(j = 1, 2, \ldots, q)$, we can apply that procedure to the computation of

$$\text{Trace } \boldsymbol{F}_1\boldsymbol{S}^{-1}\boldsymbol{F}_2\boldsymbol{S}^{-1}\cdots\boldsymbol{F}_q\boldsymbol{S}^{-1} = \sum_{i=1}^{q} \boldsymbol{e}_i^T\boldsymbol{F}_1\boldsymbol{S}^{-1}\boldsymbol{F}_2\boldsymbol{S}^{-1}\cdots\boldsymbol{F}_q\boldsymbol{S}^{-1}\boldsymbol{e}_i.$$

Here $\boldsymbol{e}_i$ denotes the ith unit coordinate vector in $\mathbb{R}^n$. Assuming that $\boldsymbol{S}^{-1}\boldsymbol{\omega}$ can be done in $O(n^2)$ arithmetic operations for any $\boldsymbol{\omega} \in \mathbb{R}^n$, the computation of the trace of matrices above requires $O(qn^2)$ arithmetic operations. In most of the computation involving $\boldsymbol{X}$ and $\boldsymbol{S}^{-1}$ in the LDIPM, q is either 1 or 2. The unique exceptional case is the computation of r_p $(p = 1, 2, \ldots, m+1)$ involved in the right-hand side of the equation on the second-order derivative $\ddot{\boldsymbol{y}}$ in (2.9). (See (2.10).) In this case, q turns out to be 3. The approximation of the second derivative is considered to be one of the most expensive parts in the LDIPM, even when we use the CG or CR method.

4.5. The limited memory BFGS quasi-Newton method. When the dimension m of the dual variable vector $\boldsymbol{y}$ is large, say more than several thousand, it is usually impossible to store the entire $m \times m$ BFGS matrix $\boldsymbol{H}$ in standard workstations. In such cases, we may replace the full BFGS quasi-Newton matrix used in Step 1-BFGS by the limited memory BFGS update. In their recent paper [20], Morales and Nocedal proposed using the limited memory BFGS quasi-Newton matrix to precondition the CG method. In our predictor procedure described in section 3.3, we can employ the limited memory BFGS quasi-Newton matrix to precondition the CG method (or the CR method).

5. Numerical results. We wrote MATLAB codes for the LDIPM and the DIPM. Each of the methods has four variants, using either of the Newton method or the BFGS quasi-Newton method as a corrector procedure, and using either of the first-order or the second-order predictor procedures. The primary purposes here are to verify that all variants work numerically and to investigate the effectiveness of the corrector procedure using the BFGS quasi-Newton method, and of the predictor procedure using the CG method preconditioned by the BFGS quasi-Newton matrix. All of the numerical experiments were done using MATLAB Version 5.2 on a Macintosh Power PC 750 running at 400MHz with 360 MB memory. In this implementation, we used two MATLAB matrix types, the standard dense matrix (two-dimensional array) type and the MATLAB sparse matrix type, to cope with both dense and sparse data matrices $\boldsymbol{C}, \boldsymbol{A}_p$ $(p = 1, 2, \ldots, m)$. Computation of the matrices $\sum_{p=1}^{m} \boldsymbol{A}_p y_p$ and $\boldsymbol{C} - \sum_{p=1}^{m} \boldsymbol{A}_p y_p$ was done via appropriate matrix types, depending on their sparsity. This saved a considerable amount of computational time. However, we employed neither of the sparsity techniques (I) and (II) mentioned in section 4.4 because simple and/or efficient MATLAB implementation of such techniques is difficult. In particular, we computed and maintained the primal matrix variable $\boldsymbol{X}$ throughout the iterations.

5.1. Test problems. We consider three kinds of SDPs, an SDP relaxation of a box constrained quadratic 0-1 program, a norm minimization problem, and a linear matrix inequality.

*BQ*01*IP* (Box constrained quadratic 0-1 integer program). Let $\boldsymbol{Q} \in \mathbb{S}^\ell$ be a matrix whose components are chosen from random numbers uniformly distributed in the interval $(0.0, 1.0)$. Consider the problem:

$$\text{maximize} \quad \boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} \quad \text{subject to} \quad x_i = -1 \ \text{ or } 1 \ (i = 1, 2, \ldots, \ell).$$

A upper bound of the objective function value can be computed by solving an SDP relaxation of this problem:

$$\begin{cases} \text{maximize} & \boldsymbol{Q} \bullet \boldsymbol{X} \\ \text{subject to} & X_{ii} = 1 \ (i = 1, 2, \ldots, \ell), \ \boldsymbol{I} \bullet \boldsymbol{X} \leq \ell + 1, \ \boldsymbol{X} \in \mathbb{S}^{\ell}_{+}. \end{cases}$$

The inequality constraint $\boldsymbol{I} \bullet \boldsymbol{X} \leq \ell+1$ is redundant but is added to transform the SDP into our standard form SDP (1.5). Let $n = \ell + 1$, $m = \ell$, $a_p = 1.0$ $(p = 1, 2, \ldots, m)$, $b = \ell + 1$, and

$$\boldsymbol{C} = \begin{pmatrix} \boldsymbol{Q} & \boldsymbol{0} \\ \boldsymbol{0}^T & 0 \end{pmatrix} \in \mathbb{S}^n, \qquad \begin{array}{l} \boldsymbol{A}_p = \text{the } n \times n \text{ matrix with the } p\text{th diagonal component} \\ \qquad\quad 1 \text{ and all other components } 0 \ (p = 1, 2, \ldots, m). \end{array}$$

Then we can rewrite the SDP above in our standard form (1.5). We used the MATLAB dense matrix type for $\boldsymbol{C}$, and the MATLAB sparse matrix type for $\boldsymbol{A}_p$ $(p = 1, 2, \ldots, m)$. Since it is difficult to guess good initial points, the following tentative initial points were chosen:

$$(5.1) \quad \begin{cases} \boldsymbol{y}^0 = \boldsymbol{0} \in \mathbb{R}^m, \ \mu^0 = \sqrt{m} & \text{for the LDIPM,} \\ (\boldsymbol{y}^0, w^0) = (\boldsymbol{0}, \lambda_{\max} + m + 1) \in \mathbb{R}^{m+1}, \ \mu^0 = \sqrt{m} & \text{for the DIPM,} \end{cases}$$

where $\lambda_{\max}$ denotes the largest eigenvalue of $\boldsymbol{C}$.

NMIN (Norm minimization problem). Let $\boldsymbol{F}_p \in \mathbb{R}^{q \times r}$ $(p = 0, 1, \ldots, m)$. We consider the problem of minimizing $\|\boldsymbol{F}_0 - \sum_{p=1}^{m} y_p \boldsymbol{F}_p\|$ in a vector variable $\boldsymbol{y} \in \mathbb{R}^m$. Here $\|\boldsymbol{F}\|$ denotes the 2-norm of $\boldsymbol{F}$. We can reformulate this problem as an SDP:

$$\text{minimize} \quad w \quad \text{subject to} \quad \sum_{p=1}^{m} \boldsymbol{A}_p y_p + \boldsymbol{I} w - \boldsymbol{S} = \boldsymbol{C}, \quad \boldsymbol{S} \in \mathbb{S}^n_{+}.$$

Here

$$n = q + r, \quad \boldsymbol{C} = \begin{pmatrix} \boldsymbol{O} & \boldsymbol{F}_0^T \\ \boldsymbol{F}_0 & \boldsymbol{O} \end{pmatrix}, \quad \boldsymbol{A}_p = \begin{pmatrix} \boldsymbol{O} & \boldsymbol{F}_p^T \\ \boldsymbol{F}_p & \boldsymbol{O} \end{pmatrix} \ (p = 1, 2, \ldots, m).$$

If we define $a_p = 0$ $(p = 1, 2, \ldots, m)$ and $b = 1$, then the SDP above turns out to be our dual form SDP (1.6). In our numerical experiments, we assigned a random number uniformly distributed in the interval $(0.0, 1.0)$ to each component of $\boldsymbol{F}_p$ $(p = 0, 1, \ldots, m)$. Each matrix $\boldsymbol{C}, \boldsymbol{A}_p$ $(p = 1, 2, \ldots, m)$ involves $2qr$ nonzeros and $q^2 + r^2$ zeros. We used the MATLAB sparse matrix type for $\boldsymbol{C}, \boldsymbol{A}_p$ $(p = 1, 2, \ldots, m)$, and initial points were chosen as in (5.1).

LMI (Linear matrix inequality), given in section 4.1. In our numerical experiments, we generated two types of sparse data matrices with nonzero element densities $d = 0.04$ and 0.2. For each $d = 0.04$ and 0.2, we assigned to each element of $\boldsymbol{F}_p$ $(p = 0, 1, \ldots, m)$ a random number uniformly distributed in the interval $(0.0, 1.0)$ with the probability d, and 0.0 with the probability $1.0 - d$; hence each matrix $\boldsymbol{F}_p$ $(p = 1, 2, \ldots, m)$ is expected to have $n^2 d$ nonzero elements. We used the MATLAB sparse matrix type for $\boldsymbol{A}_p$ $(p = 0, 1, \ldots, m)$ in both cases, and initial points were chosen as in (5.1).

5.2. Parameters, accuracy, and stopping criteria. We tried various values for the parameters ϵ_p (the tolerance used in the predictor procedure; see sections 3.1 and 3.3), ϵ_c (the tolerance used in the corrector procedures; see sections 3.1 and 3.2), γ

TABLE 5.1

Change in the condition numbers of the Hessian matrix and of the preconditioned Hessian matrix in the case of the SDP relaxation of the BQ01IP with $n = 201$, $m = 200$. Here $\nabla^2 g^k = \nabla^2 g(\boldsymbol{y}^k; \mu^k)$.

				Condition number			
k	μ^k	p.f.error	rel.error	$\nabla^2 g^k$	$\boldsymbol{H}^k \nabla^2 g^k$	CG1	CG2
1	1.42e+1	2.70e−2	2.82e+1	5.18e+4	3.44e+3	5	2
2	6.64e+0	4.67e−2	5.63e+0	4.09e+4	1.11e+3	7	6
3	4.07e+0	4.72e−3	2.10e+0	7.58e+4	1.13e+3	10	8
4	2.12e+0	4.60e−2	6.77e−1	2.48e+5	5.14e+2	12	4
5	1.11e+0	8.79e−2	2.78e−1	6.01e+5	8.70e+2	21	8
6	4.54e−1	6.73e−3	9.89e−2	9.43e+5	2.59e+1	15	7
7	8.62e−2	4.03e−3	1.75e−2	1.29e+6	5.14e+1	17	5
8	1.64e−2	4.41e−4	3.28e−3	1.49e+6	1.50e+2	20	5
9	1.64e−3	7.47e−5	3.27e−4	1.51e+7	1.92e+2	14	3
10	1.64e−4	1.93e−7	3.27e−5	1.52e+8	3.24e+1	7	1
11	1.64e−5	5.26e−7	3.26e−6	1.52e+9	3.21e+2	8	0
12	1.64e−6	5.31e−7	3.22e−7	1.52e+10	3.20e+2	-	-
				Average iterations		12.4	4.5

(the reduction factor used for the barrier parameter μ in the predictor procedure; see section 3.1), and ϵ_{cg} (the tolerance used in the CG method; see section 3.3) and then determined the following values for them: $\epsilon_p = 1.0$, $\epsilon_c = 0.01$, $\gamma = 0.9$, $\epsilon_{cg} = 0.001$ for the first-order derivative, and $\epsilon_{cg} = 0.01$ for the second-order derivative. Also we used the one-step quadratic approximation method, mentioned in section 4.2, to choose a step length in the corrector procedure along a search direction generated by either the Newton method or the quasi-Newton BFGS method.

The following symbols are used in the numerical experiments of the next subsections:

$$\text{rel.error} = \text{the relative error } = \frac{\left|\sum_{p=1}^m a_p y_p^k + b w^k - \boldsymbol{C} \bullet \boldsymbol{X}^k\right|}{\max\{|\boldsymbol{C} \bullet \boldsymbol{X}^k|,\ 1.0\}},$$

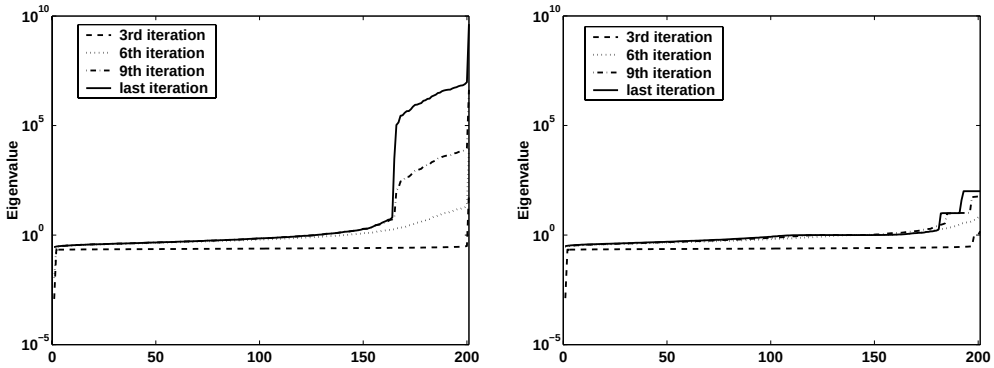
$$\text{p.f.error} = \text{the primal feasibility error}$$
$$= \max\left\{|a_p - \boldsymbol{A}_p \bullet \boldsymbol{X}^k| : p = 1, 2, \ldots, m\right\}.$$

Here k denotes the last iteration.

In the numerical experiments which we report next, we stopped the iteration when the relative error and the primal feasibility error both became less than 10^{-6}. In all cases, we succeeded in generating approximate optimal solutions with this accuracy, but more sophisticated implementations are required to compute higher accuracy optimal solutions.

5.3. The condition number of $\nabla^2 g(\boldsymbol{y}^k; \mu^k)$ vs. the condition number of $\boldsymbol{H}^k \nabla^2 g(\boldsymbol{y}^k; \mu^k)$. Table 5.1 shows how the values of

- the condition number of $\nabla^2 g(\boldsymbol{y}^k; \mu^k)$,
- the condition number of $\boldsymbol{H}^k \nabla^2 g(\boldsymbol{y}^k; \mu^k)$, where $\boldsymbol{H}^k$ denotes the quasi-Newton BFGS matrix (see section 3.3),
- "CG1," the number of iterations in the CG method with the initial point $\boldsymbol{H}^k \bar{\boldsymbol{a}}$ for approximating the first derivative $\dot{\boldsymbol{y}}(\mu^k)$ (see (2.9) for the definition of $\bar{\boldsymbol{a}}$), and
- "CG2," the number of iterations in the CG method with the initial point $\boldsymbol{H}^k \bar{\boldsymbol{r}}$



(a) *Spectrum of* $\nabla^2 g(\boldsymbol{y}^k;\mu^k)$. (b) *Spectrum of* $\boldsymbol{H}^k\nabla^2 g(\boldsymbol{y}^k;\mu^k)$.

FIG. 5.1. *Spectra of the Hessian matrix* $\nabla^2 g(\boldsymbol{y}^k;\mu^k)$ *and of the preconditioned Hessian matrix* $\boldsymbol{H}^k\nabla^2 g(\boldsymbol{y}^k;\mu^k)$ *at the* 3*rd,* 6*th,* 9*th, and* 12*th iterations for the BQ*01*IP in Table* 5.1.

for approximating the second derivative $\ddot{\boldsymbol{y}}(\mu^k)$ (see (2.9) for the definition of $\bar{\boldsymbol{r}}$)

change along the sequence $\{(\boldsymbol{y}^k,\mu^k)\}$ generated by the BFGS second-order predictor variant of the LDIPM applied to the instance of the SDP relaxation of the BQ01IP with $n = 201$ and $m = 200$. We see that the condition number of $\nabla^2 g(\boldsymbol{y}^k;\mu^k)$ got worse rapidly after iteration 8 or after the barrier parameter μ^k became smaller than 10^{-2}. But the condition number of the preconditioned matrix $\boldsymbol{H}^k\nabla^2 g(\boldsymbol{y}^k;\mu^k)$ remained small, and there was no increasing tendency in the number of iterations of the CG method in the predictor procedure. Figure 5.1(a) shows the spectrum of $\nabla^2 g(\boldsymbol{y}^k;\mu^k)$, and Figure 5.1(b) shows the spectrum of $\boldsymbol{H}^k\nabla^2 g(\boldsymbol{y}^k;\mu^k)$ at the 3rd, 6th, 9th, and 12th iterations, respectively. We can also observe from these figures that the preconditioned matrix $\boldsymbol{H}^k\nabla^2 g(\boldsymbol{y}^k;\mu^k)$ has a narrower range for its spectrum than the Hessian matrix $\nabla^2 g(\boldsymbol{y}^k;\mu^k)$, which is a favorable fact for CG methods. We conclude that it is advantageous to precondition the Hessian matrix $\nabla^2 g(\boldsymbol{y}^k;\mu^k)$ by $\boldsymbol{H}^k$.

5.4. The LDIPM vs. the DIPM. Table 5.2 shows numerical results on the SDP relaxation of the BQ01IP with $n = 101,\ 201$; Table 5.3, numerical results on NMIN with $q = 20$, $r = 30$ $(n = 50)$, $m = 100,\ 200$; and Table 5.4, numerical results on LMI with $n = 50$, $m = 200$, $d = 0.04,\ 0.2$. For each case, five problems are generated randomly in the ways mentioned in section 5.1. We use the following symbols for the average or geometric mean over the five problems:

Iter. = average number of iterations,
New. = average number of Newton iterations,
BFGS = average number of BFGS updates,
CG = average number of CG iterations,
Chol. = average number of Cholesky factorizations of $\boldsymbol{S}$,
κ = the geometric mean of the condition numbers of the Hessian matrices of $g(\cdot;\mu^k)$ at the last iterates,

TABLE 5.2
SDP relaxation of BQ01IP.

$n = 101$, $m = 100$									
Method	Correc./Pred.	Iter.	New.	BFGS	CG	Chol.	κ	p.κ	CPU
LDIPM	Newton/1st	12.2	24.2	-	-	210.0	3.6e+7	-	156
	Newton/2nd	10.0	17.0	-	-	115.2	2.4e+7	-	95
	BFGS/1st	11.6	-	148.8	118.6	511.4	2.2e+7	3.5e+1	67
	BFGS/2nd	10.0	-	136.2	121.8	412.0	2.8e+7	3.5e+1	56
DIPM	Newton/1st	15.4	44.2	-	-	81.6	4.6e+9	-	217
	Newton/2nd	11.2	28.4	-	-	53.0	3.8e+9	-	142
	BFGS/1st	17.0	-	181.6	186.0	237.8	4.4e+9	5.3e+1	79
	BFGS/2nd	11.6	-	162.0	129.6	187.2	7.8e+9	5.4e+1	60
$n = 201$, $m = 200$									
Method	Correc./Pred.	Iter.	New.	BFGS	CG	Chol.	κ	p.κ	CPU
LDIPM	Newton/1st	13.4	27.0	-	-	285.4	6.2e+7	-	3252
	Newton/2nd	10.8	19.6	-	-	165.8	3.4e+7	-	1529
	BFGS/1st	12.6	-	201.2	188.4	795.8	2.7e+7	7.8e+1	763
	BFGS/2nd	10.2	-	180.0	177.2	567.8	2.2e+7	8.6e+1	585
DIPM	Newton/1st	17.8	53.4	-	-	104.6	1.7e+10	-	3630
	Newton/2nd	12.2	32.2	-	-	63.4	1.8e+10	-	2217
	BFGS/1st	18.8	-	245.8	274.4	313.0	1.0e+10	1.2e+2	982
	BFGS/2nd	12.2	-	214.8	187.2	247.6	1.8e+10	2.1e+2	699

TABLE 5.3
NMIN with $q = 20$, $r = 30$ ($n = 50$).

$m = 100$									
Method	Correc./Pred.	Iter.	New.	BFGS	CG	Chol.	κ	p.κ	CPU
LDIPM	Newton/1st	13.0	29.2	-	-	152.6	2.0e+9	-	168
	Newton/2nd	11.8	23.0	-	-	86.6	4.0e+9	-	129
	BFGS/1st	13.2	-	179.6	121.2	360.8	2.7e+9	2.7e+2	58
	BFGS/2nd	12.0	-	162.0	141.0	273.4	3.3e+9	2.5e+2	53
DIPM	Newton/1st	16.4	42.4	-	-	79.4	1.9e+9	-	220
	Newton/2nd	13.0	28.2	-	-	49.4	2.8e+9	-	138
	BFGS/1st	18.2	-	246.8	203.0	300.4	3.5e+9	2.4e+2	89
	BFGS/2nd	13.4	-	201.8	188.4	229.4	1.6e+9	4.1e+2	69
$m = 200$									
Method	Correc./Pred.	Iter.	New.	BFGS	CG	Chol.	κ	p.κ	CPU
LDIPM	Newton/1st	14.8	39.2	-	-	198.6	7.8e+9	-	843
	Newton/2nd	12.6	28.0	-	-	107.8	9.2e+9	-	544
	BFGS/1st	14.2	-	340.0	228.2	608.2	4.8e+9	3.3e+2	240
	BFGS/2nd	12.6	-	319.8	262.2	509.4	1.2e+10	1.7e+3	210
DIPM	Newton/1st	16.2	41.8	-	-	81.0	8.8e+9	-	854
	Newton/2nd	13.0	26.8	-	-	48.8	1.2e+10	-	547
	BFGS/1st	17.4	-	379.2	335.6	428.6	1.8e+10	1.5e+3	315
	BFGS/2nd	13.6	-	332.8	298.6	359.2	1.5e+10	2.7e+3	244

p.κ = the geometric mean of the condition numbers of the preconditioned Hessian matrices of $g(\cdot; \mu^k)$ at the last iterates,

CPU = average CPU time in seconds.

In all cases in which either the Newton method or the BFGS method was used in the corrector procedure, and either the first-order or the second-order was used in the predictor procedure, the number of iterations, the number of the Newton iterations, the number of BFGS quasi-Newton iterations, and the CPU time required for the LDIPM are less than those for the DIPM method. In particular, the differences are larger in the first-order predictor cases. From these observations, we may conclude

TABLE 5.4
LMI with $n = 50$, $m = 200$.

Density: $d = 0.04$									
Method	Correc./Pred.	Iter.	New.	BFGS	CG	Chol.	κ	p.κ	CPU
LDIPM	Newton/1st	15.4	39.4	-	-	215.0	2.1e+8	-	131
	Newton/2nd	13.4	28.4	-	-	118.8	4.0e+8	-	86
	BFGS/1st	15.6	-	424.4	258.6	756.2	3.4e+8	1.8e+3	65
	BFGS/2nd	13.0	-	381.6	265.6	594.4	3.6e+8	2.5e+3	56
DIPM	Newton/1st	17.4	47.2	-	-	89.8	2.9e+8	-	138
	Newton/2nd	13.8	32.4	-	-	56.2	4.7e+8	-	86
	BFGS/1st	19.0	-	449.0	364.8	504.4	2.9e+8	2.2e+3	69
	BFGS/2nd	14.0	-	396.2	303.8	423.0	3.0e+8	2.4e+3	59
Density: $d = 0.2$									
Method	Correc./Pred.	Iter.	New.	BFGS	CG	Chol.	κ	p.κ	CPU
LDIPM	Newton/1st	14.8	41.4	-	-	200.0	3.8e+8	-	387
	Newton/2nd	12.6	29.0	-	-	106.2	3.1e+8	-	239
	BFGS/1st	14.6	-	389.4	257.0	680.4	2.2e+8	3.9e+2	130
	BFGS/2nd	12.8	-	362.2	287.6	547.8	3.1e+8	2.3e+3	116
DIPM	Newton/1st	16.6	46.8	-	-	87.0	3.1e+8	-	391
	Newton/2nd	13.0	31.6	-	-	55.0	2.2e+8	-	248
	BFGS/1st	18.0	-	424.4	354.8	477.6	3.2e+8	7.2e+2	153
	BFGS/2nd	13.6	-	373.8	328.8	400.6	4.0e+8	1.5e+3	125

that the log barrier function $g(\cdot;\mu)$ in the LDIPM behaves "less nonlinearly" than the log barrier function $\tilde{g}(\cdot;\mu)$ in the DIPM. We must mention, however, that the number of Cholesky factorizations required for the LDIPM method is a few times larger than the number required for the DIPM. This is because the evaluations of the function $g(\cdot;\mu)$, its gradient, and its Hessian matrix in the LDIPM generally require more than one Cholesky factorization, while the function $\tilde{g}(\cdot;\mu)$ in the DIPM is defined in terms of an explicit formula (1.3), and its evaluation requires only one Cholesky factorization.

5.5. The Newton variants vs. the BFGS variants. The BFGS variants and the Newton variants of the LDIPM worked quite similarly in the number of iterations, while the BFGS variants of the DIPM required a few more iterations than the corresponding Newton variants. This may be also explained by the observation that the log barrier function $g(\cdot;\mu)$ of the LDIPM is less nonlinear than the log barrier function $\tilde{g}(\cdot;\mu)$ of the DIPM.

In general, one iteration of the Newton method, one iteration of the quasi-Newton BFGS method, and one iteration of the CG method require $O(m^3)$, $O(m^2)$, and $O(m^2)$ arithmetic operations, respectively. Also each computation of the Hessian matrices $\boldsymbol{\nabla}^2 g(\boldsymbol{y};\mu)$ and $\boldsymbol{\nabla}^2 \tilde{g}(\boldsymbol{y};\mu)$ requires $O(m^2n^2 + mn^3)$ arithmetic operations. Although "CG" and "BFGS" of the BFGS variants are larger than "New." of the Newton variants in all cases, the total arithmetic operations required for the BFGS and the CG methods are much smaller than those required for the Newton method. However, we have to pay particular attention to a critical difference in the "Chol." column, which shows the total number of the Cholesky factorization of the dual matrix variable $\boldsymbol{S} = \boldsymbol{I}w - \boldsymbol{C} + \sum_{p=1}^{m} \boldsymbol{A}_p y_p$ computed from a given $(\boldsymbol{y}, w)$, required in the Newton variant and the BFGS variant. In dense computation, one computation of $\boldsymbol{S}$ and its Cholesky factorization require $O(mn^2 + n^3)$ arithmetic operations. When m is as large as n, the amount of work required for these computations can be the most time consuming part of the BFGS variant.

TABLE 5.5
Effectiveness of one additional primal-dual interior-point iteration.

		LDIPM		
Problem	Corrector Predictor One add.it.	BFGS 1st-order -	BFGS 1st-order CG method	BFGS 1st-order Cholesky factorization
BQ01IP	p.f.error	8.6e−7	9.1e−9	7.1e−10
$n = 201$	rel.error	3.7e−7	5.1e−10	5.2e−10
$m = 200$	CG	-	29.8	-
NMIN	p.f.error	6.4e−7	1.9e−9	2.9e−9
$n = 50$	rel.error	6.1e−7	6.1e−10	1.6e−9
$m = 200$	CG	-	51.0	-
LMI (density: $d = 0.2$)	p.f.error	7.7e−7	1.3e−9	6.2e−10
$n = 50$	rel.error	3.1e−7	6.4e−10	3.6e−10
$m = 200$	CG	-	53.6	-
LMI (density: $d = 0.04$)	p.f.error	8.5e−7	5.4e−9	5.1e−9
$n = 50$	rel.error	3.8e−7	2.3e−9	2.5e−9
$m = 200$	CG	-	62.8	-

5.6. The first-order predictor vs. the second-order predictor. In this MATLAB implementation, the BFGS first-order predictor variant performed a little worse than the BFGS second-order predictor variant in terms of the CPU time. In a more serious implementation using compiler languages like C and C++, the BFGS first-order predictor variant may turn out to be more efficient because the computation of the second-order derivative is expected to be more expensive than that of the first-order derivative. See (II) of section 4.4.

5.7. Sparsity. Table 5.4 shows numerical results on LMI with $n = 50$, $m = 200$, and nonzero element density $d = 0.04$ and 0.2 in the data matrices $\boldsymbol{F}_p$ ($p = 0, 1, \ldots, 200$). Although exploiting sparsity is implemented in a primitive level in this MATLAB code, this table indicates how important it is in the LDIPM and the DIPM. Recall that we exploited neither of the sparsity techniques (I) and (II) mentioned in section 4.4.

5.8. One additional primal-dual interior-point method iteration to increase the accuracy. Table 5.5 shows the effectiveness of the technique which we mentioned in section 4.3. We stopped the iteration of the BFGS first-order variant of the LDIPM, applied to five problems each of the four types of problems, at a $\hat{k}$th iteration when it attained a primal-dual pair of optimal solutions with the primal-feasibility accuracy and the relative error in the primal and dual objective values both less than 10^{-6} (middle column). Then we applied one additional primal-dual interior-point iteration using either of the Cholesky factorization or the CG method. We stopped the CG iteration when the stopping criterion (4.4) held for $\epsilon_{pd} = 10^{-5}$. From Table 5.5, we see that one iteration using the CG method worked as effectively as one iteration using the Cholesky factorization to get highly accurate optimal solutions.

6. Concluding remarks. Although we have reported some numerical results, we are not satisfied. The current code is written in pure MATLAB language, and thus it is very slow. It does not take enough sparsity consideration into account to efficiently solve large scale problems. Many issues remain to be studied further in working toward more practically efficient implementations for large scale problems. Among others, we need to explore the use of

- sparse Cholesky factorization of the dual matrix variable $\boldsymbol{S} = \sum_{p=1}^{m} \boldsymbol{A}_p y_p + \boldsymbol{I} w - \boldsymbol{C}$,
- the limited memory quasi-Newton BFGS method [20].

Acknowledgments. The authors would like to thank Professor Hiroshi Yabe for valuable discussions and suggestions on the quasi-Newton BFGS method. In particular, he brought the authors' attention to the recent paper [20] on the limited memory quasi-Newton BFGS method. Also the authors would like to thank Professor Kim Chuan Toh, who read the original version of the paper carefully; some parts of the revised version are much indebted to his comments and questions.

REFERENCES

[1] F. Alizadeh, J.-P. A. Haeberly, and M. L. Overton, *Primal-dual interior-point methods for semidefinite programming: Convergence rates, stability and numerical results*, SIAM J. Optim., 8 (1998), pp. 746–768.

[2] S. J. Benson, Y. Ye, and X. Zhang, *Solving large-scale sparse semidefinite programs for combinatorial optimization*, SIAM J. Optim., 10 (2000), pp. 443–461.

[3] B. Borchers, *CSDP* 2.3 *user's guide*, Optim. Methods Softw., 11-12 (1999), pp. 597–611; also available at http://www.nmt.edu/~borchers/csdp.html.

[4] S. Burer and R. D. C. Monteiro, *A nonlinear programming algorithm for solving semidefinite program via low-rank factorization*, Math. Program., to appear.

[5] S. Burer, R. D. C. Monteiro, and Y. Zhang, *Interior-Point Algorithms for Semidefinite Programming Based on a Nonlinear Programming Formulation*, TR99-27, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1999.

[6] C. Choi and Y. Ye, *Solving Sparse Semidefinite Programs Using the Dual Scaling Algorithm with an Iterative Solver*, working paper, Department of Management Sciences, The University of Iowa, Iowa City, IA, 2000.

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

[8] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Technique*, John Wiley, New York, 1968.

[9] K. Fujisawa, K. Nakata, and M. Kojima, *SDPA (SemiDefinite Programming Algorithm) User's Manual—Version* 5.00, Research report B-308, Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Tokyo, 1995 (revised August 1996); also available at ftp://ftp.is.titech.ac.jp/pub/OpRes/software/SDPA.

[10] M. Fukuda, M. Kojima, K. Murota, and K. Nakata, *Exploiting sparsity in semidefinite programming via matrix completion* I: *General framework*, SIAM J. Optim., 11 (2001), pp. 647–674.

[11] A. George and J. W. H. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice–Hall, Englewood Cliffs, NJ, 1981.

[12] C. Helmberg and F. Rendl, *Solving quadratic* $(0,1)$*-problems by semidefinite programming and cutting planes*, Math. Program., 82 (1998), pp. 291–315.

[13] C. Helmberg and F. Rendl, *A spectral bundle method for semidefinite programming*, SIAM J. Optim., 10 (2000), pp. 673–696.

[14] C. Helmberg, F. Rendl, R. J. Vanderbei, and H. Wolkowicz, *An interior-point method for semidefinite programming*, SIAM J. Optim., 6 (1996), pp. 342–361.

[15] M. Kojima, N. Megiddo, S. Mizuno, and S. Shindoh, *Horizontal and Vertical Decomposition in Interior-Point Methods for Linear Programs*, working paper, Department of Information Sciences, Tokyo Institute of Technology, Tokyo, 1993.

[16] M. Kojima, N. Megiddo, S. Mizuno, and S. Shindoh, *Decomposition in Interior-Point Methods*, Research report B-281, Department of Information Sciences, Tokyo Institute of Technology, Tokyo, 1994.

[17] M. Kojima, S. Shindoh, and S. Hara, *Interior-point methods for the monotone semidefinite linear complementarity problem in symmetric matrices*, SIAM J. Optim., 7 (1997), pp. 86–125.

[18] C.-J. Lin and R. Saigal, *An incomplete Cholesky factorization for dense symmetric positive definite matrices*, BIT, 40 (2000), pp. 536–558.

[19] R. D. C. Monteiro, *Primal-dual path-following algorithms for semidefinite programming*, SIAM J. Optim., 7 (1997), pp. 663–678.

[20] J. L. Morales and J. Nocedal, *Algorithm* 809: *PREQN: Fortran* 77 *subroutines for preconditioning the conjugate gradient method*, ACM Trans. Math. Software, 27 (2001), pp. 83–91.

[21] K. Nakata, K. Fujisawa, M. Fukuda, M. Kojima, and K. Murota, *Exploiting sparsity in semidefinite programming via matrix completion* II: *Implementation and numerical results*, Math. Program., to appear.

[22] K. Nakata, K. Fujisawa, and M. Kojima, *Using the conjugate gradient method in interior-point methods for semidefinite programs*, Proc. Inst. of Statist. Math., 46 (1998), pp. 297–316 (in Japanese).

[23] Yu. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM, Philadelphia, PA, 1994.

[24] M. Shida, *An Interior-Point Smoothing Technique for Lagrange Relaxation in Convex Programming*, working paper, Department of Mathematics, Kanagawa University, Yokohama, Japan, 1998.

[25] J. F. Sturm, *Using SeDuMi* 1.02, *a MATLAB toolbox for optimization over symmetric cones*, Optim. Methods Softw., 11–12 (1999), pp. 625–653; also available at http://fewcal.kub.nl/sturm/software/sedumi.html.

[26] M. J. Todd, K. C. Toh, and R. H. Tütüncü, *On the Nesterov-Todd direction in semidefinite programming*, SIAM J. Optim., 8 (1998), pp. 769–796.

[27] K. C. Toh and M. Kojima, *Solving some large scale semidefinite programs via the conjugate residual method*, SIAM J. Optim., 12 (2002), pp. 669–691.

[28] K. C. Toh, M. J. Todd, and R. H. Tütüncü, *SDPT*3*—a MATLAB software package for semidefinite programming, Version* 1.3, Optim. Methods Softw., 11–12 (1999), pp. 545–581; also available at http://www.math.nus.edu.sg/~mattohkc.

[29] R. J. Vanderbei and H. Y. Benson, *On Formulating Semidefinite Programming Problems as Smooth Convex Nonlinear Optimization Problems*, ORFE 99-01, Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ, 1999 (revised 2000).

[30] G. Y. Zhao, *Interior-point methods with decomposition for solving large-scale linear programs*, J. Optim. Theory Appl., 102 (1999), pp. 169–192.

[31] G. Y. Zhao, *A Log-barrier method with Benders decomposition for solving two-stage stochastic programs*, Math. Program., 90 (2001), pp. 507–536.

[32] G. Y. Zhao, *A Lagrangian Dual Method with Self-Concordant Barrier for Multi-Stage Stochastic Convex Nonlinear Programming*, working paper, Department of Mathematics, National University of Singapore, Singapore 1998 (revised 2000).

[33] Q. Zhao, *Semidefinite Programming for Assignment and Partitioning Problems*, Ph.D. thesis, University of Waterloo, Waterloo, ON, Canada, 1996.

[34] Q. Zhao, S. E. Karisch, F. Rendl, and H. Wolkowicz, *Semidefinite programming relaxations for the quadratic assignment problem*, J. Comb. Optim., 2 (1998), pp. 71–109.

SIAM J. OPTIM.
Vol. 12, No. 4, pp. 1032–1047

NECESSARY CONDITIONS FOR CONSTRAINED OPTIMIZATION PROBLEMS IN SMOOTH BANACH SPACES AND APPLICATIONS*

QIJI J. ZHU†

Abstract. We derive necessary optimality conditions for constrained minimization problems with lower semicontinuous inequality constraints, continuous equality constraints, and a set constraint in smooth Banach spaces. Then we apply these necessary optimality conditions to derive subdifferential characterizations of singular normal vectors to the epigraph and the graph of a function, subdifferential calculus, and necessary optimality conditions for mathematical programs with equilibrium constraints.

Key words. constrained optimization problem, necessary optimality condition, singular normal vector, smooth subdifferential, subdifferential calculus, mathematical programs with equilibrium constraints

AMS subject classifications. 49K27, 49J52, 90C30

PII. S105262340138339

1. Introduction.

Consider the constrained optimization problem

$$\begin{aligned} \mathcal{P} \qquad & \text{minimize } f_0(x) \\ & \text{subject to } f_n(x) \le 0,\ n = 1, \dots, M, \\ & \qquad\qquad f_n(x) = 0,\ n = M+1, \dots, N, \\ & \qquad\qquad x \in C. \end{aligned}$$

We focus mostly on the study of necessary optimality conditions for $\mathcal{P}$. For problems with only smooth equality constraints, necessary optimality conditions can be traced back to Lagrange. The Fritz John and the Karush–Kuhn–Tucker necessary conditions were developed to cope with problems involving inequality constraints. For research on necessary optimality conditions for problem $\mathcal{P}$ with nonsmooth data, we refer to [1, 5, 8, 10, 11, 13, 17, 21, 22, 23] and the references therein. The research in this paper continues that of [1, 14, 16, 17, 21]. In those papers necessary optimality (suboptimality) conditions were established for $\mathcal{P}$ under the mild assumptions that $f_n, n = 0, 1, \dots, M$, are lower semicontinuous extended-valued functions, $f_n, n = M+1, \dots, N$, are continuous functions, and C is a closed subset in finite dimensional spaces, reflexive Banach spaces, and Asplund spaces, respectively. However, certain questions arising in control theory lead to optimization problems in more general spaces (e.g., spaces with weak-Hadamard smooth renorm in [3, 25]). This motivates us to generalize the results in [1, 14, 16, 17, 21] to more general settings. The main purpose of this paper is to prove such necessary optimality conditions in general "bornologically" smooth Banach spaces. Our proofs use ideas similar to those in [1, 21], yet simplify them even in finite dimensional spaces. We should mention that, in Asplund space, it is shown in [14] that under additional normal compactness assumptions one can derive exact necessary optimality conditions in terms of the

*Received by the editors January 11, 2001; accepted for publication (in revised form) October 25, 2001; published electronically April 19, 2002. This research was supported by the National Science Foundation under grant DMS 0102496.

http://www.siam.org/journals/siopt/12-4/38333.html

†Department of Mathematics and Statistics, Western Michigan University, Kalamazoo, MI 49008 (zhu@math-stat.wmich.edu).

limiting normal cone to the epigraphs and graphs of the constraint functions. We will state and prove our main results in section 3 after discussing preliminaries in the next section.

It is worth pointing out that allowing inequality constraints involving lower semicontinuous extended-valued functions and equality constraints involving continuous functions is not merely a technical generalization. Such a general setting enables us to apply our necessary optimality conditions to many seemingly unrelated problems. We discuss several such applications in section 4. They are subdifferential characterizations of singular normal vectors to the epigraph and the graph of a function, subdifferential calculus, and necessary optimality conditions for mathematical programs with equilibrium constraints.

2. Preliminaries. Let X be a real Banach space with closed unit ball B_X and (topological) dual X^*. For any $x \in X$ and $r > 0$ we use $B_r(x)$ to denote the closed ball centered at x with radius r. For a set S in X, we denote its diameter by $\operatorname{diam}(S) := \sup\{||x-y|| : x, y \in S\}$, and we denote its indicator function by i_S, i.e., $i_S(x) = 0$ for $x \in S$, and $i_S(x) = +\infty$ for $x \notin S$. We denote the (minimum) distance between two sets S and T by $d(S,T) := \inf\{\|s-t\| : s \in S \text{ and } t \in T\}$. In particular, when $T = \{x\}$ this reduces to the distance between a point x and a set S, $d(S,x) := \inf\{\|x - s\| : s \in S\}$. A *bornology* β of X is a family of closed bounded and centrally symmetric subsets of X whose union is X, which is closed under multiplication by scalars and is directed upwards. (That is, the union of any two members of β is contained in some member of β.) We will denote by X^*_β the dual space of X endowed with the topology of uniform convergence on β-sets. The most important bornologies are those formed by all (symmetric) bounded sets (the Fréchet bornology, denoted by F), weak compact sets (the weak Hadamard bornology, denoted by WH), compact sets (the Hadamard bornology, denoted by H), and finite sets (the Gateaux bornology, denoted by G).

We will define a *convex bornology* as one that also contains all convex closures of the sets in the corresponding bornology. In particular, any finite dimensional subspace is included in the subspace spanned by some element of a convex bornology. In this paper, we consider only convex bornology, usually denoted by β. Note that the convex Gateaux bornology lies strictly between the Gateaux and Hadamard bornology, while for the Fréchet, weak Hadamard, and Hadamard bornologies the convex and nonconvex definitions are the same.

By a *function* we always mean an *extended-real-valued* function, usually lower semicontinuous and *proper* (that is to say, not everywhere equal to $+\infty$ and nowhere to $-\infty$). Given a function f on X, we say that f is β-differentiable at x and has a β-derivative $\nabla^\beta f(x)$ if $f(x)$ is finite and

$$t^{-1}(f(x+tu) - f(x) - t\langle \nabla^\beta f(x), u\rangle) \to 0$$

as $t \to 0$, uniformly in $u \in V$ for every $V \in \beta$. We say that a function f is β-smooth at x if f is β-differentiable in a neighborhood U of x and $\nabla^\beta f : U \to X^*_\beta$ is continuous in a neighborhood of x. It is not hard to check that a convex function f is β-smooth at x if and only if f is β-differentiable on a convex neighborhood of x. Now we can define the β-subdifferential and the related β-normal cone.

DEFINITION 2.1 (β-subdifferential and normal cone). *Let $f : X \to R \cup \{+\infty\}$ be a lower semicontinuous function and $f(x) < +\infty$. We say that f is* β-subdifferentiable *and x^* is a* β-viscosity subderivative *of f at x if there exists a locally Lipschitz function g such that g is β-smooth at x, $\nabla^\beta g(x) = x^*$, and $f - g$ attains a local minimum at*

x. We call the set of all β-viscosity subderivatives of f at x the β-subdifferential of f at x, and we denote it by $\partial_\beta f(x)$. For a closed subset $S \subset X$ and $x \in S$, we define the β-normal cone of S at x by $N_\beta(S;x) := \partial_\beta i_S(x)$.

In what follows, a lower semicontinuous function is understood as a function with range $R \cup \{+\infty\}$. Recall that a *bump function* is a bounded function with a bounded nonempty support. We say that a Banach space is *β-smooth*, provided that it has a β-smooth Lipschitzian bump function. We will often need the following weak fuzzy sum rule (see [2, 7, 27]).

THEOREM 2.2 (Weak fuzzy sum rule). *Let X be a β-smooth Banach space. Let $f_1, \ldots, f_N$ be lower semicontinuous functions on X, with $x \in \cap_{n=1}^N \mathrm{dom}(f_n)$. Then, for any $x^* \in \partial_\beta(\sum_{n=1}^N f_n)(x)$, any $\varepsilon > 0$, and any weak-star neighborhood U of 0 in X^*, there exist $x_n \in B_\varepsilon(x), x_n^* \in \partial_\beta f_n(x_n), n = 1, \ldots, N$, such that $|f_n(x_n) - f_n(x)| < \varepsilon$, $||x_n^*|| \cdot \mathrm{diam}(\{x_1, \ldots, x_N\}) < \varepsilon, n = 1, 2, \ldots, N$, and*

$$x^* \in \sum_{n=1}^N x_n^* + U.$$

For the Fréchet subdifferential, the arbitrary weak-star neighborhood U in Theorem 2.2 can be replaced by an arbitrary norm neighborhood under additional conditions. Such results are often referred to as strong fuzzy sum rules. We will need the following form of the strong fuzzy sum rule (see [15, 27]).

THEOREM 2.3 (Strong fuzzy sum rule). *Let X be an Asplund space. Let $f_1, \ldots, f_N$ be lower semicontinuous functions on X, with $x \in \cap_{n=1}^N \mathrm{dom}(f_n)$. Suppose that all but one of $f_n, n = 1, 2, \ldots, N$, are locally Lipschitz around x. Then, for any $x^* \in \partial_F(\sum_{n=1}^N f_n)(x)$ and any $\varepsilon > 0$, there exist $x_n \in B_\varepsilon(x), x_n^* \in \partial_F f_n(x_n), n = 1, \ldots, N$, such that $|f_n(x_n) - f_n(x)| < \varepsilon$, $||x_n^*|| \cdot \mathrm{diam}(\{x_1, \ldots, x_N\}) < \varepsilon, n = 1, 2, \ldots, N$, and*

$$\left\| x^* - \sum_{n=1}^N x_n^* \right\| < \varepsilon.$$

3. Necessary optimality conditions. We establish necessary optimality conditions for problem $\mathcal{P}$ in β-smooth Banach spaces that generalize those in [1, 17, 21]. Following [1], we will use the quantities $\tau_n, n = 0, 1, \ldots, N$, to simplify the notation. The τ_n's associated with the inequality constraints and the cost function are always 1, i.e., $\tau_n := 1, n = 0, 1, \ldots, M$. This corresponds to nonnegative multipliers. The τ_n's associated with the equality constraints are either 1 or -1, corresponding to multipliers with arbitrary sign, i.e., $\tau_n \in \{-1, 1\}, n = M+1, \ldots, N$. Our main result is the following necessary optimality conditions.

THEOREM 3.1 (Fuzzy multiplier rule). *Let X be a β-smooth Banach space, let C be a closed subset of X, and let f_n be lower semicontinuous for $n = 0, 1, \ldots, M$ and continuous for $n = M+1, \ldots, N$. Assume that $\bar{x}$ is a local solution of $\mathcal{P}$. Let $\varepsilon > 0$ be an arbitrary positive number, and let U be an arbitrary weak-star neighborhood of 0 in X^*. Suppose that $\liminf_{x \to \bar{x}} d(\partial_\beta f_n(x), U) > 0$ for $n = 1, \ldots, M$, and $\liminf_{x \to \bar{x}} d(\partial_\beta f_n(x) \cup \partial_\beta(-f_n)(x), U) > 0$ for $n = M+1, \ldots, N$. Then there exist $(x_n, f_n(x_n)) \in B_\varepsilon(\bar{x}, f_n(\bar{x})), n = 0, 1, \ldots, N$, and $x_{N+1} \in B_\varepsilon(\bar{x}) \cap C$ such that*

$$0 \in \partial_\beta f_0(x_0) + \sum_{n=1}^N \mu_n \partial_\beta(\tau_n f_n)(x_n) + N_\beta(C; x_{N+1}) + U,$$

where $\mu_n > 0, n = 1, \ldots, N$.

Without the condition that the β-subdifferentials of $\tau_n f_n$'s are bounded away from U, we cannot guarantee that the coefficient of $\partial_\beta f_0(x_0)$ is 1. Nevertheless, the following weaker necessary condition always holds.

THEOREM 3.2 (Weak fuzzy multiplier rule). *Let X be a β-smooth Banach space, let C be a closed subset of X, and let f_n be lower semicontinuous for $n = 0, 1, \ldots, M$ and continuous for $n = M+1, \ldots, N$. Assume that $\bar{x}$ is a local solution of $\mathcal{P}$. Then, for any positive number $\varepsilon > 0$ and any weak-star neighborhood U of 0 in X^*, there exist $(x_n, f_n(x_n)) \in B_\varepsilon(\bar{x}, f_n(\bar{x})), n = 0, 1, \ldots, N$, and $x_{N+1} \in B_\varepsilon(\bar{x}) \cap C$ such that*

$$0 \in \sum_{n=0}^{N} \mu_n \partial_\beta(\tau_n f_n)(x_n) + N_\beta(C; x_{N+1}) + U,$$

where $\mu_n \geq 0, n = 0, 1, \ldots, N$, satisfy $\sum_{n=0}^{N} \mu_n = 1$.

Proof. Let V be a weak-star neighborhood of 0 in X^*, and let $r > 0$ satisfy $V + rB_{X^*} \subset U$. If $\liminf_{x\to\bar{x}} d(\partial_\beta f_n(x), V) > 0$ for $n = 1, \ldots, M$, and $\liminf_{x\to\bar{x}} d(\partial_\beta f_n(x) \cup \partial_\beta(-f_n)(x), V) > 0$ for $n = M+1, \ldots, N$, then the conclusion of Theorem 3.1 holds. We need only to rescale the result of Theorem 3.1 by multiplying $\mu_0 = 1/(1+\sum_{n=1}^{N} \mu_n)$. Suppose that one of the conditions, say the condition corresponding to index j, fails. Then the conclusion of Theorem 3.2 holds trivially for $\mu_j = 1$ and $\mu_n = 0, n \neq j$. □

We turn to the proof of Theorem 3.1. The idea is simple. We observe that if $\bar{x}$ is a solution to the constrained optimization problem $\mathcal{P}$, then it is a local minimum of the following function:

$$f_0 + \sum_{n=1}^{M} i_{f_n^{-1}((-\infty, f_n(\bar{x})])} + \sum_{n=M+1}^{N} i_{f_n^{-1}(0)} + i_C,$$

where $f^{-1}(S) := \{x \in X : f(x) \in S\}$. Applying the weak fuzzy sum rule of Theorem 2.2 yields a necessary condition in terms of the subdifferential of f_0 and the normal cones to the level sets of the f_n and C. The key then is to relate the normal cones to the level sets of the f_n to their subdifferentials. We discuss their relationship in the following theorems.

THEOREM 3.3. *Let X be a β-smooth Banach space, and let $f : X \to R \cup \{+\infty\}$ be a lower semicontinuous function. Suppose that $\xi \in N_\beta(f^{-1}((-\infty, a]); \bar{x})$. Then, for any $\varepsilon > 0$ and any weak-star neighborhood U of 0 in X^*, either*

(a)

$$\liminf_{x\to\bar{x}} d(\partial_\beta f(x), U) = 0$$

or

(b) *there exist $\lambda > 0$, $(x, f(x)) \in B_\varepsilon((\bar{x}, f(\bar{x})))$, and $x^* \in \partial_\beta f(x)$ such that*

$$\lambda x^* - \xi \in U.$$

Proof. Assume that (a) is not true, i.e.,

$$\liminf_{x\to\bar{x}} d(\partial_\beta f(x), U) > c > 0. \tag{1}$$

We prove (b) for the nontrivial case when $\xi \neq 0$. We may always assume that $a = f(\bar{x})$, because $\bar{x} \in f^{-1}((-\infty, a])$ implies that $N_\beta(f^{-1}((-\infty, a]); \bar{x}) \subset N_\beta(f^{-1}((-\infty, f(\bar{x})]); \bar{x})$.

Choose a weak-star neighborhood V of 0 in X^*, a finite dimensional subspace L of X, and a positive number r such that $rB_{X^*} + L^\perp + \max(1, \|\xi\|(1+r)/c)V \subset U$. Without loss of generality we may assume that $\varepsilon < r/(1 + \|\xi\|)$. Choose $\delta \in (0, \varepsilon)$ such that

$$\langle \xi, h\rangle < \varepsilon\|\xi\|\|h\| \quad \forall \bar{x} + h \in f^{-1}((-\infty, a]) \cap (\bar{x} + L) \cap B_\delta(\bar{x}),\ h \neq 0, \tag{2}$$

and

$$\inf_{x \in B_\delta(\bar{x})} d(\partial_\beta f(x), U) > c.$$

Define

$$K(\xi, \varepsilon) := \{x \in X : \langle \xi, x\rangle \geq \varepsilon\|\xi\|\|x\|\}.$$

Then

$$[\bar{x} + K(\xi, \varepsilon)] \cap f^{-1}((-\infty, a]) \cap (\bar{x} + L) \cap B_\delta(\bar{x}) = \{\bar{x}\}, \tag{3}$$

and, therefore,

$$g := f + i_{\bar{x}+K(\xi,\varepsilon)} + i_{\bar{x}+L}$$

attains a (local) minimum $f(\bar{x})$ at $\bar{x}$ over $B_\delta(\bar{x})$. Note that $N_\beta(\bar{x} + K(\xi, \varepsilon); \cdot) \subset \bigcup_{\alpha \geq 0} \alpha(-\xi + \varepsilon\|\xi\|B_{X^*})$. Applying the weak fuzzy sum rule of Theorem 2.2, we conclude that there exist $x \in B_\delta(\bar{x}) \subset B_\varepsilon(\bar{x})$ such that

$$0 \in \partial_\beta f(x) + \bigcup_{\alpha \geq 0} \alpha(-\xi + \varepsilon\|\xi\|B_{X^*}) + L^\perp + V. \tag{4}$$

Now choose $x^* \in \partial_\beta f(x)$, $\alpha \geq 0$, and $b^* \in B_{X^*}$ such that

$$0 \in x^* + \alpha(-\xi + \varepsilon\|\xi\|b^*) + L^\perp + V. \tag{5}$$

We must have $\alpha\|\xi\|(1 + \varepsilon) \geq c$, for otherwise inclusion (5) would imply $d(x^*, U) \leq d(x^*, V + L^\perp) \leq c$, a contradiction. Let $\lambda = 1/\alpha > 0$. Then $\lambda \leq \|\xi\|(1 + \varepsilon)/c$. Multiplying inclusion (5) by λ, we have

$$\begin{aligned} 0 &\in \lambda x^* - \xi + \varepsilon\|\xi\|b^* + L^\perp + \lambda V \\ &\subset \lambda x^* - \xi + rB_{X^*} + L^\perp + \frac{\|\xi\|(1+\varepsilon)}{c} V \\ &\subset \lambda x^* - \xi + U. \qquad \square \end{aligned}$$

Similarly we have a corresponding result for the normal cone to the level sets $f^{-1}(a)$ of a continuous function.

THEOREM 3.4. *Let X be a β-smooth Banach space, and let $f : X \to R$ be a continuous function. Suppose that $\xi \in N_\beta(f^{-1}(a); \bar{x})$. Then, for any $\varepsilon > 0$ and any weak-star neighborhood U of 0 in X^*, either*

(a)

$$\liminf_{x \to \bar{x}} d(\partial_\beta f(x) \cup \partial_\beta(-f)(x), U) = 0$$

or

(b) *there exist* $\lambda > 0$, $(x, f(x)) \in B_\varepsilon((\bar{x}, f(\bar{x})))$, *and* $x^* \in \partial_\beta f(x) \cup \partial_\beta(-f)(x)$ *such that*

$$\lambda x^* - \xi \in U.$$

Proof. Assume that (a) is not true, i.e.,

$$\liminf_{x \to \bar{x}} d(\partial_\beta f(x) \cup \partial_\beta(-f)(x), U) > c > 0. \tag{6}$$

As in the proof of Theorem 3.3, we prove (b) for the case in which $\xi \neq 0$. Choose a weak-star neighborhood V of 0 in X^*, a finite dimensional subspace L of X, and a positive number r such that $rB_{X^*} + L^\perp + \max(1, \|\xi\|(1+r)/c)V \subset U$. Without loss of generality we may assume that $\varepsilon < r/(1 + \|\xi\|)$. Choose $\delta \in (0, \varepsilon)$ such that

$$\langle \xi, h\rangle < \varepsilon\|\xi\|\|h\| \quad \forall \bar{x} + h \in f^{-1}(a) \cap (\bar{x} + L) \cap B_\delta(\bar{x}),\ h \neq 0, \tag{7}$$

and

$$\inf_{x \in B_\delta(\bar{x})} d(\partial_\beta f(x) \cup \partial_\beta(-f)(x), U) > c.$$

Define

$$K(\xi, \varepsilon) := \{x \in X : \langle \xi, x\rangle \geq \varepsilon\|\xi\|\|x\|\}.$$

Then

$$[\bar{x} + K(\xi, \varepsilon)] \cap f^{-1}(a) \cap (\bar{x} + L) \cap B_\delta(\bar{x}) = \{\bar{x}\}. \tag{8}$$

We have that either

(a) $f(x) \geq a\ \forall x \in [\bar{x} + K(\xi, \varepsilon)] \cap (\bar{x} + L) \cap B_\delta(\bar{x})$ or

(b) $f(x) \leq a\ \forall x \in [\bar{x} + K(\xi, \varepsilon)] \cap (\bar{x} + L) \cap B_\delta(\bar{x})$.

In fact, suppose on the contrary that there exist $x_1, x_2 \in [\bar{x}+K(\xi, \varepsilon)] \cap (\bar{x}+L) \cap B_\delta(\bar{x})$ such that $f(x_1) > a$ and $f(x_2) < a$. Then $x_1, x_2 \neq \bar{x}$. Since f is continuous, there exists $r \in (0, 1)$ such that $z := rx_1 + (1 - r)x_2$ satisfies $f(z) = a$. Clearly $z \in [\bar{x} + K(\xi, \varepsilon)] \cap (\bar{x} + L) \cap B_\delta(\bar{x})$, and therefore $z = \bar{x}$ by (8). However, this leads to $0 = r(x_1 - \bar{x}) + (1-r)(x_2 - \bar{x})$ or $r(x_1 - \bar{x}) = -(1-r)(x_2 - \bar{x}) \in K(\xi, \varepsilon) \cap [-K(\xi, \varepsilon)] = \{0\}$, a contradiction.

Define

$$g := \begin{cases} f - a + i_{\bar{x}+K(\xi,\varepsilon)} + i_{\bar{x}+L} & \text{if we have (a)}, \\ -f + a + i_{\bar{x}+K(\xi,\varepsilon)} + i_{\bar{x}+L} & \text{if we have (b)}. \end{cases}$$

Then, it follows from (8) that g attains a (local) minimum $f(\bar{x})$ at $\bar{x}$ over $B_\delta(\bar{x})$.

The rest of the proof is the same as that of Theorem 3.3. □

Now we can prove Theorem 3.1.

Proof of Theorem 3.1. Let V be a weak-star neighborhood of 0 in X^* such that $(N + 1)V \subset U$. Then, we have $\liminf_{x \to \bar{x}} d(\partial_\beta f_n(x), V) > 0$ for $n = 1, \ldots, M$ and $\liminf_{x \to \bar{x}} d(\partial_\beta f_n(x) \cup \partial_\beta(-f_n)(x), V) > 0$ for $n = M + 1, \ldots, N$. Decreasing ε if necessary, we may assume that, for any $y \in B_\varepsilon(\bar{x})$, $\liminf_{x \to y} d(\partial_\beta f_n(x), V) > 0$ for $n = 1, \ldots, M$, and $\liminf_{x \to y} d(\partial_\beta f_n(x) \cup \partial_\beta(-f_n)(x), V) > 0$ for $n = M + 1, \ldots, N$. Observe that $\bar{x}$ is a minimum of the function

$$f_0 + \sum_{n=1}^{M} i_{f_n^{-1}((-\infty, f_n(\bar{x})])} + \sum_{n=M+1}^{N} i_{f_n^{-1}(0)} + i_C.$$

Since f_n, $n = 1, \dots, M$, are lower semicontinuous and f_n, $n = M+1, \dots, N$, are continuous we can choose $\eta \in (0, \varepsilon/2)$ such that $y \in B_\eta(\bar{x})$ implies $f_n(y) > f_n(\bar{x}) - \varepsilon/2$ for $n = 1, \dots, M$, and $f_n(y) \in (f_n(\bar{x}) - \varepsilon/2, f_n(\bar{x}) + \varepsilon/2)$ for $n = M+1, \dots, N$. By the weak fuzzy sum rule of Theorem 2.2 there exist $(x_0, f_0(x_0)) \in B_\eta(\bar{x})$, $x_{N+1} \in B_\eta(\bar{x}) \cap C$, $y_n \in B_\eta(\bar{x}), n = 1, \dots, N$, with $|i_{f_n^{-1}((-\infty, f_n(\bar{x})])}(y_n) - i_{f_n^{-1}((-\infty, f_n(\bar{x})])}(\bar{x})| < \eta$, $n = 1, \dots, M$, and $|i_{f_n^{-1}(0)}(y_n) - i_{f_n^{-1}(0)}(\bar{x})| < \eta$, $n = M+1, \dots, N$ (and, therefore, $(y_n, f_n(y_n)) \in B_{\varepsilon/2}((\bar{x}, f_n(\bar{x})))$), $x_0^* \in \partial_\beta f(x_0)$, $x_{N+1}^* \in N_\beta(C; x_{N+1})$, $y_n^* \in N_\beta(f_n^{-1}((-\infty, f_n(\bar{x})]); y_n)$ for $n = 1, \dots, M$, and $y_n^* \in N_\beta(f_n^{-1}(0); y_n)$ for $n = M+1, \dots, N$ such that

$$0 \in x_0^* + \sum_{n=1}^{N} y_n^* + x_{N+1}^* + V. \tag{9}$$

Theorems 3.3 and 3.4 imply that there exist $(x_n, f_n(x_n)) \in B_{\varepsilon/2}((y_n, f_n(y_n))) \subset B_\varepsilon((\bar{x}, f_n(\bar{x})))$, $\mu_n > 0$, and $x_n^* \in \partial_\beta f_n(x_n)$ $(x_n^* \in \partial_\beta f_n(x_n) \cup \partial_\beta(-f_n)(x_n))$ for $n = 1, \dots, M$ $(n = M+1, \dots, N)$ such that

$$y_n^* \in \mu_n x_n^* + V. \tag{10}$$

Combining (9) and (10) completes the proof. □

4. Applications.

4.1. Approximation of singular normal vectors to the epigraph and the graph of a function. The relationship between the normal vectors to the epigraphs and graphs of a function and the subdifferential of the function plays an important role in nonsmooth problems. The characterization of such singular normal vectors is rather delicate. It first appeared in Rockafellar [20] for functions on finite dimensional spaces. Infinite dimensional generalizations were discussed in [9, 18] for the Fréchet singular normal vectors. Here we show that an extension of such characterizations to β-smooth singular normal vectors follows directly from Theorems 3.3 and 3.4.

THEOREM 4.1. *Let X be a β-smooth Banach space, and let $f : X \to R \cup \{+\infty\}$ be a lower semicontinuous function. Suppose that $(\bar{x}^*, 0) \in N_\beta(\text{epi } f; (\bar{x}, a))$ (which implies that $a \geq f(\bar{x})$). Then, for any $\varepsilon > 0$ and any weak-star neighborhood U of 0 in X^*, there exist x, x^*, and $\lambda \in (0, \varepsilon)$ such that $x^* \in \partial_\beta f(x)$, $(x, f(x)) \in B_\varepsilon(\bar{x}, f(\bar{x}))$, and*

$$\lambda x^* - \bar{x}^* \in U.$$

Proof. We may assume that $\varepsilon < 1$. Since $N_\beta(\text{epi } f; (\bar{x}, a)) \subset N_\beta(\text{epi } f; (\bar{x}, f(\bar{x})))$, we may assume that $a = f(\bar{x})$. Set $F(x, t) = f(x) - t$. Then epi $f = F^{-1}((-\infty, 0])$. We next apply Theorem 3.3 to the function F. Since, for any (x, t), $d(\partial_\beta F(x, t), U \times [-\varepsilon, \varepsilon]) > 0$, by (b) there exist

$$((x, t), F(x, t)) \in B_{\varepsilon/2}(((\bar{x}, f(\bar{x})), 0)), \tag{11}$$

$$(x^*, t^*) \in \partial_\beta F(x, t), \tag{12}$$

and $\lambda > 0$ such that

$$\lambda(x^*, t^*) - (\bar{x}^*, 0) \in U \times (-\varepsilon, \varepsilon). \tag{13}$$

It follows from the definition of F and (11) and (12) that $t^* = -1$ and $(x, f(x)) \in B_\varepsilon((\bar{x}, f(\bar{x})))$. Thus, inclusion (13) implies that $\lambda \in (0, \varepsilon)$ and

$$\lambda x^* - \bar{x}^* \in U. \qquad \square$$

Similarly, applying Theorems 3.4 and 4.4 to $F^{-1}(0)$, we have the following parallel results.

THEOREM 4.2. *Let X be a β-smooth Banach space, and let $f : X \to R \cup \{+\infty\}$ be a continuous function. Suppose that $(\bar{x}^*, 0) \in N_\beta(\text{graph } f; (\bar{x}, f(\bar{x})))$. Then, for any $\varepsilon > 0$ and any weak-star neighborhood U of 0 in X^*, there exist x, x^*, and $\lambda \in (0, \varepsilon)$ such that $x^* \in \partial_\beta f(x) \cup \partial_\beta(-f)(x)$, $(x, f(x)) \in B_\varepsilon(\bar{x}, f(\bar{x}))$, and*

$$\lambda x^* - \bar{x}^* \in U.$$

For the Fréchet subdifferential, it is possible to strengthen Theorems 3.3 and 3.4 by replacing the arbitrary weak-star neighborhood U of 0 in X^* by an arbitrary ball around 0 under additional assumptions on the geometric property of the space X. Such stronger versions of these theorems are derived, in finite dimensional spaces, in reflexive Banach spaces, and in Asplund spaces, respectively, in [21, 1, 17]. The characterization of the Fréchet singular normal vectors in [9, 17, 20] then follows immediately with arguments similar to those in Theorems 4.1 and 4.2. For completeness, we give an alternative short proof of the Asplund space version of Theorems 3.3 and 3.4 in [18].

THEOREM 4.3. *Let X be an Asplund space, and let $f : X \to R \cup \{+\infty\}$ be a lower semicontinuous function. Suppose that $\xi \in N_F(f^{-1}((-\infty, a]); \bar{x})$. Then either*

(a)

$$\liminf_{x \to \bar{x}} d(\partial_F f(x), 0) = 0$$

or

(b) *for any $\varepsilon > 0$ there exist $\lambda > 0$, $(x, f(x)) \in B_\varepsilon((\bar{x}, f(\bar{x})))$, and $x^* \in \partial_F f(x)$ such that*

$$\|\lambda x^* - \xi\| < \varepsilon.$$

Proof. As in the proof of Theorem 3.3 we consider the case in which

$$\liminf_{x \to \bar{x}} d(\partial_F f(x), 0) > c > 0, \tag{14}$$

$\xi \neq 0$, and $a = f(\bar{x})$. Choose $\eta \in (0, \varepsilon)$ satisfying

$$2\eta\|\xi\| + \frac{\eta\|\xi\|(1 + 2\eta)}{2c} < \frac{\varepsilon}{2}, \tag{15}$$

and choose $\delta \in (0, \varepsilon)$ such that

$$\langle \xi, h \rangle < \eta\|\xi\|\|h\| \quad \forall \bar{x} + h \in f^{-1}((-\infty, a]) \cap B_\delta(\bar{x}),\ h \neq 0, \tag{16}$$

f is bounded below on $B_\delta(\bar{x})$, and

$$\inf_{x \in B_\delta(\bar{x})} d(\partial_F f(x), 0) > c.$$

Then

$$[\bar{x} + K(\xi, \eta)] \cap f^{-1}((-\infty, a]) \cap B_\delta(\bar{x}) = \{\bar{x}\}, \tag{17}$$

where $K(\xi, \eta)$ is the cone defined in the proof of Theorem 3.3. Define, for each natural number k,

$$g_k := f - a + k d_{\bar{x}+K(\xi,2\eta)}.$$

Then g_k is bounded below on $B_\delta(\bar{x})$, and $g_k(\bar{x}) = 0$. We consider two possible cases: (A) $\inf_{B_\delta(\bar{x})} g_k < 0$ and (B) $\inf_{B_\delta(\bar{x})} g_k = 0$. If we have case (A), then by the Ekeland variational principle [6] there exists $y_k \in B_\delta(\bar{x})$ such that $g_k(y_k) < 0$ and

$$z \to g_k(z) + \frac{1}{k}\|z - y_k\|$$

attains a (local) minimum at y_k over $B_\delta(\bar{x})$. We must have $y_k \notin \bar{x} + K(\xi, \eta)$, for otherwise (17) implies that $y_k \notin f^{-1}((-\infty, a])$ and $g_k(y_k) \geq 0$, a contradiction. We claim that

$$d_{\bar{x}+K(\xi,2\eta)}(y_k) \geq \frac{\eta}{(2\eta + 1)}\|y_k - \bar{x}\|. \tag{18}$$

Indeed, if

$$\|h\| < \frac{\eta}{(2\eta + 1)}\|y_k - \bar{x}\|,$$

or equivalently

$$\|h\| < \eta\|y_k - \bar{x}\| - 2\eta\|h\|,$$

then

$$\begin{aligned}
\langle \xi, y_k - \bar{x} + h\rangle &= \langle \xi, y_k - \bar{x}\rangle + \langle \xi, h\rangle \\
&< \eta\|\xi\|\|y_k - \bar{x}\| + \|\xi\|\|h\| \\
&< 2\eta\|\xi\|[\|y_k - \bar{x}\| - \|h\|] \\
&\leq 2\eta\|\xi\|\|y_k - \bar{x} + h\|.
\end{aligned}$$

That is to say, $y_k - \bar{x} + h \notin K(\xi, 2\eta)$. Since $f(y_k)$ is bounded from below, we have

$$\lim_{k\to\infty} d_{\bar{x}+K(\xi,2\eta)}(y_k) = 0. \tag{19}$$

Combining (18) and (19), we have $y_k \to \bar{x}$ as $k \to \infty$. Therefore, for k sufficiently large we have $y_k \in \text{int } B_\delta(\bar{x})$. In the case (B) we set $y_k = \bar{x}$. Thus, in both cases (A) and (B),

$$z \to g_k(z) + \frac{1}{k}\|z - y_k\|$$

attains a local minimum at y_k when k is sufficiently large. By the strong fuzzy sum rule of Theorem 2.3 there exist $x_k, z_k \in \text{int } B_\delta(\bar{x})$, $x_k^* \in \partial_F f(x_k)$, and $z_k^* \in \partial_F d_{\bar{x}+K(\xi,2\eta)}(z_k)$ such that

$$\|x_k^* + k z_k^*\| < \eta + \frac{1}{k}. \tag{20}$$

Since $K(\xi, 2\eta)$ is convex, so is the function $d_{\bar{x}+K(\xi,2\eta)}$. Moreover, this function is a contraction. By convex analysis we have

$$\partial_F d_{\bar{x}+K(\xi,2\eta)}(\cdot) \subset \{\alpha(-\xi + 2\eta\|\xi\|B_{X^*}) : \alpha > 0\} \cap B_{X^*}. \tag{21}$$

Let $z_k^* = \alpha_k(-\xi + 2\eta\|\xi\|b^*)$ for some $b^* \in B_{X^*}$. It follows from (20) that

$$\|x_k^* - k\alpha_k\xi\| < 2k\alpha_k\|\xi\|\eta + \eta + \frac{1}{k}. \tag{22}$$

We must have $k\alpha_k > c/[2\|\xi\|(1 + 2\eta)]$, for otherwise we would have $\|x_k^*\| < c$, a contradiction. Now letting $\lambda_k = 1/k\alpha_k$ and multiplying (22) by λ_k, we have

$$\|\lambda_k x_k^* - \xi\| < 2\eta\|\xi\| + \frac{2\eta\|\xi\|(1+2\eta)}{c} + \frac{2\|\xi\|(1+2\eta)}{kc}.$$

For

$$k > \frac{c}{\|\xi\|(1+2\eta)\varepsilon},$$

setting $\lambda = \lambda_k$, $x = x_k$, and $x^* = x_k^*$, we have

$$\|\lambda x^* - \xi\| < \varepsilon. \qquad \square$$

Modifying the proof of Theorem 4.3 as in the proof of Theorem 3.4, we can prove a corresponding approximation for the Fréchet normal cone of $f^{-1}(a)$ when f is a continuous function.

THEOREM 4.4. *Let X be an Asplund space, and let $f : X \to R \cup \{+\infty\}$ be a continuous function. Suppose that $\xi \in N_F(f^{-1}(a); \bar{x})$. Then either*

(a)

$$\liminf_{x \to \bar{x}} d(\partial_F f(x) \cup \partial_F(-f)(x), 0) = 0$$

or

(b) *for any $\varepsilon > 0$ there exist $\lambda > 0$, $(x, f(x)) \in B_\varepsilon((\bar{x}, f(\bar{x})))$, and $x^* \in \partial_F f(x) \cup \partial_F(-f)(x)$ such that*

$$\|\lambda x^* - \xi\| < \varepsilon.$$

The methods in the proofs of Theorems 4.1 and 4.2 provide an alternative proof for the following Fréchet subdifferential characterization of the singular Fréchet normal vectors to the epigraph and graph of a function.

THEOREM 4.5 (see [17]). *Let X be an Asplund space, and let $f : X \to R \cup \{+\infty\}$ be a lower semicontinuous function. Suppose that $(\bar{x}^*, 0) \in N_F(\text{epi } f; (\bar{x}, a))$ $((\bar{x}^*, 0) \in N_F(\text{graph } f; (\bar{x}, f(\bar{x}))))$. Then, for any $\varepsilon > 0$, there exist x, x^*, and $\lambda \in (0, \varepsilon)$ such that $x^* \in \partial_F f(x)$ $(x^* \in \partial_F f(x) \cup \partial_F(-f)(x))$, $(x, f(x)) \in B_\varepsilon(\bar{x}, f(\bar{x}))$, and*

$$\|\lambda x^* - \bar{x}^*\| < \varepsilon.$$

Remark. Theorems 4.3, 4.4, and 4.5 were discussed in [17, 18] using a different approach. Their authors first prove Theorem 4.5 with a method patterned on that of Ioffe [9] and then deduce Theorems 4.3 and 4.4 from Theorem 4.5. The latter procedure is quite involved. It appears that our approach here is somewhat more efficient.

4.2. Chain rule and subdifferential calculus. Here we show that the necessary optimality conditions in Theorem 3.2 imply a chain rule, which in turn yields other subdifferential calculus rules such as sum rules, product rules, and quotient rules. The key relationship between the chain rule and the necessary optimality condition for constrained minimization problems is established through the following observation discussed in [4, 26].

Consider functions $f_1, \dots, f_N : X \to R \cup \{+\infty\}$ and a function $f : R^N \to R \cup \{+\infty\}$ that is nondecreasing for each of its first M variables ($M \le N$). Suppose that $f(f_1, \dots, f_N)$ attains a local minimum at $\bar{x}$. Then one can check that $(\bar{x}, (f_1(\bar{x}), \dots, f_N(\bar{x})))$ is a local solution to the following minimization problem (on $X \times R^N$):

$$\begin{aligned} &\text{minimize } f(y) \\ &\text{subject to } f_n(x) - y_n \le 0,\ n = 1, \dots, M, \\ &\qquad\qquad f_n(x) - y_n = 0,\ n = M+1, \dots, N. \end{aligned}$$

Applying the fuzzy multiplier rule of Theorem 3.1 yields the following chain rule.

THEOREM 4.6 (Fuzzy chain rule). *Let X be a β-smooth Banach space. Suppose that $f_1, \dots, f_M : X \to R \cup \{+\infty\}$ are lower semicontinuous functions, that $f_{M+1}, \dots, f_N$ are continuous functions, and that $f : R^N \to R \cup \{+\infty\}$ is a lower semicontinuous function nondecreasing for each of its first M variables ($M \le N$). Suppose that $f(f_1, \dots, f_N)$ attains a local minimum at $\bar{x}$. Then, for any positive number $\varepsilon > 0$ and any weak-star neighborhood U of 0 in X^*, there exist $(x_n, f_n(x_n)) \in (\bar{x}, f_n(\bar{x})) + \varepsilon B_{X \times R}$, $n = 0, 1, \dots, N$, $(y, f(y)) \in (\bar{y}, f(\bar{y})) + \varepsilon B_{R^{N+1}}$, where $\bar{y} = (f_1(\bar{x}), \dots, f_N(\bar{x}))$, and $\mu = (\mu_1, \dots, \mu_N) \in \partial_\beta f(y) + \varepsilon B_{R^N}$ such that*

$$0 \in \sum_{n=1}^{N} \partial_\beta(\mu_n f_n)(x_n) + U.$$

Remark. (a) When f is C^1 we have more precisely $\mu = f'(\bar{y})$. This smooth version of the chain rule is useful in deriving other calculus rules. For example, setting $f(f_1, \dots, f_N) := \sum_{n=1}^N f_n$, $f(f_1, \dots, f_N) := \prod_{n=1}^N f_n$, and $f(f_1, f_2) := f_1/f_2$, we may deduce a subdifferential sum rule, a subdifferential product rule, and a subdifferential quotient rule, respectively.

(b) Similarly, setting $f(f_1, \dots, f_N) := \max\{f_n : n = 1, \dots, N\}$, we may deduce a subdifferential formula for the maximum function.

Remark. In fact, the chain rule in this section is equivalent to the necessary optimality conditions for the constrained minimization problem in Theorem 3.1. Here we use Theorem 3.1 to deduce the subdifferential chain rule. The use of a chain rule to deduce necessary optimality conditions for constrained minimization problems can be found, e.g., in [17]. (The discussion in [17] is for the Fréchet subdifferential, but the methods used there are also applicable to the more general β-subdifferentials discussed here.)

4.3. Mathematical programs with equilibrium constraints. Mathematical programs with equilibrium constraints (MPEC) are a generalization of the bilevel programming problem. In this section we show that the necessary optimality conditions in Theorem 3.1 can be used to deduce necessary optimality conditions for such problems. Lou, Pang, and Ralph's monograph [12] is an excellent source for the history and the state of the art of this problem up to 1996. We will discuss the

following generalized MPEC (in Euclidean spaces) introduced in Outrata [19], which also encompasses problems outside the traditional setting of MPEC:

$$\begin{array}{lll} \mathcal{MPEC} & \text{Minimize} & f_0(x) \\ & \text{subject to} & 0 \in h(x) + F(g(x)), \\ & & x \in C, \end{array}$$

where $f_0 : R^r \to R \cup \{+\infty\}$ is a lower semicontinuous function, C is a subset of R^r, $h : R^r \to R^q$ and $g : R^r \to R^p$ are continuous functions, and $F : R^p \to 2^{R^q}$ is a multifunction with a closed graph.

The problem $\mathcal{MPEC}$ is very general. It encompasses many optimization problems with nonstandard constraints (see [19] and the references therein). We can also recover problem $\mathcal{P}$ from $\mathcal{MPEC}$ by letting $g(x) = x$, $F(x) = [f_1(x), +\infty) \times \cdots \times [f_M(x), +\infty) \times \{0_{N-M}\}$, and $h(x) = \{0_M\} \times (f_{M+1}(x), \ldots, f_N(x))$, where 0_I is the origin of R^I.

However, the main point we would like to make in this section is that it is not hard to convert $\mathcal{MPEC}$ into $\mathcal{P}$. In doing so we can easily derive necessary optimality conditions for $\mathcal{MPEC}$ from Theorem 3.1. To make the notation easier we denote the components of g and h by $g = (f_1, \ldots, f_p)$ and $h = (f_{p+1}, \ldots, f_{p+q})$. Then it is not hard to see that if $\bar{x}$ is a solution to problem $\mathcal{MPEC}$, then $(\bar{x}, g(\bar{x}), -h(\bar{x}))$ is a solution to the following optimization problem:

$$\begin{array}{lll} \mathcal{AP} & \text{Minimize} & f_0(x) \\ & \text{subject to} & f_n(x) - u_n = 0, \ n = 1, 2, \ldots, p, \\ & & f_n(x) + v_{n-p} = 0, \ n = p+1, \ldots, p+q, \\ & & (x, u, v) \in C \times \text{graph } F. \end{array}$$

Since the usual norms in Euclidean spaces are Fréchet smooth, we can apply the necessary conditions of Theorem 3.1 to problem $\mathcal{AP}$. Observing that in finite dimensional spaces weak-star and strong neighborhoods are the same, we have the following theorem.

THEOREM 4.7 (Fuzzy necessary conditions for MPEC). *Let $\bar{x}$ be a solution to problem $\mathcal{MPEC}$. Then, for any $\varepsilon > 0$, there exist $(x_0, f_0(x_0)) \in B_\varepsilon((\bar{x}, f_0(\bar{x})))$, $\tilde{x} \in B_\varepsilon(\bar{x})$, $\tilde{u} \in B_\varepsilon(g(\bar{x}))$, $\tilde{v} \in B_\varepsilon(-h(\bar{x}))$, $x_n \in B_\varepsilon(\bar{x})$, $\tau_n \in \{-1, 1\}$, and $\mu_n > 0$, $n = 1, 2, \ldots, p+q$, such that*

$$\begin{aligned} &((\tau_1\mu_1, \ldots, \tau_p\mu_p), (-\tau_{p+1}\mu_{p+1}, \ldots, -\tau_{p+q}\mu_{p+q})) \\ &\qquad \in N_F(\text{graph } F; (\tilde{u}, \tilde{v})) + \varepsilon B \end{aligned} \tag{23}$$

and

$$0 \in \partial_F f_0(x_0) + \sum_{n=1}^{p+q} \mu_n \partial_F(\tau_n f_n)(x_n) + N_F(C; \tilde{x}) + \varepsilon B. \tag{24}$$

Multiplying (23) and (24) by $\mu_0 = 1/(1 + \sum_{n=1}^{p+q} \mu_n)$ yields the following corollary.

COROLLARY 4.8. *Let $\bar{x}$ be a solution to problem $\mathcal{MPEC}$. Then, for any $\varepsilon > 0$, there exist $(x_0, f_0(x_0)) \in B_\varepsilon((\bar{x}, f_0(\bar{x})))$, $\tilde{x} \in B_\varepsilon(\bar{x})$, $\tilde{u} \in B_\varepsilon(g(\bar{x}))$, $\tilde{v} \in B_\varepsilon(-h(\bar{x}))$, $x_n \in B_\varepsilon(\bar{x})$, $\tau_n \in \{-1, 1\}$, and $\mu_n > 0$, $n = 0, 1, 2, \ldots, p+q$, with $\sum_{n=0}^{p+q} \mu_n = 1$, such that*

$$\begin{aligned} &((\tau_1\mu_1, \ldots, \tau_p\mu_p), (-\tau_{p+1}\mu_{p+1}, \ldots, -\tau_{p+q}\mu_{p+q})) \\ &\qquad \in N_F(\text{graph } F; (\tilde{u}, \tilde{v})) + \varepsilon B \end{aligned} \tag{25}$$

and

$$0 \in \mu_0 \partial_F f_0(x_0) + \sum_{n=1}^{p+q} \mu_n \partial_F(\tau_n f_n)(x_n) + N_F(C;\tilde{x}) + \varepsilon B. \tag{26}$$

To compare the necessary conditions derived above with existing results in the literature, we write them in terms of the limiting subdifferentials, limiting normal cones, and limiting coderivatives (see [13]). We start with the definitions of these objects.

DEFINITION 4.9. *Let X be a finite dimensional Banach space. First let $f : X \to R \cup \{+\infty\}$ be a lower semicontinuous function. Define*

$$\partial f(x) := \left\{ \lim_{k\to\infty} v_k : v_k \in \partial_F f(x_k),\ (x_k, f(x_k)) \to (x, f(x)) \right\}$$

and

$$\partial^\infty f(x) := \left\{ \lim_{k\to\infty} t_k v_k : v_k \in \partial_F f(x_k),\ t_k \to 0^+,\ (x_k, f(x_k)) \to (x, f(x)) \right\},$$

and call $\partial f(x)$ and $\partial^\infty f(x)$ the limiting subdifferential and singular subdifferential of f at x, respectively. Second, let S be a closed subset of X. Define

$$N(S;x) := \left\{ \lim_{k\to\infty} v_k : v_k \in N_F(S;x_k),\ S \ni x_k \to x \right\},$$

and call $N(S;x)$ the limiting normal cone of S at x. Finally, let $F : X \to 2^Y$ be a multifunction with closed graph, and let $y \in F(x)$. We define the limiting coderivative $\partial^* F(x;y) : Y^* \to 2^{X^*}$ of F at (x,y) by $x^* \in D^*F(x;y)(y^*)$ if and only if

$$(x^*, -y^*) \in N(\text{graph}F; (x,y)).$$

THEOREM 4.10 (Limiting necessary conditions for MPEC). *Let $\bar{x}$ be a solution to problem $\mathcal{MPEC}$. Then, either*

(A1) *there exist $\mu \in D^*F(g(\bar{x}); -h(\bar{x}))(\nu)$ with $\mu := (\mu_1, \ldots, \mu_p)$ and $\nu := (-\mu_{p+1}, \ldots, -\mu_{p+q})$, $\mu_0 \geq 0$, $\tau_n \in \{-1, 1\}, n = 1, 2, \ldots, p+q$, such that*

$$0 \in \mu_0 \partial f_0(\bar{x}) + \sum_{\{n:\mu_n \neq 0\}} \partial(\mu_n f_n)(\bar{x}) + \sum_{\{n:\mu_n = 0\}} \partial^\infty(\tau_n f_n)(\bar{x}) + N(C;\bar{x}),$$

or

(A2) *there exist $x_0^* \in \partial^\infty f_0(\bar{x})$, $x_n^* \in \partial^\infty(\tau_n f_n)(\bar{x})$ not all zero, where $\tau_n \in \{-1, 1\}$, $n = 1, 2, \ldots, p+q$, such that*

$$\sum_{n=0}^{p+q} x_n^* = 0.$$

Proof. For each natural number k, let $\varepsilon = \frac{1}{k}$ in Corollary 4.8. Then there exist $(x_0^k, f(x_0^k)) \in B_{\frac{1}{k}}((\bar{x}, f(\bar{x})))$, $\tilde{u}^k \in B_{\frac{1}{k}}(g(\bar{x}))$, $\tilde{v}^k \in B_{\frac{1}{k}}(-h(\bar{x}))$, $x_n^k \in B_\varepsilon(\bar{x})$, $\tau_n^k \in \{-1, 1\}$, and $\mu_n^k > 0, n = 0, 1, 2, \ldots, p+q$, with $\sum_{n=0}^{p+q} \mu_n^k = 1$, such that

$$\begin{aligned} &((\tau_1^k \mu_1^k, \ldots, \tau_p^k \mu_p^k), (-\tau_{p+1}^k \mu_{p+1}^k, \ldots, -\tau_{p+q}^k \mu_{p+q}^k)) \\ &\quad \in N_F(\text{graph } F; (\tilde{u}^k, \tilde{v}^k)) + \frac{1}{k} B \end{aligned} \tag{27}$$

and $\xi_0^k \in \partial_F f_0(x_0^k)$, $\xi_n^k \in \partial_F(\tau_n^k f_n)(x_n^k), n = 1, 2, \ldots, p+q$, and $\tilde{\xi}^k \in N_F(C; \tilde{x}^k)$ such that

$$\left\| \sum_{n=0}^{p+q} \mu_n^k \xi_n^k + \tilde{\xi}^k \right\| < \frac{1}{k}. \tag{28}$$

We will take limits as $k \to \infty$. Observe first that, passing to subsequences if necessary, we may assume that $\tau_n = \tau_n^k \in \{-1, 1\}$ are independent of k. Now we consider the following two cases.

The regular case, when the sequence $t^k = \|\xi_0^k\| + \sum_{n=1}^{p+q} \|\mu_n^k \xi_n^k\|$ is bounded. Without loss of generality we may assume that $\mu_0^k \xi_0^k \to x_0^*$, $\mu_0^k \to \mu_0$, $\mu_n^k \xi_n^k \to x_n^*$, $\tau_n \mu_n^k \to \mu_n$, $n = 1, \ldots, p+q$, and $\tilde{\xi}^k \to \tilde{x}^*$. Then we must have $x_0^* \in \mu_0 \partial f_0(\bar{x})$, $x_n^* \in \partial(\mu_n f_n)(\bar{x})$ for $\mu_n \neq 0$, $x_n^* \in \partial^\infty(\tau_n f_n)(\bar{x})$ for $\mu_n = 0$, $\tilde{x}^* \in N(C; \bar{x})$, $\mu \in D^*F(g(\bar{x}); -h(\bar{x}))(\nu)$ with $\mu := (\mu_1, \ldots, \mu_p)$ and $\nu := (-\mu_{p+1}, \ldots, -\mu_{p+q})$, and

$$\sum_{n=0}^{p+q} x_n^* + \tilde{x}^* = 0. \tag{29}$$

This is (A1).

The singular case, when the sequence $t^k = \|\xi_0^k\| + \sum_{n=1}^{p+q} \|\mu_n^k \xi_n^k\|$ is unbounded. Without loss of generality we may assume that $t^k \to \infty$, $\mu_n^k \xi_n^k / t^k \to x_n^*$, $n = 0, 1, 2, \ldots, p+q$, and $\tilde{\xi}^k / t^k \to \tilde{x}^*$. Then we must have $x_0^* \in \partial^\infty f(\bar{x})$, $x_n^* \in \partial^\infty(\tau_n f_n)(\bar{x})$, $n = 1, 2, \ldots, p+q$, $\tilde{x}^* \in N(C; \bar{x})$, and

$$\sum_{n=0}^{p+q} x_n^* + \tilde{x}^* = 0. \tag{30}$$

It is clear that $x_n^*, n = 0, \ldots, p+q$, are not all zero. This is (A2). □

If all the functions $f_n, n = 0, 1, \ldots, p+q$, are locally Lipschitz near $\bar{x}$, then $\partial^\infty(\tau_n f_n)(\bar{x}) = \{0\}$ for all $n = 0, 1, \ldots, p+q$. Then Theorem 4.10 takes the following much simpler form.

THEOREM 4.11. *Let $\bar{x}$ be a solution to problem $\mathcal{MPEC}$. Suppose that all the functions $f_n, n = 0, 1, \ldots, p+q$, are locally Lipschitz around $\bar{x}$. Then there exist $\mu \in D^*F(g(\bar{x}); -h(\bar{x}))(\nu)$ with $\mu := (\mu_1, \ldots, \mu_p)$ and $\nu := (-\mu_{p+1}, \ldots, -\mu_{p+q})$, $\mu_0 \geq 0$, $\tau_n \in \{-1, 1\}, n = 1, 2, \ldots, p+q$, such that*

$$0 \in \sum_{n=0}^{p+q} \partial(\mu_n f_n)(\bar{x}) + N(C; \bar{x}). \tag{31}$$

First order necessary conditions for mathematical programs with equilibrium constraints (also known as optimization problems with variational inequality constraints) are the subject of much research (see [12, 24, 19] and the references therein). The model in [19] is the most general so far, and [24] represents one approach to the derivation of necessary conditions for such problems with nonsmooth data. We briefly indicate how to recover the first order necessary conditions in [19, 24] from Theorems 4.10 and 4.11. In [19], Outrata discussed the case in which f_0 is Lipschitz and all the functions $f_n, n = 1, \ldots, p+q$, are smooth. Then $\partial(\mu_n f_n)(\bar{x}) = \mu_n f'(\bar{x}), n = 1, 2, \ldots, p+q$. Noticing that the constraint qualification condition (CQ) in [19] rules out the possibility that $\mu_0 = 0$ (so that we can always rescale to make $\mu_0 = 1$), Theorem 3.1 in [19]

follows directly from Theorem 4.11. In [24], Ye discusses the optimization problem with variational inequality constraints. The essential difficulties are contained in the following form of the problem:

$$\begin{aligned} \mathcal{VI} \quad & \text{minimize } h_0(y,z) \\ & \text{subject to } 0 \in h(y,z) + N(\Omega; z), \\ & (y,z) \in C. \end{aligned}$$

Here Ω is a closed convex set, and $N(\Omega;\cdot)$ is the convex normal cone of Ω. It is easy to see that problem $\mathcal{VI}$ is a special case of $\mathcal{MPEC}$ with $x = (y,z)$, $g(y,z) = z$, $F(z) = N(\Omega; z)$. Applying Theorem 4.10, we have the following necessary conditions that generalize the necessary optimality conditions in [24] by allowing h to be a continuous function and h_0 to be a lower semicontinuous function.

THEOREM 4.12. *Let $h = (h_1, \ldots, h_q)$ be a continuous function, and let h_0 be a lower semicontinuous function. Suppose that $(\bar{y}, \bar{z})$ is a solution to problem $\mathcal{VI}$. Then, either*

(A1) *there exist $\nu_0 \geq 0$, $\nu = (-\nu_1, \ldots, -\nu_q)$, $\tau_0 = 1, \tau_n \in \{-1, 1\}, n = 1, \ldots, q$, such that*

$$\begin{aligned} 0 \in & \sum_{\{n:\nu_n \neq 0\}} \partial(\nu_n h_n)(\bar{y}, \bar{z}) + \sum_{\{n:\nu_n = 0\}} \partial(\tau_n h_n)(\bar{y}, \bar{z}) \\ & + \{0\} \times D^* N(\Omega; \bar{y}; -h(\bar{y}, \bar{z}))(\nu) + N(C; (\bar{y}, \bar{z})) \end{aligned}$$

or

(A2) *there exist $x_0^* \in \partial^\infty h_0(\bar{y}, \bar{z})$, $x_n^* \in \partial^\infty(\tau_n h_n)(\bar{y}, \bar{z})$ not all zero, where $\tau_n \in \{-1, 1\}, n = 1, 2, \ldots, q$, such that*

$$\sum_{n=0}^{q} x_n^* = 0.$$

Acknowledgments. I am indebted to Professor A. Kruger for his careful reading of an early version of the paper and for his many valuable suggestions. I also thank two referees for their helpful comments.

REFERENCES

[1] J. M. BORWEIN, J. S. TREIMAN, AND Q. J. ZHU, *Necessary conditions for constrained optimization problems with semicontinuous and continuous data*, Trans. Amer. Math. Soc., 350 (1998), pp. 2409–2429.

[2] J. M. BORWEIN AND Q. J. ZHU, *Viscosity solutions and viscosity subderivatives in smooth Banach spaces with applications to metric regularity*, SIAM J. Control Optim., 34 (1996), pp. 1568–1591.

[3] J. M. BORWEIN AND Q. J. ZHU, *Variational analysis in non-reflexive spaces and applications to control problems with L^1 perturbations*, Nonlinear Anal., 28 (1997), pp. 889–915.

[4] J. M. BORWEIN AND Q. J. ZHU, *A survey of subdifferential calculus with applications*, Nonlinear Anal., 38 (1999), pp. 687–773.

[5] F. H. CLARKE, *Optimization and Nonsmooth Analysis*, John Wiley and Sons, New York, 1983; reprinted as Classics Appl. Math. 5, SIAM, Philadelphia, 1990.

[6] I. EKELAND, *On the variational principle*, J. Math. Anal. Appl., 47 (1974), pp. 324–353.

[7] A. D. IOFFE, *On subdifferentiablility spaces*, Ann. New York Acad. Sci., 410 (1983), pp. 107–119.

[8] A. D. IOFFE, *Necessary conditions for nonsmooth optimization*, Math. Oper. Res., 9 (1984), pp. 159–189.

[9] A. D. Ioffe, *Proximal analysis and approximate subdifferentials*, J. London Math. Soc., 41 (1990), pp. 175–192.
[10] A. Y. Kruger and B. S. Mordukhovich, *Extremal points and Euler equations in nonsmooth optimization*, Dokl. Akad. Nauk. BSSR, 24 (1980), pp. 684–687 (in Russian).
[11] P. D. Loewen, *Optimal Control via Nonsmooth Analysis*, CRM Lecture Notes Series, AMS Summer School on Control, CRM, Université de Montréal, 1992, American Mathematical Society, Providence, RI, 1993.
[12] Z. Q. Lou, J. S. Pang, and D. Ralph *Mathematical Programs with Equilibrium Constraints*, Cambridge University Press, Cambridge, UK, 1996.
[13] B. S. Mordukhovich, *Approximation Methods in Problems of Optimization and Control*, Nauka, Moscow, 1988 (in Russian; English translation to appear in Wiley-Interscience).
[14] B. S. Mordukhovich, *The extremal principle and its applications to optimization and economics*, in Optimization and Related Topics, A. Rubinov and B. Glover, eds., Appl. Optim. 47, Kluwer, Dordrecht, The Netherlands, 2001, pp. 343–369.
[15] B. S. Mordukhovich and Y. Shao, *Extremal characterizations of Asplund spaces*, Proc. Amer. Math. Soc., 124 (1996), pp. 197–205.
[16] B. S. Mordukhovich and B. Wang, *Necessary suboptimality and optimality conditions via variational principles*, SIAM J. Control Optim, to appear.
[17] H. V. Ngai and M. Théra, *On Necessary Conditions for Non-Lipschitz Optimization Problems*, preprint, Universite de Limoges, Limoges, France, 2000.
[18] H. V. Ngai and M. Théra, *Metric Regularity, Subdifferential Calculus, and Applications*, Set-Valued Anal., 9 (2001), pp. 187–216.
[19] J. V. Outrata, *A generalized mathematical program with equilibrium constraints*, SIAM J. Control Optim., 38 (2000), pp. 1623–1638.
[20] R. T. Rockafellar, *Proximal subgradients, marginal values and augmented Lagrangians in nonconvex optimization*, Math. Oper. Res., 6 (1981), pp. 424–436.
[21] J. S. Treiman, *Lagrange multipliers for nonconvex generalized gradients with equality, inequality, and set constraints*, SIAM J. Control Optim., 37 (1999), pp. 1313–1329.
[22] J. Warga, *Controllability and a multiplier rule for nondifferentiable optimization problems*, SIAM J. Control Optim., 16 (1978), pp. 803–812.
[23] J. Warga, *Optimization and controllability without differentiability assumptions*, SIAM J. Control Optim., 21 (1983), pp. 837–855.
[24] J. J. Ye, *Constraint qualifications and necessary optimality conditions for optimization problems with variational inequality constraints*, SIAM J. Optim., 10 (2000), pp. 943–962.
[25] J. J. Ye and Q. J. Zhu, *Perturbed differential inclusion problem with nonadditive L^1 perturbations and applications*, J. Optim. Theory Appl., 92 (1997), pp. 189–208.
[26] Q. J. Zhu, *Subderivatives and their applications*, in Proceedings of the International Conference on Dynamical Systems and Differential Equations, Vol. II, Southwest Missouri State University, Springfield, MO, 1996, pp. 379–394.
[27] Q. J. Zhu, *The equivalence of several basic theorems for subdifferentials*, Set-Valued Anal., 6 (1998), pp. 171–185.

SIAM J. OPTIM.
Vol. 12, No. 4, pp. 1048–1074

AN INTERIOR POINT CONSTRAINED TRUST REGION METHOD FOR A SPECIAL CLASS OF NONLINEAR SEMIDEFINITE PROGRAMMING PROBLEMS*

F. LEIBFRITZ† AND E. M. E. MOSTAFA‡

Abstract. In this paper, an interior point trust region algorithm for the solution of a class of nonlinear semidefinite programming (SDP) problems is described and analyzed. Such nonlinear and nonconvex programs arise, e.g., in the design of optimal static or reduced order output feedback control laws and have the structure of abstract optimal control problems in a finite dimensional Hilbert space. The algorithm treats the abstract states and controls as independent variables. In particular, an algorithm for minimizing a nonlinear matrix objective functional subject to a nonlinear SDP-condition, a positive definiteness condition, and a nonlinear matrix equation is considered. The algorithm is designed to take advantage of the structure of the problem. It is an extension of an interior point trust region method to nonlinear and nonconvex SDPs, with a special structure which applies sequential quadratic programming techniques to a sequence of barrier problems and uses trust regions to ensure robustness of the iteration. Some convergence results are given, and, finally, several numerical examples demonstrate the applicability of the considered algorithm.

Key words. interior point method, trust region method, nonlinear semidefinite program, nonconvex programming, primal method, static output feedback, optimal control

AMS subject classifications. 90C51, 90C22, 90C26, 93D99, 49N05, 65K05

PII. S1052623400375865

1. Introduction.

This work is concerned with the development of an interior point-based algorithm combined with a trust region strategy for a special class of nonlinear and nonconvex semidefinite programming (SDP) problems. Optimization problems of this type arise for several applications in system and control theory (see, i.e., [5], [26], [35], and the references therein). In particular, this paper was motivated by the problem of designing static or reduced order output feedback compensators for stabilizing a linear quadratic control system. These problems are important examples of difficult, and in general nonconvex, nonlinear control problems. They consist of determining static output feedback (SOF) matrices which minimize a nonlinear objective functional such that the SOF gain stabilizes the corresponding closed loop system (see, i.e., [3], [22], [28]). Problems of this type can be formulated as specially structured nonconvex and nonlinear matrix optimization problems including nonlinear SDP-constraints. In particular, they have the following general form:

$$\min_{F,L} \quad J(F,L) \quad \text{s.t.} \quad h(F,L)=0,\ \ Y(F,L)\prec 0,\ \ L\succ 0, \tag{1.1}$$

where $F\in\mathbb{R}^{p\times r}$, $L\in\mathbb{R}^{n\times n}$, $L=L^T$, and $p,r<n$. The functions $h,Y:\mathbb{R}^{p\times r}\times\mathbb{R}^{n\times n}\to\mathbb{R}^{n\times n}$, $J:\mathbb{R}^{p\times r}\times\mathbb{R}^{n\times n}\to\mathbb{R}$ are assumed to be at least twice continuously differentiable, and the nonlinear matrix function $Y(\cdot)$ is supposed to be symmetric. As indicated above, minimization problems of the form (1.1) often arise in the design of optimal SOF matrices. The structure of the equation $h(F,L)=0$ allows an explicit

*Received by the editors July 27, 2000; accepted for publication (in revised form) October 16, 2001; published electronically April 19, 2002.
http://www.siam.org/journals/siopt/12-4/37586.html
†University of Trier, FB–IV Department of Mathematics, D–54286 Trier, Germany (leibfr@msun7.uni-trier.de).
‡University of Alexandria, Faculty of Science, Department of Mathematics, Alexandria, Egypt (emostafa99@ yahoo.com).

composition of the optimization variables (F, L) into basic variables L and nonbasic variables F. This structure is analogous to the one exhibited by many discretized optimal control problems. In the language of those problems, F represents the controls, L represents the states, $h(F, L) = 0$ represents the state equation, and the nonlinear SDP-constraints

$$Y(F, L) \prec 0, \quad L \succ 0 \tag{1.2}$$

can be interpreted as state and control constraints. In (1.1) the optimization variable spaces are given by $\mathbb{R}^{p\times r}$ and $\mathbb{R}^{n\times n}$, respectively. Both spaces are finite dimensional Hilbert spaces if we choose the inner product by $\langle X, Z\rangle = Tr(X^T Z)$, where $Tr(\cdot)$ denotes the trace operator.

The main goal of this paper is to propose and analyze an algorithm for finding an approximate solution to (1.1). The difficulties in solving (1.1) arise from the fact that it is a nonlinear and nonconvex matrix optimization problem. It is composed of an objective functional, which is nonlinear in the matrix variables F and L, and a constraint set, which consists of a nonlinear SDP-condition in F and L, a positive definiteness condition on L, and a nonlinear matrix equation. Thus, (1.1) is a specially structured nonlinear and nonconvex SDP-problem. To our knowledge, there is no general algorithm available for solving nonlinear and nonconvex SDP-problems expect the QQP-algorithm developed by Jarre [18]. The major drawback of the QQP-method is the use of the QR-decomposition of the Jacobian of the nonlinear equality constraints for computing a search direction and the evaluation of the Hessian matrix in every QQP-iteration, which can be very time-consuming. This is another motivation for the development of an interior point-based algorithm for solving nonlinear SDP-problems of the form (1.1). In particular, using ideas of nonlinear interior point methods and usual SDP-approaches combined with trust region techniques, we construct an optimization solver for (1.1), which exploits the inherent structure of this problem class without evaluating the Hessian matrix explicitly. Generalization of modern interior point methods to general nonconvex programs has been recognized during the last few years; see, for example, Byrd, Gilbert, and Nocedal [6], Conn, Gould, Orban, and Toint [8], Dennis, Heinkenschloss, and Vicente [11], El-Bakry, Tapia, Tsuchiya, and Zhang [13], Forsgren and Gill [14], Gay, Overton, and Wright [15], Vanderbei and Shanno [36], and the references therein. Moreover, trust region methods have proved to be very successful and robust in solving nonlinear programming problems (see, i.e., [6], [9], [8], [10], [11], [29], [37]). Finally, there is a huge number of papers dealing with linear (convex) SDP-problems; see, for example, [1], [2], [30], [35], and the references therein.

Following the strategy of interior point and SDP-methods, we associate with (1.1) the following barrier problem in the matrix variables F and L:

$$\min_{F,L} \quad \Phi^\mu(F, L) = J(F, L) - \mu[\log\det(L) + \log\det(-Y(F, L))] \quad \text{s.t.} \quad h(F, L) = 0, \tag{1.3}$$

where $\mu > 0$ and L, $-Y(F, L)$ are (implicitly) assumed to be positive definite. The main goal of this paper is to produce a constrained trust region (CTR) algorithm for finding an approximate solution of the barrier problem (1.3), for fixed μ, that is tailored to the structure of the problem and effectively enforces the positive definiteness conditions $L \succ 0$, $-Y(F, L) \succ 0$ of the (nonlinear) SDP-constraints. This algorithm can be applied repeatedly to problem (1.3), for decreasing values of μ, to approximate the solution of the original nonlinear SDP-problem (1.1). The CTR approach for

solving (1.3) can be considered as a variant of a sequential quadratic programming method, which has a good global and local convergence behavior. In our approach, the quadratic trust region subproblem decomposes into two trust region subproblems that are easier to solve. In particular, we use a tangent space method for solving the trust region subproblems (see, i.e., [6], [10], [11], [28]).

The organization of this paper is as follows. In subsection 1.1 we state the basic problem structure and assumptions which are needed to develop the algorithm. Then, in subsection 1.2, we discuss the framework of the interior point (barrier) method applied to (1.1). In subsection 1.3 we give a short overview of (static) output feedback problems arising in system and control theory, which justifies the formulation of the general nonlinear SDP-problem (1.1).

The main part of this paper is contained in section 2. In this section we discuss in detail a CTR method for solving the barrier problem (1.3). In particular, we use a so-called tangent space approach. Therein, we exploit the structure of the problem for decomposing the trial step into a quasi-normal and a tangential component as done in [28]. We include the SDP-conditions explicitly in our CTR subproblems by imposing a fraction rule which is similar to that of [6].

In sections 2.1 and 2.2 we state in detail how we compute the quasi-normal and the tangential step during the CTR algorithm for solving barrier problem (1.3). Thereafter, in subsection 2.3, we consider the overall CTR method, and in subsection 2.4 we discuss the main global convergence result for this algorithm class.

After deriving the CTR approach for solving (1.3), in section 3 we consider the whole interior point constrained trust region (IPCTR) algorithm for computing an optimal solution of the nonlinear SDP-problem (1.1) and establish a global convergence result for IPCTR; i.e., we ensure that any accumulation point of the generated sequence of IPCTR is a first order Karush–Kuhn–Tucker (KKT) point of (1.1).

Finally, in the last section we illustrate the numerical performance of the IPCTR algorithm by using test examples from the engineering literature.

Notation: Throughout this paper, $\langle \cdot, \cdot \rangle$ denotes the inner product defined by $\langle M, Z \rangle = Tr(M^T Z)$, where $Tr(\cdot)$ is the trace operator, and $\| \cdot \|$ denotes the Frobenius norm given by $\|M\| = \langle M, M \rangle^{\frac{1}{2}}$, while other norms will be specified, e.g., the 2-norm $\| \cdot \|_2$. For a matrix $M \in \mathbb{R}^{m \times m}$ we use the notation $M \succ 0$, $M \succeq 0$, $M \prec 0$, $M \preceq 0$ if it is positive definite, positive semidefinite, negative definite, negative semidefinite, respectively. For a twice differentiable mapping $g : \mathcal{U} \times \mathcal{V} \to \mathcal{W}$ ($\mathcal{U}, \mathcal{V}, \mathcal{W}$ being finite dimensional Hilbert spaces), we denote by $g_U(U, V)$, $g_V(U, V)$, $g_{UU}(U, V)$, etc., the first and second partial derivatives of g with respect to U, V at $(U, V) \in \mathcal{U} \times \mathcal{V}$, respectively. The notations $\nabla g_U(\cdot)$, $\nabla^2 g_{UU}(\cdot)$, and so on will be used in the case that g is real valued, respectively. Moreover, the notation $g_U(\cdot)H$ is used when a linear operator $g_U(\cdot)$ is applied to an element $H \in \mathcal{U}$. Furthermore, by $g_U^*(\cdot)$ we denote the adjoint operator of $g_U(\cdot)$. The space of linear and bounded operators from $\mathcal{U}$ to $\mathcal{V}$ is denoted by $\mathcal{L}(\mathcal{U}, \mathcal{V})$. Finally, the matrices A_F and Q_F are always used as abbreviations for $A + BFC$ and $C^T F^T RFC + Q$, respectively.

1.1. Problem structure and assumptions. In this section we discuss the basic problem structure to develop the algorithm and state the basic assumptions which we impose on the nonlinear SDP-problem (1.1) and the corresponding barrier problem (1.3). We will also introduce some fundamental quantities that are subsequently needed. For constructing the algorithm, we use a technique which is often considered in the applications of optimal control problems and other SQP-based methods if the variables can be decomposed into states and controls. In particular, if the mapping

$h_L(F,L)$ is invertible, the linearized equality constraint equation possesses a unique solution. This is the basic structure that we will extract in the subsequent sections for developing the algorithm. Note that the same problem structure has been already used in the past; see, for example, [21], [20], [11], [28], and the references therein.

During the whole paper we use the following basic assumptions.

ASSUMPTION 1.1.

(i) $J : \mathbb{R}^{p\times r} \times \mathbb{R}^{n\times n} \to \mathbb{R}$, $h, Y : \mathbb{R}^{p\times r} \times \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ *are at least twice continuously differentiable,* $F \in \mathbb{R}^{p\times r}$, $L \in \mathbb{R}^{n\times n}$, $L = L^T$, $p, r < n$, *and* $Y(\cdot) = Y(\cdot)^T$.

(ii) *There exist* $F_0 \in \mathbb{R}^{p\times r}$ *and* $L_0 \in \mathbb{R}^{n\times n}$ *such that* $(F_0, L_0) \in \mathcal{F}_s$, *where*

$$\mathcal{F}_s := \{(F,L) \mid Y(F,L) \prec 0, \quad L \succ 0\}. \tag{1.4}$$

(iii) *For given* $(F,L) \in \mathcal{F}_s$, *the mapping* $h_L(F,L)$ *is invertible.*

Assumption 1.1(ii) ensures that barrier problem (1.3) is well defined and that $\mathcal{F}_s$ is nonempty. Moreover, the invertibility of $Y(F,L)$ follows from the definition of $\mathcal{F}_s$. Assumptions 1.1(i) and (iii) guarantee the differentiability of the barrier functional $\Phi^\mu(F,L)$ for every $\mu > 0$.

The Lagrangian function associated with barrier problem (1.3) is defined by

$$\ell^\mu(F,L,K) = \Phi^\mu(F,L) + \langle K, h(F,L)\rangle, \tag{1.5}$$

where $K \in \mathbb{R}^{n\times n}$ denotes the Lagrange multiplier for the constraint $h(F,L) = 0$.

The linearized equality constraints are given by

$$h_F(F,L)\Delta F + h_L(F,L)\Delta L + h(F,L) = 0, \tag{1.6}$$

where $h_F(F,L)\Delta F$ and $h_L(F,L)\Delta L$ are the partial derivatives of h applied to $\Delta F \in \mathbb{R}^{p\times r}$ and $\Delta L \in \mathbb{R}^{n\times n}$, respectively. If the mapping $h_L(F,L)$ is invertible, then (1.6) implies

$$\Delta L = -h_L^{-1}(F,L)h_F(F,L)\Delta F - h_L^{-1}(F,L)h(F,L),$$

which leads to the following natural representation of the step $S = (\Delta L, \Delta F) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{p\times r}$:

$$(\Delta L, \Delta F) = T(F,L)\Delta F + (-h_L^{-1}(F,L)h(F,L), 0), \tag{1.7}$$

where 0 is the zero matrix and, denoting by $\mathcal{I}$ the identity mapping, the linear operator

$$T(F,L) = (T_1(F,L), \mathcal{I}) = (-h_L^{-1}(F,L)h_F(F,L), \mathcal{I}) \in \mathcal{L}(\mathbb{R}^{p\times r}, \mathbb{R}^{n\times n} \times \mathbb{R}^{p\times r})$$

characterizes the null space of $h'(F,L)$, which can be represented by

$$\mathcal{N}(h'(\cdot)) = \{(\Delta L, \Delta F) \mid h_F(\cdot)\Delta F + h_L(\cdot)\Delta L = 0\} = \{T(\cdot)\Delta F, \Delta F \in \mathbb{R}^{p\times r}\}. \tag{1.8}$$

Thus, the step S can be decomposed into a so-called quasi-normal component S^n and tangential component S^t. Note that S^n is of the form $S^n = (\Delta L^n, 0)$. Therefore, the displacement along S^n is made only in the L-variables. The tangential component is in the null space of h', and it is of the form $S^t = (T_1(F,L)\Delta F, \Delta F)$. Hence, the displacement along S^t can be made in the F-variables, where the L-part of S^t, also denoted by ΔL^t, depends on ΔF.

1.2. Outline of the interior point method. We begin by describing the framework of our algorithm. It is basically a sequential minimization of the logarithmic barrier function $\Phi^\mu(F, L)$, defined in (1.3), subject to a nonlinear matrix equality constraint; i.e., we propose to (approximately) solve (1.3) for a sequence of barrier parameters $\mu_j > 0$, $j = 0, 1, 2, \ldots$, whose limiting value is zero. An approximate minimizer of problem (1.3), (F_{j+1}, L_{j+1}), defines an outer iterate, and the associated adjustment of the barrier parameter and other tolerances defines the outer iteration. Outer iterations will be indexed by the subscript $j \geq 0$. Each outer iterate (F_{j+1}, L_{j+1}) is computed by using an appropriate inner iteration algorithm to approximately solve (1.3), with a corresponding sequence of inner iterates $\{(F_k, L_k)\}$.

Our method mainly works with primal variables, while dual variables are computed from the primal iterate via the solution of a certain equation. We do this because for nonconvex problems the advantage of using dual variables directly may be outdone by the difficulties arising from loss of primal-dual symmetry and from the loss of monotonicity properties which are present in convex primal-dual formulations.

The overall algorithm for decreasing values of the barrier parameter can be described as follows. Assume that for given $\mu_j > 0$ a (strictly) feasible point (F_0, L_0) for $\Phi^{\mu_j}(\cdot, \cdot)$ is known, i.e., $(F_0, L_0) \in \mathcal{F}_s$. Note that we do not suppose that (F_0, L_0) satisfies the equality constraints of (1.3). Then, the method consists of finding successively approximate solutions of the corresponding barrier problems. We formally state the outer iteration as Algorithm 1.1.

ALGORITHM 1.1. *Choose $\mu_0 > 0$, $\epsilon_0 > 0$, $a, b \in (0, 1)$, and $(F_0, L_0) \in \mathcal{F}_s$. Set $j = 0$.*

For $j = 0, 1, 2, \ldots$ do (Outer iteration)

(Inner iteration) For $\mu = \mu_j > 0$ minimize barrier problem (1.3) starting from $(F_0, L_0) := (F_j, L_j) \in \mathcal{F}_s$. Stop this inner algorithm as soon as an inner iterate (F_k, L_k) satisfies

$$||\nabla \ell_F^{\mu_j}(F_k, L_k, K_k)|| + ||h(F_k, L_k)|| \leq \epsilon_j,$$

where

$$\nabla \ell_F^\mu(F, L, K) = \nabla J_F(F, L) - \mu Y_F^*(F, L) Y^{-1}(F, L) + h_F^*(F, L) K \tag{1.9}$$

and K is the solution of the adjoint equation

$$\nabla \ell_L^\mu(F, L, K) = h_L^*(F, L) K + \nabla J_L(F, L) - \mu M(F, L) = 0, \tag{1.10}$$

$$M(F, L) = L^{-1} + Y_L^*(F, L) Y^{-1}(F, L). \tag{1.11}$$

Choose $\mu_{j+1} \in (0, a\mu_j)$, $\epsilon_{j+1} \in (0, b\epsilon_j)$ and set $(F_{j+1}, L_{j+1}) = (F_k, L_k)$.

All the iterates generated by this algorithm form a single sequence $\{(F_j, L_j)\}_{j \geq 0}$. Since each inner iteration consists of minimizing (1.3) for fixed $\mu_j > 0$, the (approximate) solution of the jth barrier problem, denoted by (F_{j+1}, L_{j+1}), also satisfies the SDP-constraints. For more detail, we refer to the following sections. Therein we consider a trust region method for finding an approximate solution of each barrier problem, which is tailored to the special structure of the nonlinear SDP-problem considered in this paper.

1.3. Application/background: Output feedback problems. The design of stabilizing feedback control laws has been an active research area of the control community for several decades. In this area, the optimal static output feedback problem

is an important example of a difficult (in general nonconvex and nonlinear) control problem. The optimal control problem consists of finding an SOF gain matrix which minimizes an infinite time quadratic objective function in such a way that the feedback gain yields an asymptotically stable closed loop system. Since the resulting nonlinear and nonconvex matrix optimization problems are of considerable importance, there exist various algorithms for obtaining a numerical solution. Several researchers have refined the algorithms by using techniques from mathematical optimization such as step size rules, iterative solvers for solving the first order optimality system, Newton's method, quasi-Newton method, and interior point-based approaches. For example, the algorithms stated in [3] are in general first order methods. The development of higher order methods was introduced by [34]. The authors in [34] used Newton's method combined with an Armijo line search rule as a globalization strategy. In [31] the optimal SOF problem was solved by a quasi-Newton method. Moreover, the author in [24] and [22] developed special interior point algorithms for tackling such problem classes numerically.

Recently, Leibfritz and Mostafa [28] used a CTR approach for solving the optimal SOF problem. However, to our knowledge, neither they nor any of the above authors considered interior point trust region methods, which can be used in general for solving nonlinear and possibly nonconvex optimization problems. This was one of our main motivations in developing an interior point trust region-based algorithm for solving the optimal SOF design problem, which can be formulated as a special nonlinear and nonconvex SDP-minimization problem. As shown in [27], the optimal SOF design problem is a special case of the nonlinear SDP-problem (1.1). In particular, defining $J(F,L) = \langle L, Q_F\rangle$, $h(F,L) = A_F L + LA_F^T + P$, and $Y(F,L) = A_F L + LA_F^T$, problem (1.1) reduces to

$$\min_{F,L} \quad Tr(LQ_F) \quad \text{s.t.} \quad A_F L + LA_F^T + P = 0, \quad A_F L + LA_F^T \prec 0, \quad L \succ 0, \tag{1.12}$$

where $A_F = A + BFC$, $Q_F = C^T F^T RFC + Q$. Therein we assume that the data matrices A, B, C, Q, R are appropriately dimensioned real constant matrices. In this case F is the so-called static output feedback controller matrix, A_F denotes the closed loop system matrix, and $h(F,L) = 0$ is a Lyapunov equation in the unknowns F and L. Moreover, the SDP-conditions $Y(F,L) = A_F L + LA_F^T \prec 0$, $L \succ 0$, represent the stability of a linear system of the form $\dot{x}(t) = A_F x(t)$, $x(0) = x_0$. A further example of an SOF design problem, which falls within the class of problem (1.1), is the so-called SOF $\mathcal{H}_2/\mathcal{H}_\infty$ problem. In its simplest form, we can define the problem functions of (1.1) by $J(F,L) = \langle L, Q_F\rangle$ and

$$\begin{aligned} h(F,L) &= A_F L + LA_F^T + P + \frac{1}{\gamma^2} LQ_F L, \\ Y(F,L) &= \left(A_F + \frac{1}{\gamma^2} LQ_F\right) L + L\left(A_F + \frac{1}{\gamma^2} LQ_F\right)^T, \end{aligned}$$

where $\gamma > 0$ is given (see [22] and the references therein). Note that in the above SOF problems, Assumptions 1.1(i) and (iii) are always satisfied for every $(F,L) \in \mathcal{F}_s$. In particular, one can show that the mapping $h_L(F,L)$ is bijective on $\mathcal{F}_s$ (see, i.e., [28, Lemma 3.2]).

Most of the available algorithms require an initial stabilizing SOF controller matrix, which can demand great computational effort. Recently, linear matrix inequalities (LMIs) have attained much attention in control engineering (see, i.e., [5] and the

references therein), since many control problems can be formulated in terms of LMIs. For example, this includes $\mathcal{H}_\infty$ (see, i.e., [7]), $\mathcal{H}_2$ (see, i.e., [16]), and mixed $\mathcal{H}_2/\mathcal{H}_\infty$ (see, i.e., [19]) design problems. However, the resulting controllers are state feedback or of order n equal to the plant. The difficulties arise if we want to design SOF controllers. In this case, the corresponding control problem results in a special structured nonconvex SDP-problem (see, i.e., [23], [26], [22], and the references therein). In particular, the problem of finding an SOF controller can be reduced to an SDP-problem with a nonconvex objective function over a convex set containing LMIs. For solving this problem, the author in [23] derived an algorithm, the so-called sequential linear programming matrix method (SLPMM). In our numerical testing we have used SLPMM for determining a feasible point for Φ^μ, i.e., a matrix pair (F, L) satisfying the nonlinear SDP-constraints (1.2) of the corresponding SOF design problem. Then, if not otherwise stated, we have taken the result of SLPMM as a starting point for our interior point trust region method. For more details on SOF design problems, we refer the interested reader to [22], [23], [26], [27], and the references therein.

During the whole paper we make the following assumptions on the constant data matrices if we consider SOF design problems.

ASSUMPTION 1.2. *Let $A, P, Q \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times p}$, $C \in \mathbb{R}^{r\times n}$, $R \in \mathbb{R}^{p\times p}$ be given with $P \succ 0$, $Q \succeq 0$, $R \succ 0$, and C having full rank $r \leq n$.*

2. Constrained trust region method for the barrier problem. In this section we give a detailed description of the algorithm for solving the barrier problem (1.3). We develop a CTR method for solving this problem class. In particular, this trust region method is used as the inner iteration procedure in the interior point Algorithm 1.1 for determining an approximate solution of nonlinear SDP-problem (1.1). The trust region algorithm that we propose generates a sequence $\{(F_k, L_k)\}$, and each pair (F_k, L_k) is strictly feasible with respect to the nonlinear SDP-constraints, $(F_k, L_k) \in \mathcal{F}_s$. At iteration k the pair (F_k, L_k) is given, and we need to compute a trial step $S_k = (\Delta L_k, \Delta F_k)$. The step S_k is computed by solving some trust region subproblems. Then the new point is tested using some merit function to decide whether it is a better approximation to a solution of (1.3). The trust region radius is then adjusted, and a new quadratic model of (1.3) is formed. Throughout the section, we assume that $\mu > 0$ is fixed and an (initial) pair $(F, L) \in \mathcal{F}_s$ is known; i.e., we can choose this pair as the solution of the previous barrier problem.

For solving the trust region subproblems, we use a tangent space approach (see, i.e., [11], [10]). In this approach the trial step is determined as $S_k = S^n + S^t$, where S^n denotes the quasi-normal component and S^t is the tangential component with respect to the null space of the constrained Jacobian. The role of S^n is to move towards feasibility, and the role of S^t is to move towards optimality. However, note that we do not require feasibility with respect to the nonlinear equality constraint $h(F, L) = 0$ of (1.3).

For deriving the CTR subproblem, we need the following lemma, which states the first and second order partial derivatives of the barrier and the Lagrangian functional applied to elements of the domain space. Since the proof of this result is straightforward, we omit it.

LEMMA 2.1. *Let $(F, L) \in \mathcal{F}_s$, $K \in \mathbb{R}^{n\times n}$, $L \in \mathbb{R}^{n\times n}$, and $\mu > 0$ be given. Then the barrier function Φ^μ of* (1.3) *and the Lagrangian functional ℓ^μ are twice continuously differentiable on $\mathcal{F}_s$. Moreover, the first and second order partial derivatives of Φ^μ and the Lagrangian function* (1.5) *applied to elements $\Delta F \in \mathbb{R}^{p\times r}$ and $\Delta L \in \mathbb{R}^{n\times n}$*

are given by

$$\begin{aligned}
\Phi_F^\mu(\cdot)\Delta F &= \langle \Delta F, \nabla \Phi_F^\mu(\cdot)\rangle = \langle \Delta F, \nabla J_F(\cdot) - \mu Y_F^*(\cdot)Y^{-1}(\cdot)\rangle,\\
\Phi_L^\mu(\cdot)\Delta L &= \langle \Delta L, \nabla \Phi_L^\mu(\cdot)\rangle = \langle \Delta L, \nabla J_L(\cdot) - \mu M(\cdot)\rangle,\\
\ell_F^\mu(\cdot)\Delta F &= \langle \Delta F, \nabla \ell_F^\mu(\cdot)\rangle = \langle \Delta F, \nabla \Phi_F^\mu(\cdot) + h_F^*(\cdot)K\rangle,\\
\ell_L^\mu(\cdot)\Delta L &= \langle \Delta L, \nabla \ell_L^\mu(\cdot)\rangle = \langle \Delta L, \nabla \Phi_L^\mu(\cdot) + h_L^*(\cdot)K\rangle,\\
\ell_{FF}^\mu(\cdot)(\Delta F, \Delta F) &= \langle \Delta F, \nabla^2 \ell_{FF}^\mu(\cdot)\Delta F\rangle,\\
\ell_{FL}^\mu(\cdot)(\Delta F, \Delta L) &= \langle \Delta F, \nabla^2 \ell_{FL}^\mu(\cdot)\Delta L\rangle = \ell_{LF}^\mu(\cdot)(\Delta L, \Delta F),\\
\ell_{LL}^\mu(\cdot)(\Delta L, \Delta L) &= \langle \Delta L, \nabla^2 \ell_{LL}^\mu(\cdot)\Delta L\rangle,
\end{aligned}$$

where $M(F,L)$ *is defined by* (1.11) *and*

$$\begin{aligned}
\nabla^2 \ell_{FF}^\mu(\cdot)\Delta F &= \nabla^2 J_{FF}(\cdot)\Delta F + (h_{FF}(\cdot)\Delta F)^* K\\
&\quad + \mu Y_F^*(\cdot)(Y^{-1}(\cdot)Y_F(\cdot)\Delta F Y^{-1}(\cdot)) - \mu(Y_{FF}(\cdot)\Delta F)^* Y^{-1}(\cdot),\\
\nabla^2 \ell_{FL}^\mu(\cdot)\Delta L &= \nabla^2 J_{FL}(\cdot)\Delta L + (h_{FL}(\cdot)\Delta L)^* K\\
&\quad + \mu Y_F^*(\cdot)(Y^{-1}(\cdot)Y_L(\cdot)\Delta L Y^{-1}(\cdot)) - \mu(Y_{FL}(\cdot)\Delta L)^* Y^{-1}(\cdot),\\
\nabla^2 \ell_{LL}^\mu(\cdot)\Delta L &= \nabla^2 J_{LL}(\cdot)\Delta L + (h_{LL}(\cdot)\Delta L)^* K\\
&\quad + \mu L^{-1}\Delta L L^{-1} + \mu Y_L^*(\cdot)(Y^{-1}(\cdot)Y_L(\cdot)\Delta L Y^{-1}(\cdot)) - \mu(Y_{LL}(\cdot)\Delta L)^* Y^{-1}(\cdot).
\end{aligned}$$

Using Lemma 2.1, the necessary optimality conditions for the barrier problem (1.3) are given by

$$\nabla \ell_F^\mu(F,L,K) = \nabla J_F(F,L) - \mu Y_F^*(F,L)Y^{-1}(F,L) + h_F^*(F,L)K = 0, \tag{2.1}$$

$$\nabla \ell_L^\mu(F,L,K) = h_L^*(F,L)K + \nabla J_L(F,L) - \mu(L^{-1} + Y_L^*(F,L)Y^{-1}(F,L)) = 0, \tag{2.2}$$

$$h(F,L) = 0, \tag{2.3}$$

and implicitly $L \succ 0$, $Y(F,L) \prec 0$. Observe that, by setting $\mu = 0$ in (2.1) and (2.2), we obtain the KKT conditions for nonlinear SDP-problem (1.1).

For given $(F,L) \in \mathcal{F}_s$ and $K \in \mathbb{R}^{n\times n}$ we introduce a quadratic model

$$q(\Delta F, \Delta L) = \ell^\mu + \ell_F^\mu \Delta F + \ell_L^\mu \Delta L + \frac{1}{2}\left(\ell_{FF}^\mu(\Delta F, \Delta F) + 2\ell_{FL}^\mu(\Delta F, \Delta L) + \ell_{LL}^\mu(\Delta L, \Delta L)\right) \tag{2.4}$$

of $\ell^\mu(F + \Delta F, L + \Delta L, K)$ about the current iterate (F,L,K), where $\Delta F \in \mathbb{R}^{p\times r}$, $\Delta L \in \mathbb{R}^{n\times n}$, and $\ell^\mu = \ell^\mu(F,L,K)$. Using the quadratic model (2.4) of ℓ^μ, we can define the CTR subproblem. For example, to determine $(\Delta F, \Delta L)$ from the current point (F,L,K), we minimize the quadratic model q subject to linearized equality constraints and a trust region constraint for restricting the step. The trust region constraint does not prevent the new variables $(F + \Delta F, L + \Delta L)$ from satisfying the SDP-constraints unless the trust region radius is sufficiently small. Since it is not desirable to impede progress of the iteration by employing small trust regions, we explicitly bound the F- and L-variables by imposing a fraction rule similar to (see [6])

$$L + \Delta L \succeq (1-\sigma)L \succ 0, \quad Y(F + \Delta F, L + \Delta L) \preceq (1-\sigma)Y(F,L),$$

where $\sigma \in (0,1)$ is chosen close to one and (F,L) is given such that $L \succ 0, Y(F,L) \prec 0$.

Hence, at each inner iteration, we solve the following CTR subproblem:

$$\begin{aligned} &\min \quad q(\Delta F, \Delta L) \\ \text{s.t.} \quad &h_F(F,L)\Delta F + h_L(F,L)\Delta L + h(F,L) = 0, \\ &Y(F+\Delta F, L+\Delta L) \preceq (1-\sigma)Y(F,L), \\ &\Delta L \succeq -\sigma L, \quad ||(\Delta F, \Delta L)|| \leq \delta, \end{aligned} \tag{2.5}$$

where $\delta > 0$ is the current trust region radius. However, in this straightforward approach it is well known that the linearized constraints and the trust region constraint may be inconsistent in such a way that the CTR subproblem (2.5) does not have a solution (see, i.e., [10]). To overcome this difficulty, two main approaches have been introduced in the trust region literature, the tangent space approach and the full space approach.

We use the tangent space approach in which each trust region problem of the form (2.5) is decomposed into two trust region subproblems; in particular, the quasi-normal problem for obtaining S^n and the tangential problem for computing S^t with respect to the null space of the constraint Jacobian. For building a basis of the null space of $h'(F,L)$, we use a technique which is often considered in the applications of optimal control problems (see, for example, [11], [28], and the references therein). Especially, as discussed in subsection 1.1, the solutions of the linearized constraints (1.6) of h are of the form $(\Delta L, \Delta F) = S^n + T(F,L)\Delta F$ if the mapping $h_L(F,L)$ is invertible. As a result of (1.8) the linearized equality constraints (1.6) can be split into two equations.

LEMMA 2.2. *Let $(F,L) \in \mathcal{F}_s$ be given and $h_L(F,L)$ be invertible; then the linearized constraints* (1.6) *decompose into the following equations:*

$$h_L(F,L)\Delta L^n + h(F,L) = 0, \tag{2.6}$$

$$h_L(F,L)\Delta L^t + h_F(F,L)\Delta F = 0. \tag{2.7}$$

Proof. By using the decomposition (1.7), constraints (1.6) can be restated as

$$h_L(F,L)\Delta L^t + h_F(F,L)\Delta F + h_L(F,L)\Delta L^n + h(F,L) = 0. \tag{2.8}$$

Since $T(F,L)\Delta F$ is the null space operator of $h'(F,L)$, (1.8) implies

$$h_L(F,L)\Delta L^t + h_F(F,L)\Delta F = 0 \quad \forall\, \Delta F \in \mathbb{R}^{p\times r},$$

which corresponds to (2.7). Thus, (2.8) reduces to (2.6). ∎

As a result of the decomposition (1.7) of S and Lemma 2.2, the CTR subproblem (2.5) can be split into two subproblems which are easier to solve. The first one gives the quasi-normal component S^n, while the second one yields the tangential component S^t of the trial step $S = S^n + S^t$. Moreover, note that the L-part ΔL^t of the tangential component defined by (2.7) depends on ΔF. In particular, the solution is given by $\Delta L^t = T_1(F,L)\Delta F$. Therefore, we always interpret ΔL^t as a matrix function depending on ΔF.

2.1. Computation of the quasi-normal component. At each step of the CTR algorithm for the barrier problem we solve a trust region subproblem for obtaining the quasi-normal component. Let δ be the current trust region radius and $(F,L) \in \mathcal{F}_s$ be given. The quasi-normal component is required to have the form $S^n = (\Delta L^n, 0)$ and is related to the trust region subproblem for the linearized constraints. Thus, the displacement along S^n is made only in the L-variables, and as a

consequence, (L, F) and $(L, F) + S^n$ have the same F-components. Therefore, the quasi-normal step is computed by solving the following normal subproblem:

$$\min \frac{1}{2}||h_L(F,L)\Delta\hat{L}^n + h(F,L)||^2 \quad \text{s.t.} \quad ||\Delta\hat{L}^n|| \le \omega\delta, \quad \Delta\hat{L}^n \succeq -\sigma\omega L,$$

where $\sigma, \omega \in (0,1)$ are given scalars.

One approach to solving the quasi-normal problem is the following: In the first step, compute a solution $\Delta\hat{L}^n$ of (2.6), and in a second step, control the size of $\Delta\hat{L}^n$ such that it lies inside of the trust region and such that all eigenvalues of $\Delta\hat{L}^n + \sigma\omega L$ are nonnegative. This can be done by determining the solution of the one-dimensional minimization problem

$$\min_{\beta>0} \frac{1}{2}||\beta h_L(F,L)\Delta\hat{L}^n + h(F,L)||^2 \quad \text{s.t.} \quad \beta||\Delta\hat{L}^n|| \le \omega\delta, \quad \beta\Delta\hat{L}^n \succeq -\sigma\omega L.$$

First we consider only the SDP-condition

$$\sigma\omega L + \beta\Delta\hat{L}^n \succeq 0 \tag{2.9}$$

for $\beta > 0$. Let $E^T E = L$ be the Cholesky factorization of $L \succ 0$. Then, if $\Delta\hat{L}^n \succeq 0$, the above SDP-condition is fulfilled for all $\beta > 0$. Thus, suppose that $\Delta\hat{L}^n$ is not positive semidefinite. In this case, (2.9) is equivalent to finding $\beta > 0$ such that

$$\sigma\omega I + \beta E^{-T}\Delta\hat{L}^n E^{-1} \succeq 0.$$

With this change, the problem of determining $\beta > 0$ satisfying (2.9) is equivalent to the step length problem in ordinary linear programming; i.e., we obtain (see, for example, [32, section 4.2])

$$\beta = -\frac{\sigma\omega}{\lambda_{\min}(E^{-T}\Delta\hat{L}^n E^{-1})}, \tag{2.10}$$

where $\lambda_{\min}(E^{-T}\Delta\hat{L}^n E^{-1}) < 0$ denotes the smallest eigenvalue of $E^{-T}\Delta\hat{L}^n E^{-1}$.

With this observation, an approximate solution of this minimization problem is given by

$$\beta = \begin{cases} 1 & \text{if } ||\Delta\hat{L}^n|| \le \omega\delta, \Delta\hat{L}^n \succeq 0, \\ \frac{\omega\delta}{||\Delta\hat{L}^n||} & \text{if } ||\Delta\hat{L}^n|| > \omega\delta, \Delta\hat{L}^n \succeq 0, \\ \min\left(1, \frac{\omega\delta}{||\Delta\hat{L}^n||}, -\frac{\sigma\omega}{\lambda_{\min}(E^{-T}\Delta\hat{L}^n E^{-1})}\right) & \text{if } \Delta\hat{L}^n \not\succeq 0. \end{cases} \tag{2.11}$$

Finally, the component $\Delta\hat{L}^n$ is replaced by $\Delta L^n = \beta\Delta\hat{L}^n$, which is an approximate solution of the quasi-normal problem.

As outlined above, we do not have to solve the quasi-normal subproblem exactly. We only have to guarantee that the quasi-normal component is small close to feasible points and that it satisfies a form of Cauchy decrease condition. Assuming that (F, L) is in $\mathcal{F}_s$, which is an open set in $\mathbb{R}^{p\times r} \times \mathbb{R}^{n\times n}$, and that the matrix operator $h_L^{-1}(F, L)$ is uniformly bounded in $\mathcal{F}_s$, we can show the following important result.

LEMMA 2.3. *Let $\omega, \sigma \in (0,1)$, $\delta > 0$ be defined as above, and $(F, L) \in \mathcal{F}_s$. Suppose that there exist positive constants ν_1, ν_2 such that $||h_L^{-1}(F, L)|| \le \nu_1$ and $||E^{-1}|| \le \nu_2$ for all $(F, L) \in \mathcal{F}_s$, where E denotes the Cholesky factor of $L \succ 0$. Then the quasi-normal component*

$$S^n = (-\beta h_L^{-1}(F,L)h(F,L), 0) = (\Delta L^n, 0),$$

where $\beta > 0$ is defined by (2.11), *satisfies*

$$||S^n|| \leq \nu_1 ||h(F,L)|| \tag{2.12}$$

and

$$||h(F,L)||^2 - ||h(F,L) + h_L(F,L)\Delta L^n||^2 \geq \kappa ||h(F,L)|| \min(||h(F,L)||, \omega\delta, \omega\sigma) \tag{2.13}$$

for the positive constant $\kappa = \min(1, \frac{1}{\nu_1}, \frac{1}{\nu_1\nu_2^2})$.

Proof. Using $\beta \in (0,1]$, the boundedness of $h_L^{-1}(F,L)$, and $||S^n|| = ||\Delta L^n||$, we deduce that

$$||S^n|| \leq \beta ||h_L^{-1}(F,L)||\, ||h(F,L)|| \leq \nu_1 ||h(F,L)||.$$

Now, let us prove (2.13). Obviously, we have

$$\begin{aligned}
&||h(F,L)||^2 - ||h(F,L) + h_L(F,L)\Delta L^n||^2 \\
&\quad = ||h(F,L)||^2 - ||h(F,L) - \beta h_L(F,L)(h_L^{-1}(F,L)h(F,L))||^2 \\
&\quad = ||h(F,L)||^2 - ((1-\beta)||h(F,L)||)^2 \\
&\quad = \beta(2-\beta)||h(F,L)||^2 \;\geq\; \beta ||h(F,L)||^2.
\end{aligned}$$

Thus, we need to consider three cases. If $\beta = 1$, then

$$||h(F,L)||^2 - ||h(F,L) + h_L(F,L)\Delta L^n||^2 \geq ||h(F,L)|| \min(||h(F,L)||, \omega\delta, \omega\sigma).$$

If $\beta = \frac{\omega\delta}{||\Delta \hat{L}^n||}$ and using

$$\frac{1}{||\Delta \hat{L}^n||} \geq \frac{1}{||h_L^{-1}(F,L)||\, ||h(F,L))||} \geq \frac{1}{\nu_1 ||h(F,L))||},$$

we obtain

$$\begin{aligned}
||h(F,L)||^2 - ||h(F,L) + h_L(F,L)\Delta L^n||^2 &\geq \frac{\omega\delta}{\nu_1} ||h(F,L)|| \\
&\geq \frac{||h(F,L)||}{\nu_1} \min(||h(F,L)||, \omega\delta, \omega\sigma).
\end{aligned}$$

Otherwise, $\beta = -\frac{\sigma\omega}{\lambda_{\min}(E^{-T}\Delta \hat{L}^n E^{-1})}$. Using $\lambda_{\min}(E^{-T}\Delta \hat{L}^n E^{-1}) < 0$, the boundedness of E, and the relation

$$|\lambda_{\min}| \leq \rho(A) := \max_i |\lambda_i| \leq ||A||,$$

where $A \in \mathbb{R}^{n\times n}$ denotes an arbitrary matrix and λ_i the corresponding eigenvalues of A, we get

$$\begin{aligned}
||h(F,L)||^2 - ||h(F,L) + h_L(F,L)\Delta L^n||^2 &\geq \frac{\sigma\omega}{|\lambda_{\min}(E^{-T}\Delta \hat{L}^n E^{-1})|} ||h(F,L)||^2 \\
&\geq \frac{\sigma\omega}{||E^{-T}||\, ||E^{-1}||\, ||\Delta \hat{L}^n||} ||h(F,L)||^2 \\
&\geq \frac{\sigma\omega}{||E^{-1}||^2\, ||h_L^{-1}(F,L)||\, ||h(F,L)||} ||h(F,L)||^2 \\
&\geq \frac{||h(F,L)||}{\nu_1\nu_2^2} \min(||h(F,L)||, \omega\delta, \omega\sigma).
\end{aligned}$$

Hence, the result holds for $\kappa = \min(1, \frac{1}{\nu_1}, \frac{1}{\nu_1\nu_2^2})$. ∎

Condition (2.12) tells us that the quasi-normal component $S^n = (\Delta L^n, 0) = (\beta\Delta\hat{L}^n, 0)$ is indeed small close to feasible points. Condition (2.13) is just a weaker form of Cauchy decrease for the quasi-normal subproblem.

2.2. Approximate solution of the tangential problem. In this section we show how to derive a conjugate gradient (CG) algorithm to compute the tangential component of the step. Assuming that the quasi-normal component $S^n = (\Delta L^n, 0)$ is computed by the procedure stated in the previous section, the second trust region subproblem for determining the tangential component is defined by minimizing $q(S^n + S^t)$ subject to a trust region constraint and the SDP-conditions defined in (2.5).

By using Lemma 2.1 and the representation (1.7) of the step, and observing that ℓ^μ, $\ell^\mu_L\Delta L^n$, $\ell^\mu_{LL}(\Delta L^n, \Delta L^n)$ are constant terms not depending on ΔF, a simple calculation shows that $q(S^n + S^t)$ can be restated as a function ψ depending only on ΔF, i.e.,

$$\begin{aligned}\psi(\Delta F) &= \langle \Delta F, \nabla\Phi^\mu_F\rangle + \langle \Delta F, T_1^*(F,L)\nabla\Phi^\mu_L\rangle + \langle \Delta F, \nabla^2\ell^\mu_{FL}\Delta L^n\rangle \\ &\quad + \langle \Delta F, T_1^*(F,L)\nabla^2\ell^\mu_{LL}\Delta L^n\rangle + \frac{1}{2}\langle \Delta F, \nabla^2\ell^\mu_{FF}\Delta F\rangle \\ &\quad + \langle \Delta F, \nabla^2\ell^\mu_{FL}T_1(F,L)\Delta F)\rangle + \frac{1}{2}\langle \Delta F, T_1^*(F,L)\nabla^2\ell^\mu_{LL}T_1(F,L)\Delta F)\rangle, \end{aligned} \tag{2.14}$$

where $\Phi^\mu = \Phi^\mu(F,L)$ and $T_1(F,L) = -h_L^{-1}(F,L)h_F(F,L) \in \mathcal{L}(\mathbb{R}^{p\times r}, \mathbb{R}^{n\times n})$ denotes the first component mapping of the operator T defined by (1.7). Then we approximately solve the trust region subproblem

$$\begin{aligned} &\min_{\Delta F}\ \psi(\Delta F) \\ \text{s.t.}\quad &Y(F+\Delta F, L + T_1(F,L)\Delta F + \Delta L^n) \preceq (1-\sigma)Y(F,L), \\ &T_1(F,L)\Delta F + \Delta L^n \succeq -\sigma L, \quad ||\Delta F|| \le \delta \end{aligned} \tag{2.15}$$

for computing the F-part of the tangential component. Thereafter, having found an approximate solution of (2.15), the L-part of S^t can be obtained by solving (2.7) for ΔL^t, i.e., $\Delta L^t = T_1(F,L)\Delta F$. In [11] such an approach is referred to as the decoupled approach, because the trust region constraint is of the form $||\Delta F|| \le \delta$. Note that the minimization has to be taken over the null space $\mathcal{N}(h'(F,L))$. Interpreting ΔL^t as a function of ΔF means that $\mathcal{N}(h'(F,L))$ is projected onto the set $\{\Delta F \in \mathbb{R}^{p\times r}\}$. The decoupled approach has the advantage that the minimization problem (2.15) is defined in the space of the F-variables, which, in general, has a much smaller dimension than the space of the L-variables.

Now we apply a modification of the CG algorithm proposed by Steihaug [33] for finding an approximate solution of the tangential problem (2.15), where $\psi(\Delta F)$ is given by (2.14). In order to state the modified CG approach, we ignore for a moment the trust region and the SDP-constraints in (2.15) and derive the Newton step for minimizing $\psi(\Delta F)$ in the following result.

LEMMA 2.4. *Let $\mu > 0$, $(F,L) \in \mathcal{F}_s$, and $K = (h_L^{-1})^*\nabla\Phi^\mu_L$ be given. Assume that $\Delta F \in \mathbb{R}^{p\times r}$ is a solution to the problem $\min\psi(\Delta F)$, where $\psi(\Delta F)$ is defined by* (2.14); *then ΔF satisfies the Newton equation*

$$\begin{aligned}\mathcal{U}^t(\Delta F) &= \nabla^2\ell^\mu_{FL}T_1(F,L)\Delta F + \nabla^2\ell^\mu_{FF}\Delta F + h_F^*\Delta K^t(\Delta F) \\ &= -\left(\nabla\Phi^\mu_F + \nabla^2\ell^\mu_{FL}\Delta L^n + h_F^*(K + \Delta K^n)\right), \end{aligned} \tag{2.16}$$

where ΔL^n denotes the quasi-normal component as defined in the previous paragraph, $T_1(F,L) = -h_L^{-1}(F,L)h_F(F,L) \in \mathcal{L}(\mathbb{R}^{p\times r}, \mathbb{R}^{n\times n})$, and ΔK^n, $\Delta K^t(\Delta F)$ are solutions of the following equations:

$$h_L^* \Delta K^n + \nabla^2 \ell^\mu_{LL} \Delta L^n = 0, \tag{2.17}$$

$$h_L^* \Delta K^t(\Delta F) + (\nabla^2 \ell^\mu_{FL})^* \Delta F + \nabla^2 \ell^\mu_{LL} T_1(F,L)\Delta F = 0, \tag{2.18}$$

where $h_L^*, h_F^*, (\nabla^2 \ell^\mu_{FL})^*$ denote the adjoint operators of $h_L, h_F, \nabla^2 \ell^\mu_{FL}$, respectively.

Proof. Using properties of the defined inner product, the Newton equation (2.16) follows by a direct differentiation of $\psi_c(\Delta F)$ with respect to ΔF. ∎

Note that by defining $\Delta K = \Delta K^n + \Delta K^t(\Delta F)$, the solutions of (2.17) and (2.18) can be interpreted as the quasi-normal and the tangential component, respectively, of the step ΔK for the multiplier K, which itself is defined by the adjoint equation (2.2).

A practical way for approximately solving the tangential subproblem (2.15) is to apply a modification of the CG algorithm which is tailored to the special structure of this nonlinear SDP-problem. The main difference of this modified CG version is the following. Assuming that $\mu > 0$, $\sigma \in (0,1)$, $(F,L) \in \mathcal{F}_s$, and K and ΔL^n are given, then during each CG iteration we compute a positive scalar τ, which makes sure that, on exit, the step ΔF stays inside the current trust region and $F + \Delta F$, $T_1(F,L)\Delta F + \Delta L^n$ satisfy the SDP-constraints of (2.15). Thus, we know that an approximate solution of (2.15) yields simultaneously a direction ΔF which lies inside of the current trust region and $(F + \Delta F, L + T_1(F,L)\Delta F + \Delta L^n) \in \mathcal{F}_s$. This leads to the following algorithm for computing an approximation of ΔF.

ALGORITHM 2.1 (CG method for solving the tangential subproblem).

0. *Set $V_0 = 0$ and solve (2.17) for ΔK^n. Compute*

$$U_0 = -\left(\nabla \Phi^\mu_F + \nabla^2 \ell^\mu_{FL} \Delta L^n + h_F^*(K + \Delta K^n)\right). \tag{2.19}$$

Choose $D_0 = U_0$, and $\epsilon > 0$.

1. *For $i = 0, 1, 2, \ldots$ do*
   - 1.1 *Solve (2.18) for $\Delta K^t(D_i)$ and compute $a_i = \frac{||U_i||^2}{\langle D_i, \mathcal{U}^t(D_i)\rangle}$.*
   - 1.2 *Compute*

$$\begin{aligned}\tau_i = \max\{\tau > 0 \mid\ & ||V_i + \tau D_i|| \le \delta, \Delta L^n + T_1(F,L)(V_i + \tau D_i) \succeq -\sigma L, \\ & Y(F + V_i + \tau D_i, L + T_1(F,L)(V_i + \tau D_i) + \Delta L^n) \\ & \preceq (1-\sigma) Y(F,L)\}.\end{aligned}$$

   - 1.3 *If $a_i \le 0$ or $a_i > \tau_i$, then set $\Delta F = V_i + \tau_i D_i$ and stop; otherwise, set $V_{i+1} = V_i + a_i D_i$.*
   - 1.4 *Set $\eta = \min\{\epsilon, ||U_0||\}$, and update the residual: $U_{i+1} = U_i - a_i \mathcal{U}(D_i)$.*
   - 1.5 *Check truncation criteria: If $\frac{||U_{i+1}||}{||U_0||} \le \eta$, set $\Delta F = V_{i+1}$ and stop.*
   - 1.6 *Compute $b_i = \frac{||U_{i+1}||^2}{||U_i||^2}$ and set $D_{i+1} = U_{i+1} + b_i D_i$.*

2. *Solve (2.7) for ΔL^t.*

There are different ways in which the modified CG method can terminate.

(1) A direction of negative curvature is encountered in the CG iteration. In this case, we follow this direction until reaching the boundary of the intersection of the trust region and the SDP-constraints. Then the resulting step is returned as an approximate solution of tangential subproblem (2.15).

(2) The CG iterate has stepped outside of the intersection of the trust region and the SDP-constraints. In this case, we backtrack to this region and return the resulting step as an approximate solution of (2.15).

(3) The algorithm terminates with the inexact criterion 1.6 of Algorithm 2.1.

Note that, on exit, Algorithm 2.1 returns a step ΔF, and it is ensured that this step is an approximate solution of the tangential subproblem (2.15). Thereafter, for given ΔF, we obtain the L-part of the tangential component S^t by solving (2.7) for ΔL^t, i.e., $\Delta L^t = T_1(F, L)\Delta F$. By the construction of the CG algorithm, it is assured that, on exit, the pair $(F + \Delta F, \Delta L^t + \Delta L^n)$ satisfies the SDP-constraints of the tangential problem (2.15). Furthermore, if the CG algorithm terminates in step 1.6, then ΔF can be interpreted as an inexact Newton step which lies inside the trust region and satisfies the nonlinear SDP-constraints. Finally, in the implementation of Algorithm 2.1, we also include a preconditioner for speeding up the CG method (see, i.e., [22, page 121, equation (3.4.146)]).

Since the conjugate gradient Algorithm 2.1 starts by minimizing the quadratic function $\psi(\Delta F)$ along the direction U_0 defined by (2.19), it is quite clear that it produces a reduced tangential component that satisfies the fraction of Cauchy decrease condition

$$\psi(0) - \psi(\Delta F) \geq \theta^d(\psi(0) - \psi(c^d)) \tag{2.20}$$

with $\theta^d = 1$, where c^d denotes the Cauchy step (compare with [11, section 5.2.3]). Note that the Cauchy step is defined as the solution of the problem

$$\begin{aligned} &\min_{\Delta F} \ \psi(\Delta F) \\ \text{s.t.} \quad &Y(F + \Delta F, L + T_1(F, L)\Delta F + \Delta L^n) \preceq (1 - \sigma)Y(F, L), \\ &T_1(F, L)\Delta F + \Delta L^n \succeq -\sigma L, \quad ||\Delta F|| \leq \delta, \quad \Delta F \in \text{ span}\{U_0\}, \end{aligned} \tag{2.21}$$

where U_0 defined by (2.19) is the steepest-descent direction for the function $\psi(\Delta F)$ at $\Delta F = 0$ (see Lemma 2.4). Here, the fixed parameter $\sigma \in (0, 1)$ guarantees that the Cauchy step c^d remains strictly feasible with respect to the SDP-constraints. Thus, as in many trust region algorithms, the tangential component gives a decrease on $\psi(\Delta F)$ which is smaller than a uniform fraction of the Cauchy decrease given by c^d for the same function $\psi(\Delta F)$. The fulfillment of the Cauchy decrease condition (2.20) for the tangential component is important for assuring the global convergence of the trust region method to a first order KKT point of the barrier problem.

2.3. Detailed description of the trust region algorithm. In the previous sections we have specified how the normal and the tangential subproblems are to be solved. Now we can give a precise description of the CTR algorithm for solving the barrier problem (1.3) for a fixed barrier parameter $\mu > 0$. We need to introduce a merit function and the corresponding actual and predicted reduction for measuring the improvement of the algorithm. As a merit function we use the augmented Lagrangian function

$$\Lambda_\rho(F, L, K) = \ell^\mu(F, L, K) + \rho||h(F, L)||^2,$$

where ρ denotes a positive penalty parameter and $\ell^\mu(\cdot)$ denotes the Lagrangian function associated with barrier problem (1.3).

In our derivation of the CTR method we closely follow [10], [11], and [28]. At iteration k the actual decrease in the merit function is given by

$$ared_{\rho_k}(S_k) = \Lambda_{\rho_k}(F_k, L_k, K_k) - \Lambda_{\rho_k}(F_k + \Delta F_k, L_k + \Delta L_k, K_{k+1}),$$

where $\Delta L_k = \Delta L_k^n + \Delta L_k^t$ and $S_k = S^n + S^t$ denotes the trial step which is computed by the procedures described in the subsections above. The predicted decrease is defined by

$$\begin{aligned} pred_{\rho_k}(S_k) = q_k(0) - q_k(S_k) - \langle \Delta K_k, h_L^k \Delta L_k + h_F^k \Delta F_k + h^k \rangle \\ + \rho_k(||h^k||^2 - ||h_L^k \Delta L_k + h_F^k \Delta F_k + h^k||^2), \end{aligned}$$

where $\Delta K_k = K_{k+1} - K_k$ and $h^k = h(F_k, L_k)$.

From the computation of the quasi-normal component ΔL_k^n, we know that $\Delta L_k^n = \beta_k \Delta \hat{L}_k^n$, $\beta_k \in (0, 1]$, and $\Delta \hat{L}_k^n$ is a solution of (2.6), which implies $h_L^k \Delta L_k^n = -\beta_k h^k$. Moreover, the L-part of S^t is given by $\Delta L_k^t = T_1(F_k, L_k)\Delta F^k$. Thus, $h_L^k \Delta L_k^t + h_F^k \Delta F_k = 0$. Combining the last two facts and using $h_L^k \Delta L = h_L^k \Delta L_k^n + h_L^k \Delta L_k^t$, we obtain $h_L^k \Delta L_k + h_F^k \Delta F_k + h^k = (1 - \beta_k) h^k$. Therefore, the predicted decrease can be restated as

$$pred_{\rho_k}(S_k) = q_k(0) - q_k(S_k) - (1 - \beta_k)\langle \Delta K_k, h^k \rangle + \rho_k \beta_k ||h^k||^2. \tag{2.22}$$

To decide whether to accept or reject a trial step S_k, we evaluate the ratio $r_k = ared_{\rho_k}(S_k)/pred_{\rho_k}(S_k)$. If the ratio is too small, we reject the trial step and decrease the trust region δ_k. On the other hand, if the trial step is accepted, we increase the trust region. To update the penalty parameter ρ_k we use the scheme proposed by El-Alem [12]. For estimating the Lagrange multiplier K_k, we solve the adjoint equation

$$h_L^*(F_k, L_k) K_k + \nabla J_L(F_k, L_k) - \mu M(F_k, L_k) = 0, \tag{2.23}$$

where $M(F, L)$ is defined by (1.11), $(F_k, L_k) \in \mathcal{F}_s$, and $\mu > 0$ is fixed.

A reasonable termination criterion for the CTR algorithm is

$$||\nabla \ell_F^{\mu_j}(F_k, L_k, K_k)|| + \|h(F_k, L_k)\| \le \epsilon, \tag{2.24}$$

where $\epsilon > 0$ is a prespecified tolerance and $\nabla \ell_F^{\mu}(F, L, K)$ is defined by (1.9).

We can now outline the main procedures of the CTR algorithm for solving the barrier problem.

ALGORITHM 2.2 (CTR algorithm for the barrier problem). *Let* $0 < a_1 \le a_2 < 1$, $0 < \gamma_1 < \gamma_2 < \gamma_3 < 1$, $\bar{\rho} > 0$, $\rho_{-1} \ge 1$, $\mu > 0$, *and* $\epsilon > 0$ *be given. Choose* $(F_0, L_0) \in F_s$. *Calculate* K_0 *by* (2.23), *and pick* δ_0 *such that* $0 < \delta_{min} \le \delta_0 \le \delta_{max}$. *For* $k = 0, 1, 2, \ldots$ *do*

1. *If* $||\nabla \ell_F^{\mu_j}(F_k, L_k, K_k)|| + ||h^k|| \le \epsilon$, *stop.*
2. *Compute* ΔL_k^n *and* β_k *as stated in subsection* 2.1.
   Solve the tangential problem (2.15) *for obtaining* ΔF_k *as stated in subsection* 2.2.
   Determine ΔL_k^t *by solving* (2.7).
   Set $\Delta L_k = \Delta L_k^n + \Delta L_k^t$ *and define* $S_k = (\Delta L_k, \Delta F_k)$.
3. *Compute* K_{k+1} *by* (2.23) *with* $F_k + \Delta F_k$ *and set* $\Delta K_k = K_{k+1} - K_k$.
4. *Compute* $pred_{\rho_{k-1}}(S_k)$ *by using* (2.22), *and update the penalty parameter as follows:*
   If $pred_{\rho_{k-1}}(S_k) \ge \frac{\rho_{k-1}\beta_k}{2}||h^k||^2$, *then set* $\rho_k = \rho_{k-1}$. *Otherwise set*

$$\rho_k = \frac{2(q_k(S_k) - q_k(0) + (1 - \beta_k)\langle \Delta K_k, h^k \rangle)}{\beta_k ||h^k||^2} + \bar{\rho}.$$

5. *Compute the ratio* $r_k = \frac{ared_{\rho_k}(S_k)}{pred_{\rho_k}(S_k)}$.
6. *Update the trust region and accept or reject the step by the following:*
   (a) *If* $r_k < \gamma_1$, *set* $\delta_{k+1} = a_1 \max\{||\Delta L_k^n||, ||\Delta F_k||\}$, *or*
   if $\gamma_1 \le r_k < \gamma_2$, *set* $\delta_{k+1} = a_2 \max\{||\Delta L_k^n||, ||\Delta F_k||\}$, *and*
   reject the step. Set $F_{k+1} = F_k$, $L_{k+1} = L_k$, $K_{k+1} = K_k$.
   (b) *If* $\gamma_2 \le r_k < \gamma_3$, *set* $\delta_{k+1} = \delta_k$, *or*
   if $r_k \ge \gamma_3$, *set* $\delta_{k+1} = \min\{\max\{2\delta_k, \delta_{min}\}, \delta_{max}\}$, *and*
   accept the step. Set $F_{k+1} = F_k + \Delta F_k$, $L_{k+1} = L_k + \Delta L_k$,
   $K_{k+1} = K_k + \Delta K_k$.

Note that the sequence $\{(F_k, L_k)\}$ generated by the CTR Algorithm 2.2 is always contained in the set $\mathcal{F}_s$ (see Lemma 2.5 below).

The global as well as local convergence behavior of Algorithm 2.2 can been shown by using the same general assumptions as in Dennis, El-Alem, and Maciel [10] or in Dennis, Heinkenschloss, and Vicente [11]. Since the whole convergence analysis is beyond the scope of this paper, in the next section we state only the main global convergence result.

2.4. Convergence behavior of the CTR Algorithm 2.2. The proof of the global convergence of the CTR Algorithm 2.2 for solving the barrier problem (1.3) to a first order KKT point can be established similarly to the convergence theory presented in [10] for general equality constrained optimization problems. Therefore, we omit the proof and refer the interested reader to [11, section 7] or [10, section 7] for the details of the whole convergence theory of CTR methods.

We start by stating some general assumptions under which global convergence can be proved for Algorithm 2.2. These assumptions are used by several authors; for example, see [6], [11], [10], [12], [29], and the references therein. Note that the set $\mathcal{F}_s$, which is the set of all pairs (F, L) satisfying the nonlinear SDP-constraints, is an open subset of $\mathbb{R}^{p\times r} \times \mathbb{R}^{n\times n}$. Let $\Omega \subseteq \mathcal{F}_s$ be such that for all iterations k of Algorithm 2.2, (F_k, L_k) and $(F_k + \Delta F_k, L_k + \Delta L_k)$ are in Ω. By Lemma 2.1, we know that all problem functions are twice continuously differentiable on $\mathcal{F}_s$. For applying the convergence theory of [11] or [10] to our algorithm, we need the following assumptions.

ASSUMPTION 2.1. *Let Assumptions* 1.1(i)–(iii) *be fulfilled and* $\mu > 0$ *be given. Assume that the functions* $J(F, L)$, $\Phi^\mu(F, L)$, $h(F, L)$, $Y(F, L)$, *and their first and second order derivatives are bounded in* $\Omega \subseteq \mathcal{F}_s$. *Moreover, suppose that the sequences* $\{T(F_k, L_k)\}$ *and* $\{\mathcal{H}(F_k, L_k)\}$ *are bounded in* $\Omega \subseteq \mathcal{F}_s$, *where* $\{\mathcal{H}(F_k, L_k)\}$ *denotes the Hessian of the Lagrangian of the barrier problem defined by* (1.5). *Furthermore, let* $Y^{-1}(F, L)$ *and the operator* $h_L^{-1}(F, L)$ *be uniformly bounded in* $\mathcal{F}_s$. *Finally, the sequences* $\{F_k\}$, $\{L_k^{-1}\}$, $\{K_k\}$ *are bounded.*

Before stating the global convergence result of Algorithm 2.2, we show the following result.

LEMMA 2.5. *Let Assumption* 2.1 *be satisfied; then the sequence of iterates* $\{(F_k, L_k)\}$ *generated by the CTR Algorithm* 2.2 *satisfies* $(F_k, L_k) \in \mathcal{F}_s$ *for all* $k \ge 0$.

Proof. Let (F_0, L_0) be chosen in $\mathcal{F}_s$. Without loss of generality, we may assume for $k \ge 0$ that the pair $(F_k, L_k) \in \mathcal{F}_s$ lies in $\mathcal{F}_s$. Using the results of section 2.1, at iteration $k \ge 0$ the quasi-normal component satisfies $\Delta L_k^n = \beta_k \Delta \hat{L}_k^n \succeq -\sigma\omega L_k$, where $\sigma, \omega \in (0, 1)$ are given scalars and $\beta_k \in (0, 1]$ is defined by (2.11). Moreover, using the discussion in section 2.2, at iteration $k \ge 0$ the approximate solution of the tangential problem (2.15) fulfills the nonlinear SDP-conditions

$$\begin{aligned} Y(F_k + \Delta F_k, L_k + T_1(F_k, L_k)\Delta F_k + \Delta L_k^n) &\preceq (1 - \sigma) Y(F_k, L_k), \\ T_1(F_k, L_k)\Delta F_k + \Delta L_k^n &\succeq -\sigma L_k, \end{aligned} \tag{2.25}$$

where ΔF_k is computed by the CG Algorithm 2.1. From the rules that update δ_k in step 6 of Algorithm 2.2, [10, Theorem 8.1] tells us that an acceptable step is always found after a finite number of unsuccessful iterations. Using this fact, we can ignore the rejected steps and work only with successful iterates. Hence, a successful iterate $(F_{k+1}, L_{k+1}) = (F_k + \Delta F_k, L_k + \Delta L_k^t + \Delta L_k^n)$ satisfies $Y(F_{k+1}, L_{k+1}) \preceq (1 - \sigma)Y(F_k, L_k) \prec 0$ and $L_{k+1} \succeq (1-\sigma)L_k \succ 0$, where $\Delta L_k^t = T_1(F_k, L_k)\Delta F_k$. Thus, $(F_k, L_k) \in \mathcal{F}_s$ for all $k \geq 0$. ∎

This lemma guarantees the feasibility of the generated sequence with respect to the nonlinear SDP-conditions. It assures that Algorithm 2.2 produces an approximate solution of the barrier problem which always satisfies these conditions.

In the previous sections, we have described in detail how we compute the approximate solutions of the quasi-normal and the tangential subproblems. Particularly, in Lemma 2.3 we have shown that the quasi-normal component S^n, as computed in subsection 2.1, is small close to feasible points and that it fulfills a certain form of the Cauchy decrease condition; i.e., S^n satisfies (2.12) and (2.13). Moreover, in subsection 2.2, the tangential component is computed by the CG Algorithm 2.1. Thus, the tangential component also satisfies a Cauchy decrease condition, i.e., the fraction of Cauchy decrease (2.20). The fulfillment of these conditions together with Assumption 2.1 is enough for showing the global convergence of Algorithm 2.2. In particular, making straightforward modifications to [11, Theorem 7.5 and Corollary 7.6], the following convergence result can be proved.

THEOREM 2.6. *Let Assumption* 2.1 *be satisfied. Then the sequence of iterates* $\{(F_k, L_k)\}$ *generated by the CTR Algorithm* 2.2 *for solving the barrier problem* (1.3) *satisfies*

$$\liminf_{k\to\infty} \left(||\nabla \ell_F^{\mu_j}(F_k, L_k, K_k)|| + ||h(F_k, L_k)|| \right) = 0,$$

where $\nabla \ell_F^{\mu}(F, L, K)$ *is given by* (1.9) *and* K_k *denotes the solution of the adjoint equation* (1.10).

Moreover, if $\{(F_k, L_k)\}$ *is a bounded sequence, then* $\{(F_k, L_k)\}$ *has a limit point satisfying the first order necessary optimality conditions of the barrier problem* (1.3), *i.e., conditions* (2.1)–(2.3) *and* (1.2).

In the next section, we will see that this theorem guarantees the finite termination property of the interior point algorithm; for example, we can terminate the CTR method as soon as a current iterate satisfies $||\nabla \ell_F^{\mu}(F, L, K)|| + ||h(F, L)|| \leq \epsilon$, where $\epsilon > 0$ is a prespecified tolerance.

3. IPCTR algorithm. In this section we consider the overall algorithm, the IPCTR method, in which Algorithm 2.2 is executed for decreasing values of the barrier parameter μ. We are not concerned here with conditions assuring a good local rate of convergence, but consider only the global convergence properties of the IPCTR algorithm. The study of the local convergence behavior is a part of our current research and will be considered in a forthcoming paper.

ALGORITHM 3.1 (IPCTR Algorithm). *Choose* $\mu_0 > 0$, $\epsilon_0 > 0$, $a, b \in (0, 1)$, *and* $(F_0, L_0) \in \mathcal{F}_s$. *Set* $j = 0$.

1. *For fixed* $\mu_j > 0$ *apply Algorithm* 2.2 *from* $(F_j, L_j) \in \mathcal{F}_s$ *until it finds a point* (F_{j+1}, L_{j+1}) *satisfying*

$$||\nabla \ell_F^{\mu_j}(F_{j+1}, L_{j+1}, K_{j+1})|| + ||h(F_{j+1}, L_{j+1})|| \leq \epsilon_j, \tag{3.1}$$

where $\nabla \ell_F^{\mu}(F, L, K)$ *is defined by* (1.9) *and* K_{j+1} *denotes the solution of the adjoint equation* (1.10) *evaluated at* (F_{j+1}, L_{j+1}).

2. *Choose* $\mu_{j+1} \in (0, a\mu_j)$ *and* $\epsilon_{j+1} \in (0, b\epsilon_j)$.
3. *Increase* j *by one, and go to step* 1.

To establish the global convergence result of Algorithm 3.1 we need the same assumptions as for the convergence of the inner loop to a first order KKT point of the barrier problem. In particular, we have the following theorem.

THEOREM 3.1. *Suppose that* $\{(F_j, L_j)\}$ *is generated by Algorithm* 3.1 *and that, for each barrier problem, Assumption* 2.1 *holds. Furthermore, let* $\{(F_j, L_j)\}$ *be bounded. Then, at each outer iteration, the inner algorithm succeeds in finding a pair* (F_{j+1}, L_{j+1}) *satisfying* (3.1). *If* $\mu_j \to 0$ *and* $\epsilon_j \to 0$, *then any accumulation point* $(\hat{F}, \hat{L})$ *of* $\{(F_j, L_j)\}$ *satisfies the first order necessary optimality condition of the nonlinear SDP-problem* (1.1); *i.e., there exists* $\hat{K} \in \mathbb{R}^{n\times n}$ *such that* $\hat{L} \succ 0$, $Y(\hat{F}, \hat{L}) \prec 0$, *and*

$$\nabla J_F(\hat{F}, \hat{L}) + h_F^*(\hat{F}, \hat{L})\hat{K} = 0, \quad h(\hat{F}, \hat{L}) = 0, \quad h_L^*(\hat{F}, \hat{L})\hat{K} + \nabla J_L(\hat{F}, \hat{L}) = 0.$$

Proof. Suppose that for some value of μ_j the inner Algorithm 2.2 fails to find a point satisfying (3.1). This implies that Algorithm 2.2 generates an infinite sequence $\{(F_k, L_k)\}$ for the barrier problem (1.3) with $\mu = \mu_j$ such that

$$||\nabla J_F(F_k, L_k) - \mu Y_F^*(F_k, L_k)Y^{-1}(F_k, L_k) + h_F^*(F_k, L_k)K_k|| + ||h(F_k, L_k)|| \not\to 0,$$

which contradicts the result of Theorem 2.6. Thus, the inner loop succeeds in finding (F_{j+1}, L_{j+1}) satisfying (3.1) for $j \geq 0$.

Let $\mathcal{J}$ be a subsequence of indices j such that (F_j, L_j) converges to $(\hat{F}, \hat{L})$ when $j \to \infty$ in $\mathcal{J}$. Since, by construction of Algorithm 2.2, $0 \leq \epsilon_{j+1} < \epsilon_j$ and $0 \leq ||\nabla \ell_F^{\mu_j}(F_{j+1}, L_{j+1}, K_{j+1})|| + ||h(F_{j+1}, L_{j+1})|| \leq \epsilon_j$ for all $j \geq 0$, the sequence $\{||\nabla \ell_F^{\mu_j}(F_{j+1}, L_{j+1}, K_{j+1})|| + ||h(F_{j+1}, L_{j+1})||\}$ is a monotonically decreasing sequence which is bounded below by zero. Thus, using $\epsilon_j \to 0$, it converges to zero and this implies

$$||\nabla \ell_F^{\mu_{j-1}}(F_j, L_j, K_j)|| \to 0, \qquad ||h(F_j, L_j)|| \to 0$$

if $j \to \infty$ in $\mathcal{J}$. Using this relation and Lemma 2.5, we deduce that $h(F_j, L_j) \to 0$ and $(F_j, L_j) \in \mathcal{F}_s$ for all $j \geq 0$ and $\mu_j > 0$. Therefore, we conclude that $(\hat{F}, \hat{L}) \in \mathcal{F}_s$ and $h(F_j, L_j) \to h(\hat{F}, \hat{L}) = 0$ if $j \to \infty$ in $\mathcal{J}$. Hence, $(\hat{F}, \hat{L})$ is feasible for (1.1).

Moreover, the sequence $\{K_j\}$ defined as the solution of the adjoint equation (2.23) is uniformly bounded in $\mathcal{F}_s$. Now, using Assumption 2.1, the boundedness of $Y_L^*(F_j, L_j)$, $\{L_j^{-1}\}$, and $Y^{-1}(F, L)$ in $\mathcal{F}_s$ implies the existence of positive constants $\theta_i, i = 1, 2, 3$, independent of j such that

$$||Y_L^*(F_j, L_j)|| \leq \theta_1, \quad ||L_j^{-1}|| \leq \theta_2, \quad ||Y^{-1}(F_j, L_j)|| \leq \theta_3$$

for all $j \geq 0$. Therefore, since K_j denotes the exact (unique) solution of the adjoint equation (2.23), we obtain

$$\begin{aligned}
&||h_L^*(F_j, L_j)K_j + \nabla J_L(F_j, L_j)|| \\
&\leq ||h_L^*(F_j, L_j)K_j + \nabla J_L(F_j, L_j) - \mu_{j-1}M(F_j, L_j)|| + \mu_{j-1}||M(F_j, L_j)|| \\
&\leq \mu_{j-1}\left(||L_j^{-1}|| + ||Y_L^*(F_j, L_j)||\,||Y^{-1}(F_j, L_j)||\right) \;\leq\; \mu_{j-1}\,(\theta_2 + \theta_1\,\theta_3)\,.
\end{aligned}$$

Hence, using $\mu_j \to 0$, we know that $||h_L^*(F_j, L_j)K_j + \nabla J_L(F_j, L_j)|| \to 0$ and

$$h_L^*(F_j, L_j)K_j + \nabla J_L(F_j, L_j) - \mu_{j-1}M(F_j, L_j) \to h_L^*(\hat{F}, \hat{L})\hat{K} + \nabla J_L(\hat{F}, \hat{L}) = 0, \tag{3.2}$$

whenever $j \to \infty$ in $\mathcal{J}$. Since $\{K_j\}$, defined as the solution of the adjoint equation (2.23), is uniformly bounded, and using (3.2), we know that it converges to the limit point $\hat{K}$.

Finally, the boundedness of $Y_F^*(F_j, L_j)$, $Y^{-1}(F, L)$, and (3.1) implies

$$\begin{aligned} ||\nabla J_F(F_j, L_j) + h_F^*(F_j, L_j)K_j|| &\leq \mu_{j-1}||Y_F^*(F_j, L_j)Y^{-1}(F_j, L_j)|| \\ &\quad + ||\nabla J_F(F_j, L_j) + h_F^*(F_j, L_j)K_j - \mu_{j-1}Y_F^*(F_j, L_j)Y^{-1}(F_j, L_j)|| \\ &\leq \epsilon_{j-1} + \mu_{j-1}\, ||Y_F^*(F_j, L_j)||\, ||Y^{-1}(F_j, L_j)|| \;\leq\; \epsilon_{j-1} + \mu_{j-1}\, \theta_4\, \theta_3, \end{aligned}$$

where $\theta_4 > 0$ denotes the positive constant such that $||Y_F^*(F_j, L_j)|| \leq \theta_4$. Now, since $\mu_j \to 0$ and $\epsilon_j \to 0$, we deduce that $||\nabla J_F(F_j, L_j) + h_F^*(F_j, L_j)K_j|| \to 0$ and

$$\nabla J_F(F_j, L_j)+h_F^*(F_j, L_j)K_j-\mu_{j-1}Y_F^*(F_j, L_j)Y^{-1}(F_j, L_j) \to \nabla J_F(\hat{F}, \hat{L})+h_F^*(\hat{F}, \hat{L})\hat{K} = 0$$

if $j \to \infty$ in $\mathcal{J}$. Therefore, any accumulation point $(\hat{F}, \hat{L})$ of $\{(F_j, L_j)\}$ satisfies the first order necessary optimality condition of (1.1). ∎

In a practical implementation of the IPCTR algorithm, we would like the step to satisfy the following properties near a solution of the nonlinear SDP-problem:

(i) it should provide a fast local rate of convergence;
(ii) it should at least satisfy approximately the KKT conditions of (1.1);
(iii) it should have problem depending updating rules for the barrier parameter sequence and the inner termination tolerances which enforce the fast local rates.

As shown in Theorem 3.1, the second condition is satisfied. We can guarantee that at least an accumulation point of the generated sequence satisfies the first order conditions of (1.1). Under stronger assumptions, it will be also possible to show that any accumulation point of this sequence is a local solution of the nonlinear SDP-problem (1.1). As noted above, the study of the local convergence behavior of the IPCTR method is a part of our current research which will be considered in a forthcoming paper. But the numerical results stated in the next section show that the IPCTR approach achieves fast local rates for sufficiently small barrier parameters in the vicinity of a solution of (1.1). Therein, we have chosen the parameter sequences $\{\mu_j\}$ and $\{\epsilon_j\}$ to be problem-dependent. In particular, instead of using a linear decrease in these parameter sequences as stated in Algorithm 3.1, we have chosen a more attractive updating rule for the barrier parameter and the inner termination tolerance. The barrier parameter is selected according to how much reduction one has made in the optimality conditions of the nonlinear SDP-problem, i.e.,

$$\mu_{j+1} = \min\left\{a\mu_j, ||H(F_{j+1}, L_{j+1}, K_{j+1})||^{1+\xi}, \frac{\langle L_{j+1}, Z\rangle^{1+\xi}}{n}\right\}, \tag{3.3}$$

and the updating rule for the inner termination criterion is chosen by

$$\epsilon_{j+1} = \min\left\{b\epsilon_j, ||H(F_{j+1}, L_{j+1}, K_{j+1})||^{1+\xi}\right\}, \tag{3.4}$$

where $0 < \xi \leq 1$, Z denotes the solution of the Lyapunov equation

$$L_{j+1}Z + ZL_{j+1} - 2\mu_j I = 0,$$

and $H(F, L, K)$ represents the KKT conditions of (1.1) defined by

$$H(F, L, K) = (h_F^*(F, L)K + \nabla J_F(F, L), h(F, L), h_L^*(F, L)K + \nabla J_L(F, L))^T.$$

With these updating rules, it will be possible to establish, under assumptions similar to those used in damped Newton methods, superlinear and quadratic convergence rates for the IPCTR algorithm near a solution of the nonlinear SDP-problem (1.1).

As indicated in Theorem 3.1, we can terminate the IPCTR Algorithm 3.1 as soon as an actual iterate (F_j, L_j) approximately satisfies the KKT conditions of (1.1). For example, choosing $\epsilon_{out} > 0$ sufficiently small with $\epsilon_{out} < \epsilon_0$, we terminate Algorithm 3.1 as soon as

$$\|H(F_j, L_j, K_j)\| \leq \epsilon_{out}. \tag{3.5}$$

Then, we know that the actual approximate solution of the barrier problem, with corresponding (in general small) barrier parameter μ, is an approximate guess of the solution of (1.1). Finally, in our numerical tests presented in the next section, we always set the inner termination criterion $\epsilon_{j+1} = \epsilon_{out}$ iff ϵ_{j+1} generated by (3.4) is less than the outer termination tolerance ϵ_{out} for some $j \geq 0$.

4. Numerical results. In this section, several examples are given for test purposes in order to test the IPCTR approach. We present examples borrowed from the control literature for designing an optimal (static or reduced order) output feedback control law. As noted in the introduction, these problems can be formulated as a nonlinear SDP-problem of the form (1.1). In particular, the goal is to solve (1.12) with our IPCTR method, which can be found in Algorithm 3.1.

The IPCTR algorithm was implemented using MATLAB 5.1 facilities. In particular, for solving the several (Lyapunov) equations during the procedures, we used the Control System Toolbox function LYAP. Moreover, since the algorithm initially requires a (strictly) feasible starting point (F_0, L_0) with respect to the nonlinear SDP-constraints (1.2), in most of the cases we determine such a starting point by the SLPMM proposed by Leibfritz [22], [23].

We compare the performance of the IPCTR algorithm with the CTR approach developed in [28] and with Newton's method combined with an Armijo step size rule as proposed by [34] and [31]. In the numerical examples, we denote the Newton algorithm by ARMIJO.

For IPCTR, the following data are given in the tables: the outer iteration counter j; the barrier parameter μ_j; the inner termination criterion ϵ_j; the inner iteration counter k; the inner termination measure $||(\nabla \ell_F^{\mu_j})_k|| + ||h^k||$, evaluated at each inner iterate, where $(\nabla \ell_F^{\mu_j})_k = \nabla \ell_F^{\mu_j}(F_k, L_k, K_k)$ is defined as in (3.1) and $h^k = h(F_k, L_k)$; the norm of the equality constraints $||h^k||$; the trust region radius δ_k; and finally, the accumulated number of CG iterations needed for determining an approximate solution of the tangential subproblem during each trust region iteration k. Furthermore, note that for each $j \geq 1$ the index $k = 0$ only indicates an update of the barrier parameter and an evaluation of all quantities at the actual outer iterate for the new barrier parameter μ; i.e., it contains only the initial data information for the next inner loop. Hence, we do not take this step into account if we count the overall number of inner iterations needed by the IPCTR algorithm.

We test our code with several different examples and various parameter selections. Since it is impossible to present all of them within the limitation of this paper, we restrict ourselves to the following representative examples.

Example 1 *(Chemical reactor models).* We consider two examples of a chemical reactor. The goal is to determine an optimal solution of the nonlinear SDP-problem (1.12). Note that this solution corresponds to an optimal stabilizing SOF control law. We compare the convergence behavior of the IPCTR method with the CTR and

ARMIJO approaches used in the past. The first chemical reactor example appeared in Appendix D of [17], and the data matrices are given by

$$A = \begin{bmatrix} 1.38 & -0.2077 & 6.715 & -5.676 \\ -0.5814 & -4.29 & 0 & 0.675 \\ 1.067 & 4.273 & -6.654 & 5.893 \\ 0.048 & 4.273 & 1.343 & -2.104 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & 0 \\ 5.679 & 0 \\ 1.136 & -3.146 \\ 1.136 & 0 \end{bmatrix}, \qquad C = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

$R = I_2$, and $P = Q = I_4$. The real parts of the eigenvalues of A are $(-8.66, 1.99, -5.06, 0.06)$. This implies that the SDP-conditions (1.2) for the choice $F = 0$ are not satisfied.

Thus, initially, we compute a pair (F_0, L_0) satisfying (1.2) by SLPMM, which yields

$$F_0 = \begin{bmatrix} 0.0923 & -1.3344 \\ 1.2776 & 0.1647 \end{bmatrix}, \quad L_0 = \begin{bmatrix} 1.9105 & 0.1459 & -1.2034 & 0.1808 \\ 0.1459 & 0.9652 & -0.0956 & 0.1363 \\ -1.2034 & -0.0956 & 2.3199 & 0.0511 \\ 0.1808 & 0.1363 & 0.0511 & 0.6457 \end{bmatrix}.$$

Choosing $\epsilon_{out} = 10^{-6}$ and considering Figure 4.1, we observe that IPCTR terminates after 4 outer and a total of 9 inner iterations, and CTR as well as ARMIJO also require 9 iterations for reaching the optimal solution

$$F_* = \begin{bmatrix} 0.3571 & -2.6242 \\ 2.5816 & 0.7764 \end{bmatrix}, \quad L_* = \begin{bmatrix} 0.5113 & 0.0205 & -0.1258 & 0.0629 \\ 0.0205 & 0.0275 & 0.0097 & 0.0166 \\ -0.1258 & 0.0097 & 0.5313 & 0.4636 \\ 0.0629 & 0.0166 & 0.4636 & 0.5421 \end{bmatrix}.$$

In IPCTR, we have taken the parameters $\mu_0 = 0.1$, $\epsilon_0 = 1$, $a = 0.001$, $b = 0.4$. After the 4th outer iteration, IPCTR terminates, since (F_3, L_3, K_3) satisfies (3.5). Moreover, Figure 4.1 also illustrates the global as well as the fast local behavior of all three methods. Thus, for this example, all three algorithms are competitive.

The convergence behavior for the second chemical reactor model can be found in Figure 4.2, where for this instance the data matrices are defined as follows: $R = I_2$, $P = Q = I_4$, and

$$A = \begin{bmatrix} 1.400 & -0.208 & 6.715 & -5.676 \\ -0.581 & -4.290 & 0 & 0.675 \\ 1.067 & 4.273 & -6.654 & 5.893 \\ 0.048 & 4.273 & 1.343 & -2.104 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & 0 \\ 5.679 & 0 \\ 1.136 & -3.146 \\ 1.136 & 0 \end{bmatrix}, \qquad C = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}.$$

Starting the algorithms with (F_0, L_0) computed by SLPMM and setting $\epsilon_{out} = 10^{-6}$, after 4 outer and a total of 8 inner IPCTR, 9 CTR, and 13 ARMIJO iterations, respectively, the methods converge to the same solution (F_*, L_*) giving $J_* = 3.503526$. Furthermore, Figure 4.2 illustrates the quadratic local convergence rates of IPCTR and CTR, while ARMIJO achieves only a linear rate of convergence for this example. Finally, we observe that IPCTR needs fewer iterations than CTR and ARMIJO. In the next example, we will see that the performance of the IPCTR algorithm can be much better than CTR and ARMIJO.

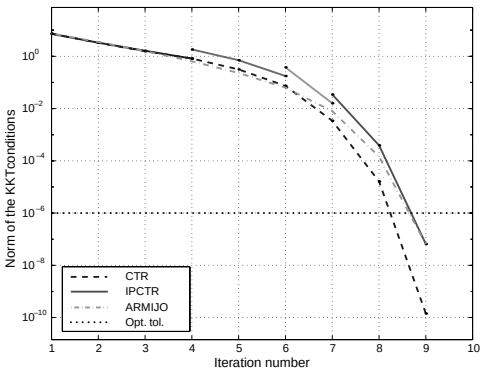


FIG. 4.1. *Convergence: first chemical reactor model.*

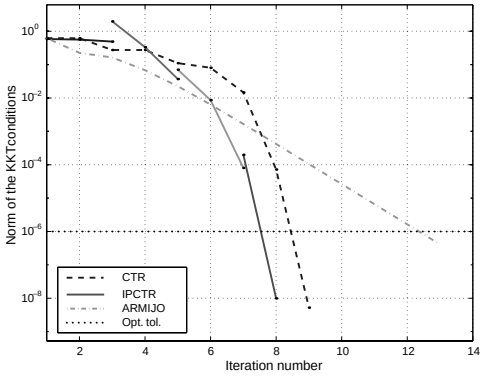


FIG. 4.2. *Convergence: second chemical reactor model.*

Example 2 *(Aircraft model).* This model appeared in Appendix F of [17] and describes a linearized model of the longitudinal equations of motions of an airplane. Again, the goal is to determine an optimal SOF control gain for the linearized dynamics of this model which is equivalent to an optimal solution of the nonlinear SDP-problem (1.12). The following are the data matrices for the resulting linearized state space model of the aircraft:

$$A = \begin{bmatrix} 0 & 0 & 1.132 & 0 & -1 \\ 0 & -0.0538 & -0.1712 & 0 & 0.0705 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0.0485 & 0 & -0.8556 & -1.013 \\ 0 & -0.2909 & 0 & 1.0532 & -0.6859 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 \\ -0.12 & 1 & 0 \\ 0 & 0 & 0 \\ 4.419 & 0 & -1.665 \\ 1.575 & 0 & -0.0732 \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \quad P = \begin{bmatrix} 0.0089 & 0 & 0 & 0.0047 & 0.0059 \\ 0 & 0.0015 & -0.0037 & 0 & 0 \\ 0 & -0.0037 & 0.0295 & 0 & 0 \\ 0.0047 & 0 & 0 & 0.0169 & 0.0190 \\ 0.0059 & 0 & 0 & 0.0190 & 0.0237 \end{bmatrix},$$

$R = 0.5I_3$, and $Q = \mathrm{diag}(0.001, 2.501, 2.501, 0.001, 0.001)$. Choosing P, Q, and R by the data above, we obtain the data set referred to as (AC1). On the other hand, (AC2) refers to the data set with $R = I_3$ and $P = Q = I_5$.

For the different data sets (AC1) and (AC2), the performance of the trust region algorithms are plotted in Figure 4.3 and Figure 4.4, respectively. These figures illustrate that for these instances IPCTR performs much better than CTR. In particular, for the data set (AC1), IPCTR needs only 8 outer and a total of 39 inner iterations, while CTR requires 70 iterations. Moreover, the ARMIJO algorithm reaches the solution after 115 iterations. For (AC1), we have used the result of SLPMM for initializing the algorithms, i.e., a run of SLPMM gives F_0 as

$$F_0 = \begin{bmatrix} 0.8190 & -0.0180 & 0.3200 \\ 0.0247 & -1.7164 & 0.0356 \\ 2.1739 & 0.0236 & 1.8139 \end{bmatrix},$$

and we have terminated the algorithms as soon as (3.5) holds with $\epsilon_{out} = 10^{-8}$. In IPCTR the following parameters have been used: $\mu_0 = 0.01$, $\epsilon_0 = 1$, $a = 0.001$, $b = 0.4$.

On the other hand, considering the data set (AC2), we have chosen $\mu_0 = 0.1$, $\epsilon_0 = 31$, $a = 0.01$, $b = 0.4$, and F_0 by

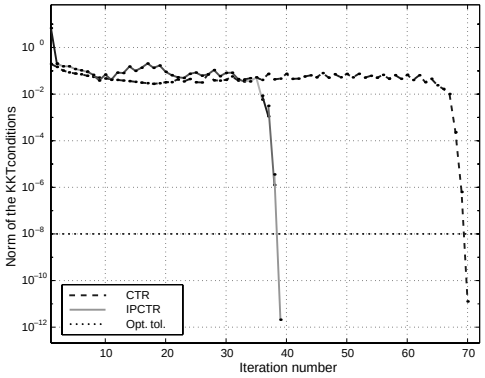


FIG. 4.3. *Convergence (AC1): IPCTR vs. CTR.*

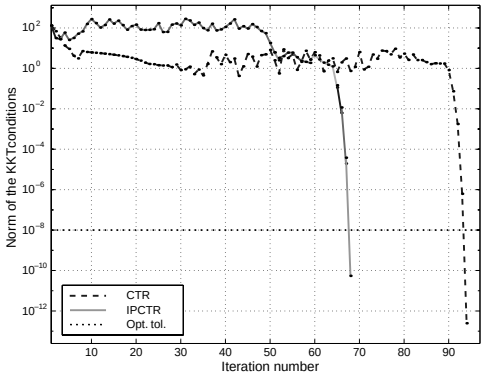


FIG. 4.4. *Convergence (AC2): IPCTR vs. CTR.*

$$F_0 = \begin{bmatrix} -3.04 & 205.19 & -1.9500 \\ 74.30 & -4808.6 & 73.20 \\ -7.00 & 463.00 & -3.60 \end{bmatrix},$$

together with an L_0 satisfying the nonlinear SDP-conditions (1.2). For this data set and this initial gain, we can draw the same observations as for (AC1). The IPCTR algorithm converges within 8 outer and a total of 68 inner iterations, while CTR needs 94 iterations. In contrast to this, the ARMIJO approach breaks down. This indicates the robustness of the trust region methods compared with an Armijo globalization strategy. Moreover, it underlines that the interior point approach can be faster than non–interior point methods.

Table 4.1 demonstrates numerically the global as well as the local behavior of the IPCTR algorithm. For computing the tangential component during the inner trust region iteration procedure, IPCTR requires no more than 10 (preconditioned) CG iterations. In this table we see that the barrier parameter sequence tends very rapidly to zero. The observed rate of convergence is quadratic for sufficiently small barrier parameters and close to the solution of the nonlinear SDP-problem, i.e., consider the last three outer iterations. The behavior illustrated in Table 4.1 is typical for the IPCTR algorithm. During the first few outer iterations, for relatively big μ's, it computes an approximate solution of the corresponding barrier problem very inaccurately. Then, as μ tends to zero, IPCTR solves the barrier problems with a higher accuracy and reaches the region of fast convergence during the last few outer iterations. Moreover, in a vicinity of the solution of (1.1), IPCTR requires only one inner iteration per outer iteration; for example, it reduces μ at each iteration. This local behavior is typical for IPCTR. For all examples that we have tested, we have always observed this behavior of IPCTR close to a solution of (1.1). Finally, these examples justify the usefulness of the interior point strategy combined with trust region globalization for solving the problem class considered in this paper. They indicate that the IPCTR algorithm is potentially efficient for the solution of nonconvex and nonlinear SDP-problems of the form (1.1).

Collection of test examples. Since there is no test set available for nonlinear SDP-problems (like the CUTE [4] test set for nonlinear programming), we have built our own collection of test examples for the special class of nonlinear SDP-problems considered in this paper. To test the IPCTR algorithm, we selected several examples

TABLE 4.1
Convergence behavior of IPCTR for (AC2).

j	μ_j	ϵ_j	k	$\|\|(\nabla \ell_F^{\mu_j})_k\|\| + \|\|h^k\|\|$	$\|\|h^k\|\|$	δ_k	i_{cg}
0	1.0e–01	3.1e+01	0	1.3167e+02	5.085e+00	1.317e+02	–
			1	3.1825e+01	1.120e+00	2.633e+02	1
			2	2.8282e+01	1.195e–01	5.267e+02	7
1	9.8e–04	1.2e+01	0	2.7823e+01	1.286e–00	5.267e+02	–
			1	5.8808e+01	3.974e–01	1.053e+03	10
			⋮	⋮	⋮	⋮	⋮
			48	6.1222e–00	1.135e–00	1.000e+10	10
2	1.2e–05	4.9e–00	0	6.1227e–00	1.135e–00	1.000e+10	–
			1	2.5075e–00	6.685e–01	1.000e+10	3
3	1.7e–09	1.9e–00	0	3.2909e–00	1.080e–00	1.000e+10	–
			1	4.0449e–00	1.125e–00	1.000e+10	10
			⋮	⋮	⋮	⋮	⋮
			7	1.8711e–00	5.158e–01	4.129e–01	1
4	2.9e–17	7.9e–01	0	2.8029e–00	1.081e–00	4.129e–01	–
			1	4.5173e–00	7.110e–01	4.129e–01	2
			⋮	⋮	⋮	⋮	⋮
			6	1.1405e–01	1.318e–02	8.259e–01	3
5	8.8e–33	1.3e–02	0	1.4669e–01	3.953e–02	8.259e–01	–
			1	6.2357e–03	2.515e–03	1.652e–00	6
6	7.7e–64	3.8e–05	0	1.1697e–02	7.546e–03	1.652e–00	–
			1	1.9680e–05	5.866e–06	3.303e–00	8
7	6.1e–128	1.0e–08	0	3.8270e–05	1.760e–05	3.303e–00	–
			1	5.4517e–11	1.755e–11	6.607e–00	9

from the test collection described in [25]. This testing environment contains different examples for designing optimal static or reduced order output feedback controllers. As already noted, such problems can be transformed to nonlinear SDP-problems of the form (1.1). For more information about (static or reduced order) output feedback design, we refer the interested reader to [22] and the references therein.

In Table 4.2, we give the results of our preliminary tests. For each example, we report the name along with its dimensions n, p, and r; the problem type (SOF for static output feedback, ROF for reduced order output feedback); and the number of the overall inner iterations performed by IPCTR. For comparison, the table also shows the number of iterations taken by CTR and ARMIJO. Finally, it lists the initial pair chosen for (F_0, L_0) (SLPMM for the result of the SLPMM procedure, OTHER if otherwise taken). Note that the main computational work for IPCTR, CTR, and ARMIJO is comparable. For example, all of these methods compute a step using a CG procedure similar to Algorithm 2.1. Therein, in IPCTR and CTR, we must solve three linear equations per CG iteration, while in ARMIJO, we need to solve five linear equations in every CG iteration. Moreover, in IPCTR and CTR the same number of linear equations must be solved during each trust region iteration of these algorithms, while in ARMIJO, we need to solve five linear equations. Thus, we can compare the performance of the algorithms by the number of iterations that they need for finding an approximate solution of (1.1). Observe that for the test examples listed in Table 4.2, the IPCTR algorithm outperforms the CTR and the ARMIJO rival. Furthermore, the Newton method combined with the Armijo line search strategy fails

TABLE 4.2
Several examples from the test problem set [25].

Name	n	p	r	Type	ARMIJO	CTR	IPCTR	(F_0, L_0) ?
(AC1)	5	3	3	SOF	115	70	39	SLPMM
(AC1)	5	3	3	SOF	272	94	83	OTHER
(AC2)	5	3	3	SOF	–	14	11	SLPMM
(AC2)	5	3	3	SOF	–	94	68	OTHER
(AC3)	5	2	4	SOF	59	14	12	SLPMM
(AC4)	5	3	3	SOF	299	18	14	OTHER
(AC9)	10	2	3	ROF	–	83	63	OTHER
(NN2)	4	2	3	SOF	–	36	25	SLPMM
(NN3)	4	2	3	SOF	59	55	43	SLPMM
(NN6)	5	3	2	SOF	–	1921	721	OTHER
(HE4)	8	4	6	SOF	333	75	59	SLPMM
(HE4)	8	4	6	SOF	–	14	14	OTHER
(MFP)	4	3	2	SOF	–	28	15	SLPMM
(EB1)	10	1	1	SOF	642	27	27	SLPMM
(CM1)	20	1	2	SOF	–	28	23	SLPMM
(CM2)	60	1	2	SOF	–	37	30	SLPMM
(ROC1)	9	2	2	ROF	–	709	151	SLPMM
(ROC1)	9	2	2	ROF	–	725	101	OTHER
(ROC2)	9	2	2	ROF	–	6982	373	OTHER
(ROC4)	5	3	3	ROF	–	2251	379	OTHER
(MS1)	9	4	4	ROF	–	65	39	OTHER
(MS2)	6	3	3	ROF	571	56	31	OTHER

in most of the cases (indicated by the dash) for finding an approximate solution of these nonconvex SDP-problems with accuracy less than $\epsilon_{out} = 10^{-8}$. In contrast to this, the trust region strategies are very robust. For all tested examples, they find a solution within the desired accuracy independently of the starting point. But the IPCTR method is often much faster than the CTR rival.

Table 4.2 contains only a selection of test examples from the test problem set of [25]. However, for other examples in this test collection we can draw the same observations. IPCTR requires fewer or not more iterations than CTR or ARMIJO, and in most of the cases, ARMIJO needs more iterations than the trust region methods. But note that, as presented in Example 1, for some examples all three algorithms can behave very similarly. Moreover, near a solution of (1.1), we have always observed the fast local convergence rates together with the one-step termination of the inner loop and the reduction of the barrier parameter in every iteration as demonstrated in Table 4.1. Finally, we have never observed a failure of the IPCTR algorithm in any of the examples tested. The IPCTR method had always reached an approximate solution within the desired accuracy. Again, this indicates that the IPCTR algorithm is very robust and potentially efficient for the solution of nonconvex and nonlinear SDP-problems of the form considered in this paper.

REFERENCES

[1] F. ALIZADEH, *Interior point methods in semidefinite programming with applications to combinatorial optimization*, SIAM J. Optim., 5 (1995), pp. 13–51.

[2] F. ALIZADEH, J.-P. A. HAEBERLY, AND M. L. OVERTON, *Primal-dual interior-point methods for semidefinite programming: Convergence rates, stability and numerical results*, SIAM J. Optim., 8 (1998), pp. 746–768.

[3] B. D. O. ANDERSON AND J. B. MOORE, *Linear Optimal Control*, Prentice–Hall, Englewood Cliffs, NJ, 1971.

[4] I. Bongartz, A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *CUTE: Constrained and unconstrained testing environment*, ACM Trans. Math. Software, 21 (1995), pp. 123–160.
[5] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, SIAM Stud. Appl. Math. 15, SIAM, Philadelphia, 1994.
[6] R. H. Byrd, J. C. Gilbert, and J. Nocedal, *A trust region method based on interior point techniques for nonlinear programming*, Math. Program., 89 (2000), pp. 149–185.
[7] M. Chilali and P. Gahinet, *$\mathcal{H}_\infty$ design with pole placement constraints: An LMI approach*, IEEE Trans. Automat. Control, 41 (1996), pp. 358–367.
[8] A. R. Conn, N. I. M. Gould, D. Orban, and Ph. L. Toint, *A primal-dual trust-region algorithm for non-convex nonlinear programming*, Math. Program., 87 (2000), pp. 215–249.
[9] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Trust-Region Methods*, MPS-SIAM Series on Optimization, SIAM, Philadelphia, 2000.
[10] J. E. Dennis, Jr., M. El-Alem, and M. C. Maciel, *A global convergence theory for general trust-region-based algorithms for equality constrained optimization*, SIAM J. Optim., 7 (1997), pp. 177–207.
[11] J. E. Dennis, M. Heinkenschloss, and L. N. Vicente, *Trust-region interior-point SQP algorithms for a class of nonlinear programming problems*, SIAM J. Control Optim., 36 (1998), pp. 1750–1794.
[12] M. El-Alem, *A global convergence theory for the Celis–Dennis–Tapia trust-region algorithm for constrained optimization*, SIAM J. Numer. Anal., 28 (1991), pp. 266–290.
[13] A. S. El-Bakry, R. A. Tapia, T. Tsuchiya, and Y. Zhang, *On the formulation and theory of Newton interior-point method for nonlinear programming*, J. Optim. Theory Appl., 89 (1996), pp. 507–541.
[14] A. Forsgren and Ph. E. Gill, *Primal-dual interior methods for nonconvex nonlinear programming*, SIAM J. Optim., 8 (1998), pp. 1132–1152.
[15] D. M. Gay, M. L. Overton, and M. H. Wright, *A primal-dual interior point method for nonconvex nonlinear programming*, in Advances in Nonlinear Programming, Y. Yuan, ed., Kluwer Academic Publishers, Norwell, MA, 1998, pp. 31–56.
[16] J. C. Geromel and P. B. Gapski, *Synthesis of positive real $\mathcal{H}_2$ controllers*, IEEE Trans. Automat. Control, 42 (1997), pp. 988–992.
[17] Y. S. Hung and A. G. J. MacFarlane, *Multivariable Feedback: A Quasi-Classical Approach*, Lecture Notes in Control and Inform. Sci. 40, Springer-Verlag, Berlin, Heidelberg, New York, 1982.
[18] F. Jarre, *A QQP-Minimization Method for Semidefinite and Smooth Nonconvex Programs*, Technical report, University of Notre Dame, Notre Dame, IN, 2000.
[19] P. P. Khargonekar and M. A. Rotea, *Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control: A convex optimization approach*, IEEE Trans. Automat. Control, 36 (1991), pp. 824–837.
[20] F. S. Kupfer, *Reduced SQP in Hilbert Space with Applications to Optimal Control*, Ph.D. thesis, Universität Trier, FB IV-Mathematik, Trier, Germany, 1992.
[21] F.-S. Kupfer and E. W. Sachs, *A prospective look at SQP methods for semilinear parabolic control problems*, in Optimal Control of Partial Differential Equations, K.-H. Hoffmann and W. Krabs, eds., Lecture Notes in Control and Inform. Sci. 149, Springer, 1991, pp. 143–157.
[22] F. Leibfritz, *Static Output Feedback Design Problems*, Shaker Verlag, Aachen, Germany, 1998.
[23] F. Leibfritz, *Computational Design of Stabilizing Static Output Feedback Controllers*, Technical report 99–01, Universität Trier, Germany, 1999.
[24] F. Leibfritz, *Static Output Feedback Design by Using a Newton-SQP Interior Point Method*, Technical report 99–03, Universität Trier, Trier, Germany, 1999.
[25] F. Leibfritz, *A Collection of Test Examples for a Special Class of Nonlinear SDP-Problems*, Technical report, Universität Trier, Trier, Germany, 2000.
[26] F. Leibfritz *An LMI-based algorithm for designing suboptimal static $\mathcal{H}_2/\mathcal{H}_\infty$ output feedback controllers*, SIAM J. Control Optim., 39 (2001), pp. 1711–1735.
[27] F. Leibfritz and E. M. E. Mostafa, *Optimal Static Output Feedback Design by Using a Trust Region Interior Point Method,* Technical report 00–03, Universität Trier, Trier, Germany, 2000.
[28] F. Leibfritz and E. M. E. Mostafa, *Trust Region Methods for Solving the Optimal Output Feedback Design Problem*, Technical report 00–01, Universität Trier, Trier, Germany, 2000.
[29] El-S. M. E. Mostafa, *Efficient Trust-Region Methods in Numerical Optimization*, Ph.D. thesis, Department of Mathematics, Faculty of Science, Alexandria University, Alexandria, Egypt, 2000.
[30] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM Stud. Appl. Math. 13, SIAM, Philadelphia, 1994.

[31] T. Rautert and E. W. Sachs, *Computational design of optimal output feedback controllers*, SIAM J. Optim., 7 (1997), pp. 837–852.
[32] F. Rendel, R. J. Vanderbei, and H. Wolkowicz, *Max-min eigenvalue problems, primal-dual interior point algorithms, and trust region subproblems*, Optim. Methods Softw., 5 (1995), pp. 1–16.
[33] T. Steihaug, *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., 20 (1983), pp. 626–637.
[34] H. T. Toivonen and P. M. Mäkilä, *Newton's method for solving parametric linear quadratic control problems*, Internat. J. Control, 46 (1987), pp. 897–911.
[35] L. Vandenberghe and S. Boyd, *Semidefinite programming*, SIAM Rev., 38 (1996), pp. 49–95.
[36] R. J. Vanderbei and D. F. Shanno, *An interior point algorithm for nonconvex nonlinear programming*, Comput. Optim. Appl., 13 (1999), pp. 231–252.
[37] J. Zhang and D. Zhu, *A trust-region typed dogleg method for nonlinear optimization*, Optimization, 21 (1990), pp. 543–557.

SIAM J. OPTIM.
Vol. 12, No. 4, pp. 1075–1089

A GLOBALLY CONVERGENT AUGMENTED LAGRANGIAN PATTERN SEARCH ALGORITHM FOR OPTIMIZATION WITH GENERAL CONSTRAINTS AND SIMPLE BOUNDS*

ROBERT MICHAEL LEWIS† AND VIRGINIA TORCZON‡

Abstract. We give a pattern search method for nonlinearly constrained optimization that is an adaption of a bound constrained augmented Lagrangian method first proposed by Conn, Gould, and Toint [*SIAM J. Numer. Anal.*, 28 (1991), pp. 545–572]. In the pattern search adaptation, we solve the bound constrained subproblem approximately using a pattern search method. The stopping criterion proposed by Conn, Gould, and Toint for the solution of the subproblem requires explicit knowledge of derivatives. Such information is presumed absent in pattern search methods; however, we show how we can replace this with a stopping criterion based on the pattern size in a way that preserves the convergence properties of the original algorithm. In this way we proceed by successive, inexact, bound constrained minimization without knowing exactly how inexact the minimization is. As far as we know, this is the first provably convergent direct search method for general nonlinear programming.

Key words. augmented Lagrangian, constrained optimization, direct search, nonlinear programming, pattern search

AMS subject classifications. 65K05, 90C30, 90C56

PII. S1052623498339727

1. Introduction.

In this paper we consider the extension of pattern search methods to nonlinearly constrained optimization problems of the form

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & c(x) = 0, \\ & \ell \le x \le u, \end{array} \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $c(x) = (c_1(x), \ldots, c_m(x))$. We allow the possibility that some of the variables are unbounded either above or below by permitting $\ell_j, u_j = \pm\infty$, $j \in \{1, \ldots, n\}$. This formulation assumes that any general inequality constraints have been converted into equality constraints by the introduction of nonnegative slack variables, leaving bounds as the only explicit inequality constraints.

The pattern search method presented here is an adaptation of an augmented Lagrangian method due to Conn, Gould, and Toint [4], which is the basis for the subroutine `AUGLG` in the `LANCELOT` optimization package [5]. The method of Conn, Gould, and Toint involves successive bound constrained minimization of an augmented Lagrangian. Since the analysis of pattern search methods has recently been extended to

*Received by the editors May 22, 1998; accepted for publication (in revised form) September 7, 2001; published electronically April 19, 2002.

http://www.siam.org/journals/siopt/12-4/33972.html

†Department of Mathematics, College of William & Mary, P.O. Box 8795, Williamsburg, VA 23187–8795 (buckaroo@math.wm.edu). This research was supported by the National Aeronautics and Space Administration under NASA contract NAS1-97046 while the author was in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23681–2199.

‡Department of Computer Science, College of William & Mary, P.O. Box 8795, Williamsburg, VA 23187–8795 (va@cs.wm.edu). This research was supported by the National Science Foundation under grant CCR-9734044, and by the National Aeronautics and Space Administration under NASA Contract NAS1-97046 while the author was in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23681–2199.

bound constrained minimization [17, 18], an adaptation of the augmented Lagrangian method of Conn, Gould, and Toint to pattern search naturally suggests itself. Furthermore, the multiplier update of Algorithm 1 in [4] does not involve information about derivatives of the objective or constraints, so the augmented Lagrangian approach is consistent with the derivative-free nature of pattern search algorithms.

Since there exist broad classes of pattern search methods for unconstrained [16, 28] and bound constrained minimization [17, 18], it seems to us natural to first extend pattern search methods to nonlinearly constrained minimization via algorithms that proceed by successive unconstrained or bound constrained minimization, such as the augmented Lagrangian method we discuss here. In the absence of information about derivatives of the objective and constraints, it is difficult to design pattern search algorithms for general nonlinearly constrained minimization that produce only feasible directions or feasible iterates. This is due to the fact that a pattern in a pattern search algorithm would need to include a sufficiently rich set of search directions to capture any feasible improvement in the objective. When nonlinear constraints are present, it is not clear how to design such a pattern without first-order information.

We show that despite the absence of an explicit estimation of any derivatives (a characteristic of pattern search methods), our pattern search augmented Lagrangian approach exhibits all of the first-order convergence properties of the original algorithm of Conn, Gould, and Toint. This at first is surprising, since the original algorithm allows its subproblems to be solved approximately, and the stopping criterion for the solution of the subproblems is based on the magnitude of a measure of first-order stationarity for bound constrained minimization. This information is not explicitly available in a direct search method. However, as we discuss in section 5.1, there is a correlation between the size of the pattern in bound constrained pattern search and the amount of local feasible descent. Using this correlation, we are able to establish convergence to Karush–Kuhn–Tucker points of (1.1) even without explicit knowledge of derivatives. That is, we are able to proceed by successive, inexact minimization of the augmented Lagrangian via pattern search methods, even without knowing exactly how inexact the minimization is.

This is the main contribution of the work presented here. Otherwise, the extension of pattern search to constrained minimization by means of the augmented Lagrangian approach of Conn, Gould, and Toint is straightforward, due to the strength and generality of the convergence analysis presented in [4].

The question of treating general nonlinear constraints with direct search minimization algorithms has a long history, beginning with the original work on direct search methods. Rosenbrock [24] proposed treating constraints, using his rotating directions method, by redefining the objective near the boundary of the feasible region in a way that would tend to keep the iterates feasible, a form of penalization. Similar ideas for modifying the objective in the case of bound constraints are discussed by Spendley, Hext, and Himsworth [26] and Nelder and Mead [21] in connection with their simplex-based methods. In these approaches the objective is given a suitably large value (in the case of minimization) at all infeasible points.

More systematic approaches to penalization have also appeared. The treatment of inequality constraints via exact, nonsmooth penalization (though not by that name) appears as early as the work of Hooke and Jeeves [12]. More recently, Kearsley and Glowinski [10, 13] have applied pattern search methods with exact, nonsmooth penalization to equality constrained problems arising in control. Weisman's `MINIMAL` algorithm (see [11]) applies the pattern search algorithm of Hooke and Jeeves to a nonsmooth quadratic penalty function and incorporates an element of random search.

Davies and Swann [6], in connection with applying the pattern search method of Hooke and Jeeves to constrained optimization, recommend the use of the reciprocal barrier method of Carroll [3] (also see [8]).

A direct search method for constrained minimization that has proven popular in application is the complex method of Box [2], which was originally developed to address difficulties encountered with Rosenbrock's method. In this algorithm, the objective is sampled at a broader set of points than in the simplex-based methods, to try to avoid premature termination. There is also an element of random search involved. The `ACSIM` algorithm of Dixon [7] combines ideas from the simplex method of Nelder and Mead and the complex method with elements of hem-stitching and quadratic modeling to accelerate convergence.

In the special case of bound constraints, Spendley [25] also suggested the expedient of simply setting to the corresponding bound any variable that would otherwise become infeasible when applying the simplex algorithm of Nelder and Mead. In [14], Keefer proposed a hybrid, feasible iterates algorithm for bound constrained minimization that uses the algorithm of Nelder and Mead for variables suitably far from their bounds, and the method of Hooke and Jeeves for variables that are on or near one of their bounds, since the pattern in the algorithm of Hooke and Jeeves conforms in a natural way to the boundary of the feasible region. In the case of linear constraints there is the algorithm of May [19], which is an extension of Mifflin's derivative-free unconstrained minimization method [20]. May's algorithm also takes into account the particular geometry of the feasible region. May's algorithm is notable because it is accompanied by convergence analysis results; however, it is not a direct search method, insofar as it does rely on a model of the objective.

Others have proposed modifications of the method of Hooke and Jeeves along the lines of feasible directions algorithms. These methods involve a limited calculation of sensitivity information to compute feasible directions at the boundary of the feasible region if the algorithm appears to have stalled. Klingman and Himmelblau [15] give an algorithm with a simple construction of a suitable feasible direction. The method of Glass and Cooper [9] is more sophisticated and computes a new search direction by solving a linear programming problem involving a linear approximation of the objective and constraints, just as one would in a derivative-based feasible directions algorithm.

Finally, we note the flexible tolerance method of Paviani and Himmelblau [11, 22]. This algorithm, based on the method of Nelder and Mead, alternatively attempts to reduce the objective and constraint violation, depending on the extent to which the iterates are infeasible.

These proposals for direct search algorithms for constrained minimization have often proven effective in practice but have not been accompanied by any convergence analysis. In historical context, this is not surprising. The early development of direct search methods (particularly the work cited here) predates even the first global convergence analysis of practical unconstrained minimization algorithms using the Armijo–Goldstein–Wolfe conditions. Instead, in the 1960s the emphasis in optimization was on the development of new computational methods, not on proving theoretical properties. And, in fact, some of the heuristics in the approaches discussed above do not always work in practice. For instance, see Box's comments on Rosenbrock's method in [2], and Keefer's comments on Box's method in [14].

Nevertheless, some of the heuristics proposed in this early research on direct search methods can be placed on firm theoretical grounds. For instance, Keefer's observation that the pattern search method of Hooke and Jeeves works particularly

well for bound constrained problems can be explained analytically [17]. In this paper we apply analytical and algorithmic advances made since the original development of direct search methods to construct a direct search method for general nonlinear programming with provable first-order global convergence properties.

2. The augmented Lagrangian method of Conn, Gould, and Toint. We base our augmented Lagrangian pattern search method on Algorithm 1 of [4]. To facilitate comparison of the pattern search approach with the original algorithm, we adhere to the notation of [4] throughout.

The augmented Lagrangian in [4] is

$$\Phi(x;\lambda,S,\mu) = f(x) + \sum_{i=1}^{m} \lambda_i c_i(x) + \frac{1}{2\mu}\sum_{i=1}^{m} s_{ii} c_i(x)^2. \tag{2.1}$$

The vector $\lambda = (\lambda_1,\ldots,\lambda_m)^T$ is the Lagrange multiplier estimate for the equality constraints, μ is the penalty parameter, and the entries s_{ii} of the diagonal matrix S are positive weights. The equality constraints of (1.1) are incorporated in the augmented Lagrangian Φ, while the simple bounds are left explicit. For a particular choice of multiplier estimate $\lambda^{(k)}$, penalty parameter $\mu^{(k)}$, and scaling $S^{(k)}$, we define

$$\Phi^{(k)}(x) = \Phi(x;\lambda^{(k)},S^{(k)},\mu^{(k)}).$$

Following [4], unless otherwise indicated by an explicit argument, $\nabla_x\Phi^{(k)}$ denotes

$$\nabla_x\Phi^{(k)} \equiv \nabla_x\Phi^{(k)}(x^{(k)}) = \nabla_x\Phi(x^{(k)};\lambda^{(k)},S^{(k)},\mu^{(k)})$$

for the iterate $x^{(k)}$.

Conn, Gould, and Toint define the first-order Lagrange multiplier update to be

$$\overline{\lambda}(x,\lambda,S,\mu) = \lambda + Sc(x)/\mu. \tag{2.2}$$

This is the Hestenes–Powell multiplier update for the augmented Lagrangian (2.1). For the purposes of a pattern search augmented Lagrangian approach, which assumes no explicit knowledge of derivative information, one appears to have no choice other than some variant of the Hestenes–Powell multiplier update, since other multiplier update formulae (such as those discussed in [1, 27]) require information about derivatives.

We denote by P the projection onto the set $B = \{\, x \mid \ell \le x \le u \,\}$; P is defined componentwise by

$$(P[x])_i = \begin{cases} \ell_i & \text{if } x_i \le \ell_i, \\ u_i & \text{if } x_i \ge u_i, \\ x_i & \text{otherwise.} \end{cases}$$

Given $x \in B$ and a vector v, we define

$$P(x,v) = x - P[x-v].$$

The geometrical meaning of $P(x,v)$ is illustrated in Figure 2.1. If x is interior to B, then $P(x,v) = 0$ if and only if $v = 0$, while if x is on the boundary of B, then $P(x,v) = 0$ if and only if v is normal to B (in the sense of convex analysis).

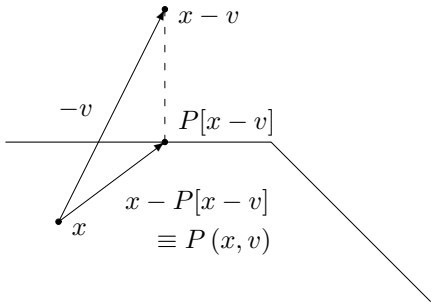


FIG. 2.1. *An example of* $P(\cdot, \cdot)$.

At iteration k of the original augmented Lagrangian algorithm described in [4], we approximately solve the subproblem

$$\begin{array}{ll} \text{minimize} & \Phi^{(k)}(x) \\ \text{subject to} & \ell \le x \le u. \end{array} \tag{2.3}$$

The degree to which this subproblem must be solved is given by

$$\| P(x^{(k)}, \nabla_x \Phi^{(k)}) \| \le \omega^{(k)}, \tag{2.4}$$

where $\omega^{(k)}$ is updated at each iteration k. (Unless otherwise noted, we use $\| \cdot \|$ to denote the Euclidean vector norm or its induced matrix norm.)

We adapt Algorithm 1 in [4] to pattern search by solving the bound constrained subproblem (2.3) using a bound constrained pattern search method. However, pattern search methods do not have recourse to derivatives or explicit approximations thereof.

For that reason we replace (2.4) with a new stopping criterion that is based on the size of the pattern. As we discuss in section 5, we retain the convergence properties of the original Conn, Gould and Toint algorithm because the size of the pattern and the stationarity condition (2.4) are correlated, even though we do not have explicit control of $\|P(x^{(k)}, \nabla_x \Phi^{(k)})\|$.

3. Bound constrained pattern search algorithms. We next review relevant features of the general pattern search method for the bound constrained problem

$$\begin{array}{ll} \text{minimize} & F(x) \\ \text{subject to} & \ell \le x \le u. \end{array} \tag{3.1}$$

We concentrate only on features that we need for the results that follow. For a full discussion, see [17, 18].

3.1. The bound constrained pattern search method. Figure 3.1 outlines the generalized pattern search method for minimization with bound constraints. To define a particular pattern search method, we must specify the pattern (a set of possible trial directions) $\Pi^{(j)}$, the bound constrained exploratory moves algorithm used to find a feasible step $s^{(j)}$, and the algorithms for updating $\Pi^{(j)}$ and $\Delta^{(j)}$. The options and conditions accompanying these choices are discussed in [17, 18].

We make use of the following observations.

1. At iteration j, the step $s^{(j)}$ must be in the set $\Delta^{(j)}\Pi^{(j)}$, and $x^{(j)} + s^{(j)}$ must be feasible. We allow the possibility $s^{(j)} = 0$.

Let $x^{(0)} \in B$ and $\Delta^{(0)} > 0$ be given.
For $j = 0, 1, \ldots,$

a) Compute $F(x^{(j)})$.

b) Determine a step $s^{(j)}$ using a bound constrained exploratory moves algorithm.

c) If $F(x^{(j)} + s^{(j)}) < F(x^{(j)})$, then $x^{(j+1)} = x^{(j)} + s^{(j)}$. Otherwise $x^{(j+1)} = x^{(j)}$.

d) Update $\Pi^{(j)}$ and $\Delta^{(j)}$.

FIG. 3.1. *The generalized pattern search method for bound constrained problems.*

2. The pattern $\Pi^{(j)}$ contains a distinguished subset of trial directions known as the *core pattern*, which we denote by $\Gamma^{(j)}$. The core pattern is constructed to ensure that if $x^{(j)}$ is not a constrained stationary point of (3.1), then at least one element p in $\Gamma^{(j)}$ is a feasible direction of descent. The elements of $\Gamma^{(j)}$ are required to be uniformly bounded in norm: there exists d^*, independent of j, such that $\| p \| \leq d^*$ for all $p \in \Gamma^{(j)}$.
3. We may accept any step $s^{(j)}$ that yields simple decrease in F.
4. If

$$\min \left\{ F(x^{(j)} + s) \;\mid\; s \in \Delta^{(j)}\Gamma^{(j)},\ x^{(j)} + s \in B \right\} < F(x^{(j)}), \tag{3.2}$$

then the step $s^{(j)}$ returned by the bound constrained exploratory moves algorithm must also produce simple decrease on $F(x^{(j)})$. (Note, though, that $s^{(j)}$ need not be an element of $\Delta^{(j)}\Gamma^{(j)}$.)
5. The update of $\Delta^{(j)}$ depends on whether or not the step $s^{(j)}$ satisfied the simple decrease criterion.

3.2. The update of $\Delta^{(j)}$. The conditions under which we allow $\Delta^{(j)}$ to be *reduced* are at the heart of the results that follow. The aim of the update of $\Delta^{(j)}$ is to force a strict reduction in F. An iteration with $F(x^{(j)} + s^{(j)}) < F(x^{(j)})$ is *successful*; otherwise, the iteration is *unsuccessful*. We cannot update $\Delta^{(j)}$ in an arbitrary manner, as discussed in [17, 18]. However, for the purposes of analyzing the augmented Lagrangian pattern search algorithm, the update of $\Delta^{(j)}$ can be summarized as

$$\text{if } F(x^{(j)} + s^{(j)}) < F(x^{(j)}), \text{ then } \Delta^{(j+1)} \geq \Delta^{(j)}; \tag{3.3}$$

$$\text{if } F(x^{(j)} + s^{(j)}) \geq F(x^{(j)}), \text{ then } \Delta^{(j+1)} < \Delta^{(j)}. \tag{3.4}$$

If an iteration is successful, it may be possible to increase the scale factor $\Delta^{(j)}$, but $\Delta^{(j)}$ is not allowed to decrease. If an iteration is unsuccessful, the scale factor $\Delta^{(j)}$ must be decreased.

4. The pattern search augmented Lagrangian method. We now state the augmented Lagrangian pattern search algorithm. At iteration k in the outermost loop of the algorithm, we denote by $\{x^{(k,j)}\}$ the sequence of iterates produced in the solution of (2.3) via a bound constrained pattern search algorithm. Thus, for a given value of k, we look for an approximate solution of the subproblem (2.3) starting from $x^{(k,0)} = x^{(k)}$ and proceed until we find j^* such that $x^{(k,j^*)}$ solves (2.3) to an acceptable degree. We modify the original algorithm by replacing the stopping criterion (2.4) for the solution of the subproblems with one that is suitable for pattern search while still allowing us to use the analysis from [4].

In order to relate the stopping criterion in the pattern search solution of the subproblems to the multiplier estimates and the penalty parameter, we introduce the function

$$\theta(\lambda, \mu) = (1 + \| \lambda \| + 1/\mu)^{-1}.$$

We note that any function $\theta(\lambda, \mu)$ such that $\theta(\lambda, \mu) = O((\| \lambda \| + 1/\mu)^{-1})$ as $(\| \lambda \| + 1/\mu) \to \infty$ suffices for the purposes of proving convergence.

Our algorithm closely resembles Algorithm 1 in [4]. We use boxes to highlight the elements that differ.

Step 0 [Initialization]. An initial vector of Lagrange multiplier estimates $\lambda^{(0)}$ is given. The positive constants η_0, μ_0, ω_0, $\tau < 1$, $\gamma_1 < 1$, $\boxed{\delta_* \ll 1}$, $\eta_* \ll 1$, α_ω, β_ω, α_η, and β_η are specified. The diagonal matrices S_1 and S_2, for which $0 < S_1^{-1} \leq S_2 < \infty$, are given. (The inequalities are to be understood elementwise for the diagonal elements.) Set $\mu^{(0)} = \mu_0$, $\alpha^{(0)} = \min(\mu^{(0)}, \gamma_1)$, $\omega^{(0)} = \omega_0(\alpha^{(0)})^{\alpha_\omega}$, $\boxed{\delta^{(0)} = \theta(\lambda^{(0)}, \mu^{(0)})\omega^{(0)}}$, $\eta^{(0)} = \eta_0(\alpha^{(0)})^{\alpha_\eta}$, and $k = 0$.

Step 1 [Inner iteration]. Define a scaling matrix $S^{(k)}$ for which $S_1^{-1} \leq S^{(k)} \leq S_2$.

Set $x^{(k,0)} = x^{(k)}$. Apply a bound constrained pattern search method to

$$\begin{array}{ll} \text{minimize} & \Phi^{(k)}(x) \\ \text{subject to} & \ell \leq x \leq u \end{array} \tag{4.1}$$

to find the first iteration j^* for which the scale factor is sufficiently small; that is,

$$\Delta^{(k,j^*)} \leq \delta^{(k)}. \tag{4.2}$$

Set $x^{(k)} = x^{(k,j^*)}$.

If

$$\| c(x^{(k)}) \| \leq \eta^{(k)},$$

execute Step 2. Otherwise, execute Step 3.

Step 2 [Test for convergence and update Lagrange multiplier estimates]. If $\boxed{\delta^{(k)} \leq \delta^*}$ and $\|c(x^{(k)})\| \leq \eta_*$, stop. Otherwise, set

$$\begin{aligned} \lambda^{(k+1)} &= \overline{\lambda}(x^{(k)}, \lambda^{(k)}, S^{(k)}, \mu^{(k)}), \\ \mu^{(k+1)} &= \mu^{(k)}, \\ \alpha^{(k+1)} &= \min(\mu^{(k+1)}, \gamma_1), \\ \omega^{(k+1)} &= \omega^{(k)}(\alpha^{(k+1)})^{\beta_\omega}, \end{aligned}$$

$$\boxed{\delta^{(k+1)} = \theta(\lambda^{(k+1)}, \mu^{(k+1)})\ \omega^{(k+1)}},$$

$$\eta^{(k+1)} = \eta^{(k)}(\alpha^{(k+1)})^{\beta_\eta},$$

increment k by one, and go to Step 1.

Step 3 [Reduce the penalty parameter]. Set

$$\begin{aligned}
\lambda^{(k+1)} &= \lambda^{(k)}, \\
\mu^{(k+1)} &= \tau\mu^{(k)}, \\
\alpha^{(k+1)} &= \min(\mu^{(k+1)}, \gamma_1), \\
\omega^{(k+1)} &= \omega_0(\alpha^{(k+1)})^{\alpha_\omega}, \\
\boxed{\delta^{(k+1)} = \theta(\lambda^{(k+1)}, \mu^{(k+1)})\ \omega^{(k+1)}}&, \\
\eta^{(k+1)} &= \eta_0(\alpha^{(k+1)})^{\alpha_\eta},
\end{aligned}$$

increment k by one, and go to Step 1.

We have replaced the stopping criterion (2.4) for the inner iteration of Algorithm 1 in [4] with (4.2), which is based on the scale factor Δ, because we do not assume explicit information about the derivatives. The remaining modifications to Algorithm 1 in [4] concern the management of the sequence $\{\delta^{(k)}\}$, which controls the stopping criterion we have introduced.

5. Convergence analysis. We now discuss the convergence properties of the augmented Lagrangian pattern search algorithm. As we shall see, altering the original algorithm by solving the bound constrained subproblem via pattern search leaves the convergence properties of the original algorithm almost entirely unchanged.

In [4], Conn, Gould, and Toint call a component of $x^{(k)}$ *floating* if

$$\ell_i < x_i^{(k)} - (\nabla_x \Phi^{(k)})_i < u_i.$$

For a convergent subsequence $\{x^{(k)}\}$, $k \in K$, with limit point x^*, they define the index set

$$I_1 = \left\{ i \mid x_i^{(k)} \text{ are floating for all } k \in K \text{ sufficiently large and } \ell_i < x_i^* < u_i \right\}$$

and let $\hat{A}(x)$ denote the corresponding columns of the Jacobian of $c(x)$, where $A(x)$ is the entire Jacobian of $c(x)$.

The following assumptions are made in [4].

AS1. *The functions $f(x)$ and $c(x)$ are twice continuously differentiable for all $x \in B$.*

AS2. *The iterates $\{x^{(k)}\}$ considered lie within a closed, bounded domain Ω.*

AS3. *The matrix $\hat{A}(x^*)$ has column rank no smaller than m at any limit point x^* of the sequences $\{x^{(k)}\}$ considered in this paper.*

In addition, in order to be assured that a bound constrained pattern search algorithm applied to the subproblem (4.1) will find an iterate satisfying (4.2), we assume the following.

PS1. *For a given k, the set $B \cap \{x \mid \Phi^{(k)}(x) \le \Phi^{(k)}(x^{(k,0)})\}$ is compact.*

That is, we assume compactness of the set of $x \in B$ for which the augmented Lagrangian is no larger than the value of the augmented Lagrangian at the point at which we begin the solution of the subproblem. Under hypothesis (PS1), we are assured that in the inner iteration (the pattern search minimization of the bound constrained augmented Lagrangian),

$$\liminf_{j \to +\infty} \Delta^{(k,j)} = 0$$

(see [17, 18]). Thus the termination criterion (4.2) will eventually be satisfied, the pattern search solution of the augmented Lagrangian subproblem will halt, and the overall iteration of the pattern search augmented Lagrangian algorithm is well-defined.

We also assume the following uniform bound.

PS2. *There exists d^* such that for all k and j we have $\|p\| \le d^*$ for all $p \in \Gamma^{(k,j)}$.*

This uniformity in the pattern search algorithms used in the successive minimization of the augmented Lagrangian is not at all restrictive. For instance, one could simply choose for all (k, j) a single set Γ.

5.1. The relationship between the pattern size and stationarity. The following result is the key to analyzing the augmented Lagrangian pattern search method. The important observation in connection with the stopping criterion (4.2) is that at unsuccessful iterations of the pattern search solution of (4.1) there is a correlation between $\Delta^{(k,j)}$ and the stationarity of the augmented Lagrangian. The rules for updating $\Delta^{(k,j)}$, summarized in (3.3) and (3.4), mean that $\Delta^{(k,j)}$ can drop below $\delta^{(k)}$ only at an unsuccessful iteration of the pattern search. Thus (4.2) can occur only at an unsuccessful iteration of the solution of the subproblem. At an unsuccessful iteration, we do *not* find an acceptable step in $\Delta^{(k,j)}\Gamma^{(k,j)}$; that is,

$$\Phi^{(k)}(x^{(k,j)} + s) \ge \Phi^{(k)}(x^{(k,j)}) \text{ for all } s \in \Delta^{(k,j)}\Gamma^{(k,j)} \text{ with } (x^{(k,j)} + s) \in B.$$

Now, the set of steps s for $s \in \Delta^{(k,j)}\Gamma^{(k,j)}$ includes a set of generators for the tangent cone of the bound constrained feasible region [17, 18]. The fact that none of the steps s yields a feasible trial point with a smaller value of $\Phi^{(k)}$ tells us something about the size of $\|P(x^{(k)}, \nabla_x \Phi^{(k)})\|$. Proposition 5.1 makes this precise and shows that the weaker condition (4.2) we have introduced guarantees that (2.4) will be satisfied.

PROPOSITION 5.1. *There exists $C_{5.1}$, independent of k, such that*

$$\| P(x^{(k)}, \nabla_x \Phi^{(k)}) \| \le C_{5.1}\ \omega^{(k)}$$

for all iterations k of the pattern search augmented Lagrangian method.

Proof. Given k, we know that at the end of Step 1, the inner iteration, $x^{(k)} \equiv x^{(k,j^*)}$ for some j^*. For convenience, let

$$q^{(k,j^*)} = P(x^{(k)}, \nabla_x \Phi^{(k)}) \equiv P(x^{(k,j^*)}, \nabla_x \Phi^{(k)}(x^{(k,j^*)})).$$

First suppose

$$\Delta^{(k,j^*)} \ge \frac{\| q^{(k,j^*)} \|_\infty}{d^*}. \tag{5.1}$$

Then (5.1), (4.2), and the rule for updating $\delta^{(k)}$ in either Step 2 or Step 3 give us

$$\| q^{(k,j^*)} \|_\infty \ \le\ d^* \Delta^{(k,j^*)} \ \le\ d^* \delta^{(k)} \ \le\ d^* \omega^{(k)},$$

and thus

$$\| q^{(k,j^*)} \| \le n^{\frac{1}{2}} d^* \omega^{(k)}. \tag{5.2}$$

On the other hand, suppose

$$\Delta^{(k,j^*)} < \frac{\| q^{(k,j^*)} \|_\infty}{d^*}.$$

The proof of Proposition 5.2 in [17] shows that if $\Delta^{(k,j^*)} < \|q^{(k,j^*)}\|_\infty/d^*$, there is a step $s \in \Delta^{(k,j^*)}\Gamma^{(k,j^*)}$ such that $x^{(k,j^*)} + s \in B$ and

$$\nabla_x \Phi^{(k)}(x^{(k,j^*)})^T s < -n^{-\frac{1}{2}} \parallel q^{(k,j^*)} \parallel \parallel s \parallel. \tag{5.3}$$

Because $x^{(k,j^*)}$ is an unsuccessful iterate, we know from (3.2) that

$$0 \leq \Phi^{(k)}(x^{(k,j^*)} + s) - \Phi^{(k)}(x^{(k,j^*)}). \tag{5.4}$$

At the same time we have

$$\Phi^{(k)}(x^{(k,j^*)} + s) - \Phi^{(k)}(x^{(k,j^*)}) = \nabla_x \Phi^{(k)}(\xi)^T s \tag{5.5}$$

for some ξ in the line segment $(x^{(k,j^*)}, x^{(k,j^*)} + s)$ connecting $x^{(k,j^*)}$ and $x^{(k,j^*)} + s$. Thus from (5.4), (5.5), and (5.3), we obtain

$$\begin{aligned} 0 &\leq \Phi^{(k)}(x^{(k,j^*)} + s) - \Phi^{(k)}(x^{(k,j^*)}) \\ &= \nabla_x \Phi^{(k)}(x^{(k,j^*)})^T s + (\nabla_x \Phi^{(k)}(\xi) - \nabla_x \Phi^{(k)}(x^{(k,j^*)}))^T s \\ &\leq -n^{-\frac{1}{2}} \parallel q^{(k,j^*)} \parallel \parallel s \parallel + \parallel \nabla_x \Phi^{(k)}(\xi) - \nabla_x \Phi^{(k)}(x^{(k,j^*)}) \parallel \parallel s \parallel, \end{aligned}$$

which yields

$$\parallel q^{(k,j^*)} \parallel \leq n^{\frac{1}{2}} \parallel \nabla_x \Phi^{(k)}(\xi) - \nabla_x \Phi^{(k)}(x^{(k,j^*)}) \parallel. \tag{5.6}$$

Applying the mean-value theorem again, for some $\zeta \in (x^{(k,j^*)}, \xi)$ we have

$$\nabla_x \Phi^{(k)}(\xi) - \nabla_x \Phi^{(k)}(x^{(k,j^*)}) = \nabla^2_{xx} \Phi^{(k)}(\zeta)(\xi - x^{(k,j^*)}),$$

and thus

$$\begin{aligned} \|\nabla_x \Phi^{(k)}(\xi) - \nabla_x \Phi^{(k)}(x^{(k,j^*)})\| &\leq \|\nabla^2_{xx} \Phi^{(k)}(\zeta)\| \, \|\xi - x^{(k,j^*)}\| \\ &\leq \|\nabla^2_{xx} \Phi^{(k)}(\zeta)\| \, \|s\|. \end{aligned} \tag{5.7}$$

Now,

$$\nabla^2_{xx} \Phi^{(k)}(\zeta) = \nabla^2_{xx} f(\zeta) + \sum_{i=1}^m \lambda_i^{(k)} \nabla^2 c_i(\zeta) + \frac{1}{\mu^{(k)}} \left(\nabla c(\zeta) S \nabla c(\zeta)^T + \sum_{i=1}^m s_{ii} c_i(\zeta) \nabla^2 c_i(\zeta) \right).$$

By construction, $\omega^{(k)} \to 0$, so $\delta^{(k)} \to 0$, so by (AS2), ζ lies in a compact subset that is independent of k. Furthermore, the bound $S^{(k)} \leq S_2$ is independent of k. Thus we can find M, independent of k, such that

$$\parallel \nabla^2_{xx} \Phi^{(k)}(\zeta) \parallel \; \leq \; M + M \parallel \lambda^{(k)} \parallel + M \frac{1}{\mu^{(k)}} \; = \; M/\theta(\lambda^{(k)}, \mu^{(k)}).$$

Returning to (5.7), we have

$$\parallel \nabla_x \Phi^{(k)}(\xi) - \nabla_x \Phi^{(k)}(x^{(k,j^*)}) \parallel \leq \left(M/\theta(\lambda^{(k)}, \mu^{(k)}) \right) \parallel s \parallel. \tag{5.8}$$

Thus from (5.6), (5.8), the fact that $s \in \Delta^{(k,j^*)}\Gamma^{(k,j^*)}$, and (4.2), we have

$$\begin{aligned} \parallel q^{(k,j^*)} \parallel &\leq n^{\frac{1}{2}} \parallel \nabla_x \Phi^{(k)}(\xi) - \nabla_x \Phi^{(k)}(x^{(k,j^*)}) \parallel \\ &\leq n^{\frac{1}{2}} \left(M/\theta(\lambda^{(k)}, \mu^{(k)}) \right) \parallel s \parallel \\ &\leq n^{\frac{1}{2}} d^* \left(M/\theta(\lambda^{(k)}, \mu^{(k)}) \right) \Delta^{(k,j^*)} \\ &\leq n^{\frac{1}{2}} d^* \left(M/\theta(\lambda^{(k)}, \mu^{(k)}) \right) \delta^{(k)}. \end{aligned}$$

Finally, the rule for updating $\delta^{(k)}$ in either Step 2 or Step 3 is $\delta^{(k)} = \theta(\lambda^{(k)}, \mu^{(k)})\omega^{(k)}$, whence

$$\| q^{(k,j^*)} \| \leq n^{\frac{1}{2}} d^* M \omega^{(k)}. \tag{5.9}$$

Combining (5.2) and (5.9) yields the proposition. ∎

5.2. Convergence results. Proposition 5.1 means that the asymptotic behavior of $\| P(x^{(k)}, \nabla_x \Phi^{(k)}) \|$ in the augmented Lagrangian pattern search algorithm is like that of the same quantity in the original algorithm. This, in turn, allows us to piggyback the convergence analysis for the augmented Lagrangian pattern search algorithm on that for the original augmented Lagrangian algorithm in [4]. Because of Proposition 5.1, the original proofs of all these results still hold.

The first convergence result corresponds to Theorem 4.4 and Lemma 4.3 in [4]. Let

$$g_L(x; \lambda) = \nabla f(x) + \sum_{i=1}^{m} \lambda_i \nabla c_i(x).$$

THEOREM 5.2. *Assume that* (AS1) *holds. Let x^* be any limit point of the sequence $\{x^{(k)}\}$ generated by the augmented Lagrangian pattern search algorithm for which* (AS2) *and* (AS3) *hold, and let K be the set of indices of an infinite subsequence of the $x^{(k)}$ whose limit is x^*. Then the following hold:*

(i) $c(x^*) = 0$.

(ii) *x^* is a Karush–Kuhn–Tucker point (first-order stationary point) for problem* (1.1)*, λ^* is the corresponding vector of Lagrange multipliers, and the sequence $\{\overline{\lambda}(x^{(k)}, \lambda^{(k)}, S^{(k)}, \mu^{(k)})\}$ converges to λ^* for $k \in K$.*

(iii) *There are positive constants a_1, a_2, s_1 and an integer k_0 such that*

$$\| \overline{\lambda}(x^{(k)}, \lambda^{(k)}, S^{(k)}, \mu^{(k)}) - \lambda^* \| \leq a_1 \omega^{(k)} + a_2 \| x^{(k)} - x^* \|$$

and

$$\| c(x^{(k)}) \| \leq s_1 (a_1 \omega^{(k)} \mu^{(k)} + \mu^{(k)} \| \lambda^{(k)} - \lambda^* \| + a_2 \mu^{(k)} \| x^{(k)} - x^* \|)$$

for all $k \geq k_0$, $k \in K$.

(iv) *The gradients $\nabla_x \Phi^{(k)}$ converge to $g_L(x^*; \lambda^*)$ for $k \in K$.*

As in [4], under additional assumptions we obtain stronger results. Following [4], if J_1 and J_2 are any index sets, and $H_L(x^*, \lambda^*)$ is the Hessian of the Lagrangian, then $H_L(x^*, \lambda^*)_{[J_1, J_2]}$ is the matrix formed by taking the rows and columns of $H_L(x^*, \lambda^*)$ indexed by J_1 and J_2, respectively, while $A(x^*)_{[J_1]}$ is the matrix formed by taking the columns of $A(x^*)$ indexed by J_1. We then make the following assumptions.

AS4. *The second derivatives of the functions $f(x)$ and $c_i(x)$ are Lipschitz continuous at all points within Ω.*

AS5. *Suppose that (x^*, λ^*) is a Karush–Kuhn–Tucker point for problem* (1.1) *and that*

$$J_1 = \{\, i \mid (g_L(x^*; \lambda^*))_i = 0 \text{ and } \ell_i < x_i^* < u_i \,\},$$
$$J_2 = \{\, i \mid (g_L(x^*; \lambda^*))_i = 0 \text{ and } (x_i^* = \ell_i \text{ or } x_i^* = u_i) \,\}.$$

Then we assume that the matrix

$$\begin{bmatrix} H_L(x^*, \lambda^*)_{[J,J]} & (A(x^*)_{[J]})^T \\ A(x^*)_{[J]} & 0 \end{bmatrix}$$

is nonsingular for all sets J, where J is any set made up from the union of J_1 and any subset of J_2.

The next result is Lemma 5.1 from [4]. This result also holds for the augmented Lagrangian pattern search algorithm and relates the convergence of the iterates to the error in the multipliers, a relationship characteristic of augmented Lagrangian methods [1, 27]. Again, the proof in [4] holds for the pattern search variant because of Proposition 5.1.

LEMMA 5.3. *Suppose that* (AS1) *holds. Let $\{x^{(k)}\} \subset B$, $k \in K$, be a subsequence which converges to the Karush–Kuhn–Tucker point x^* for which* (AS2), (AS4), *and* (AS5) *hold, and let λ^* be the corresponding vector of Lagrange multipliers. Assume that $\{\lambda^{(k)}\}$, $k \in K$, is any sequence of vectors, that $\{S^{(k)}\}$, $k \in K$, is any sequence of diagonal matrices satisfying $0 < S_1^{-1} \leq S^{(k)} \leq S_2 < \infty$, and that $\{\mu^{(k)}\}$, $k \in K$, form a nonincreasing sequence of positive scalars, so that the product $\mu^{(k)} \| \lambda^{(k)} - \lambda^* \|$ converges to zero as k increases. Now, suppose further that*

$$\| P(x^{(k)}, \nabla_x \Phi^{(k)}) \| \leq \omega^{(k)},$$

where the $\omega^{(k)}$ are positive scalar parameters which converge to zero as $k \in K$ increases. Then there are positive constants $\overline{\mu}$, a_3, a_4, a_5, a_6, and s_1 and an integer value k_0 so that if $\mu^{(k_0)} \leq \overline{\mu}$, then

$$\| x^{(k)} - x^* \| \leq a_3 \omega^{(k)} + a_4 \mu^{(k)} \| \lambda^{(k)} - \lambda^* \|, \tag{5.10}$$

$$\| \overline{\lambda}(x^{(k)}, \lambda^{(k)}, S^{(k)}, \mu^{(k)}) - \lambda^* \| \leq a_5 \omega^{(k)} + a_6 \mu^{(k)} \| \lambda^{(k)} - \lambda^* \|,$$

and

$$\| c(x^{(k)}) \| \leq s_1 (a_5 \omega^{(k)} \mu^{(k)} + (\mu^{(k)} + a_6 (\mu^{(k)})^2) \| \lambda^{(k)} - \lambda^* \|) \tag{5.11}$$

for all $k \geq k_0$, $k \in K$.

The following is Corollary 5.2 in [4].

COROLLARY 5.4. *Suppose that the conditions of Lemma* 5.3 *hold and that $\hat{\lambda}^{(k+1)}$ is any Lagrange multiplier estimate for which*

$$\| \hat{\lambda}^{(k+1)} - \lambda^* \| \leq a_{16} \| x^{(k)} - x^* \| + a_{17} \omega^{(k)}$$

for some positive constants a_{16} and a_{17} and all $k \in K$ sufficiently large. Then there are positive constants $\overline{\mu}$, a_3, a_4, a_5, a_6, s_1 and an integer value k_0 so that if $\mu^{(k_0)} \leq \overline{\mu}$, then (5.10),

$$\| \hat{\lambda}^{(k+1)} - \lambda^* \| \leq a_5 \omega^{(k)} + a_6 \mu^{(k)} \| \lambda^{(k)} - \lambda^* \|,$$

and (5.11) *hold for all $k \geq k_0$, $k \in K$.*

We also inherit the following result indicating that we may generally expect the penalty parameter to remain bounded away from zero. This is Theorem 5.3 in [4]. Taken together with the convergence of the multiplier estimates, this means that the stopping tolerance for the inexact minimization of the augmented Lagrangian is decreasing at the same rate as in the original algorithm. However, in section 6 of [4] the authors show that in the case of nonunique limit points one can have $\mu^{(k)} \to 0$, in which case the stopping tolerance δ^k decreases more like $(\mu^{(k)})^2$.

THEOREM 5.5. *Suppose that the iterates $\{x^{(k)}\}$ of the augmented Lagrangian pattern search algorithm converge to the single limit point x^*, that* (AS1), (AS2), (AS4),

and (AS5) *hold, and that* α_η *and* β_η *satisfy* $\alpha_\eta < \min(1, \alpha_\omega)$ *and* $\beta_\eta < \min(1, \beta_\omega)$. *Then there is a constant* $\mu > 0$ *such that* $\mu^{(k)} > \mu$ *for all* k.

The proof of Theorem 5.5 makes use of the fact that $\|P(x^{(k)}, \nabla_x \Phi^{(k)})\| = O(\omega^{(k)})$, whereas the proofs of the preceding convergence results require only that

$$\| P(x^{(k)}, \nabla_x \Phi^{(k)}) \| \to 0.$$

Finally, we have the following result on the rate of convergence of the outer iteration, corresponding to Theorem 5.5 in [4].

THEOREM 5.6. *Under the assumptions of Theorem* 5.5, *the iterates* $x^{(k)}$ *and the Lagrange multiplier estimates* $\overline{\lambda}^{(k)}$ *of the augmented Lagrangian pattern search algorithm are at least R-linearly convergent with R-factor at most* $\hat{\mu}^{\min(\beta_\omega, \beta_\eta)}$, *where* $\hat{\mu} = \min[\gamma_1, \mu]$ *and where* μ *is the smallest value of the penalty parameter generated by the algorithm in question.*

6. Application to inequality constrained minimization. Special consideration is due to the general problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g(x) \le 0, \\ & \ell \le x \le u, \end{array} \tag{6.1}$$

converted into the form (1.1) via the introduction of nonnegative slack variables:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g(x) + z = 0, \\ & \ell \le x \le u, \\ & z \ge 0. \end{array} \tag{6.2}$$

The augmented Lagrangian associated with (6.2) is

$$\Phi(x, z; \lambda, S, \mu) = f(x) + \lambda^T (g(x) + z) + \frac{1}{2\mu} \sum_{i=1}^{m} s_{ii} (g_i(x) + z_i)^2. \tag{6.3}$$

Explicit equality constraints may also be present in (6.1); we ignore them here for brevity.

The introduction of slacks increases the dimension of the bound constrained subproblem that we must solve at each outer iteration. Unfortunately, increases in dimension usually cause a degradation in performance for pattern search methods. We can avoid this increase in dimension because of the simple way in which the slacks z enter into (6.3). One approach [1, 23] is to note that, given x, we can minimize $\Phi(x, z; \lambda, S, \mu)$ explicitly in z for $z \ge 0$. This leads to a subproblem in x alone:

$$\begin{array}{ll} \text{minimize} & \Phi(x, z(x); \lambda, S, \mu) \\ \text{subject to} & \ell \le x \le u, \end{array}$$

where

$$\Phi(x, z(x); \lambda, S, \mu) = f(x) + \frac{\mu}{2} \sum_{i=1}^{m} \frac{1}{s_{ii}} \left(\max\left(0, \lambda_i + \frac{s_{ii}}{\mu} g_i(x)\right)^2 - \lambda_i^2 \right).$$

The multiplier update formula (2.2) is also modified:

$$\overline{\lambda}_i(x, \lambda, S, \mu) = \max(0, \lambda_i + s_{ii} c_i(x)/\mu), \quad i = 1, \ldots, m.$$

See [1] for further discussion. The reduced augmented Lagrangian $\Phi(x, z(x); \lambda, S, \mu)$ has Lipschitz first derivatives. If one were using a quasi-Newton method for the minimization of the augmented Lagrangian, one might be loath to eliminate z, since the resulting problem is not C^2 and one loses any assurance of local superlinear convergence. However, pattern search methods do not have such favorable local convergence properties, and thus ostensibly nothing is lost and much is gained by the reduction of dimension of the subproblems.

7. Conclusion. We have demonstrated that it is possible to construct a globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds. Extensive numerical tests of this algorithm remain to be done. We agree with the perspective of the authors in [4]:

> We have deliberately not included the results of numerical testing as, in our view, the construction of appropriate software is by no means trivial and we wish to make a thorough job of it. We will report on our numerical experience in due course.

This caution is particularly apt in view of the sort of problems to which pattern search is typically applied.

Acknowledgments. We wish to thank the associate editor and the two referees for their careful reading of the paper and their many helpful comments. The presentation is much clearer as a consequence of their efforts.

REFERENCES

[1] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, New York, 1982.
[2] M. J. Box, *A new method of constrained optimization and a comparison with other methods*, The Computer Journal, 8 (1965), pp. 42–52.
[3] C. W. Carroll, *The created response surface technique for optimizing nonlinear restrained systems*, Oper. Res., 9 (1961), pp. 169–185.
[4] A. R. Conn, N. I. M. Gould, and P. L. Toint, *A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds*, SIAM J. Numer. Anal., 28 (1991), pp. 545–572.
[5] A. R. Conn, N. I. M. Gould, and P. L. Toint, *LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A)*, Springer Ser. Comput. Math. 17, Springer-Verlag, New York, 1992.
[6] D. Davies and W. H. Swann, *Review of constrained optimization*, in Optimization, R. Fletcher, ed., Academic Press, New York, 1969, pp. 187–202.
[7] L. C. W. Dixon, *ACSIM—An accelerated constrained simplex technique*, Computer Aided Design, 5 (1973), pp. 22–32.
[8] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, Classics Appl. Math. 4, SIAM, Philadelphia, 1990.
[9] H. Glass and L. Cooper, *Sequential search: A method for solving constrained optimization problems*, J. Assoc. Comput. Machinery, 12 (1965), pp. 71–82.
[10] R. Glowinski and A. J. Kearsley, *On the simulation and control of some friction constrained motions*, SIAM J. Optim., 5 (1995), pp. 681–694.
[11] D. M. Himmelblau, *Applied Nonlinear Programming*, McGraw–Hill, New York, 1972.
[12] R. Hooke and T. A. Jeeves, *Direct search solution of numerical and statistical problems*, J. Assoc. Comput. Machinery, 8 (1961), pp. 212–229.
[13] A. J. Kearsley, *The Use of Optimization Techniques in the Solution of Partial Differential Equations from Science and Engineering*, Ph.D. thesis, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1996; available as Tech. report 96–11, Department of Computational and Applied Mathematics, Rice University, Houston, TX.
[14] D. L. Keefer, *Simpat: Self-bounding direct search method for optimization*, Industrial and Engineering Chemistry Process Design and Development, 12 (1973), pp. 92–99.

[15] W. R. Klingman and D. M. Himmelblau, *Nonlinear programming with the aid of a multiple-gradient summation technique*, J. Assoc. Comput. Machinery, 11 (1964), pp. 400–415.
[16] R. M. Lewis and V. Torczon, *Rank ordering and positive bases in pattern search algorithms*, Tech. report 96–71, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA, 1996.
[17] R. M. Lewis and V. Torczon, *Pattern search algorithms for bound constrained minimization*, SIAM J. Optim., 9 (1999), pp. 1082–1099.
[18] R. M. Lewis and V. Torczon, *Pattern search methods for linearly constrained minimization*, SIAM J. Optim., 10 (2000), pp. 917–941.
[19] J. H. May, *Linearly Constrained Nonlinear Programming: A Solution Method That Does Not Require Analytic Derivatives*, Ph.D. thesis, Yale University, New Haven, CT, 1974.
[20] R. Mifflin, *A superlinearly convergent algorithm for minimization without evaluating derivatives*, Math. Programming, 9 (1975), pp. 100–117.
[21] J. A. Nelder and R. Mead, *A simplex method for function minimization*, The Computer Journal, 7 (1965), pp. 308–313.
[22] D. Paviani and D. M. Himmelblau, *Constrained nonlinear optimization by heuristic programming*, Oper. Res., 17 (1969), pp. 872–882.
[23] R. T. Rockafellar, *A dual approach to solving nonlinear programming problems by unconstrained optimization*, Math. Programming, 5 (1973), pp. 354–373.
[24] H. H. Rosenbrock, *An automatic method for finding the greatest or least value of a function*, The Computer Journal, 3 (1960), pp. 175–184.
[25] W. Spendley, *Nonlinear least squares fitting using a modified simplex minimization method*, in Optimization, R. Fletcher, ed., Academic Press, New York, 1969, pp. 259–270.
[26] W. Spendley, G. R. Hext, and F. R. Himsworth, *Sequential application of simplex designs in optimisation and evolutionary operation*, Technometrics, 4 (1962), pp. 441–461.
[27] R. A. Tapia, *Diagonalized multiplier methods and quasi-Newton methods for constrained optimization*, J. Optim. Theory Appl., 22 (1977), pp. 135–194.
[28] V. Torczon, *On the convergence of pattern search algorithms*, SIAM J. Optim., 7 (1997), pp. 1–25.

SIAM J. OPTIM.
Vol. 12, No. 4, pp. 1090–1099

ON THE CONVERGENCE OF THE CENTRAL PATH IN SEMIDEFINITE OPTIMIZATION*

M. HALICKÁ†, E. DE KLERK‡, AND C. ROOS§

Abstract. The central path in linear optimization always converges to the analytic center of the optimal set. This result was extended to semidefinite optimization in [D. Goldfarb and K. Scheinberg, *SIAM J. Optim.*, 8 (1998), pp. 871–886]. In this paper we show that this latter result is not correct in the absence of strict complementarity. We provide a counterexample, where the central path converges to a different optimal solution. This unexpected result raises many questions. We also give a short proof that the central path always converges in semidefinite optimization by using ideas from algebraic geometry.

Key words. semidefinite optimization, linear optimization, interior point method, central path, analytic center

AMS subject classifications. 90C51, 90C22

PII. S1052623401390793

1. Introduction.

The central path is of fundamental importance in the study of interior point algorithms. The geometric view of the central path is that of an analytic curve which converges to an optimal solution. Most interior point methods "follow" the central path approximately to reach the optimal set. In this paper we will re-examine the convergence property of the central path for semidefinite optimization (SDO). We will show that the characterization of the limit point of the central path as found in [1] is not correct in the absence of strict complementarity. This negative result raises the question of whether the central path always converges. Since there does not seem to be any simple proof of the convergence property in the literature, we include a complete proof as an appendix to this paper.

We first formulate SDO problems in standard form and recall the definition of the central path and some of its properties.

1.1. The central path in SDO.

By S^n we denote the space of all real symmetric $n \times n$ matrices, and for any $M, N \in S^n$ we define

$$M \bullet N = \operatorname{trace}(MN) = \sum_{i,j} m_{ij} n_{ij}.$$

The convex cones of symmetric positive semidefinite matrices and positive definite matrices will be denoted by S^n_+ and S^n_{++}, respectively; $X \succeq 0$ and $X \succ 0$ mean that a symmetric matrix X is positive semidefinite and positive definite, respectively.

*Received by the editors June 13, 2001; accepted for publication (in revised form) October 9, 2001; published electronically April 19, 2002.
http://www.siam.org/journals/siopt/12-4/39079.html

†Faculty of Mathematics, Physics and Informatics, Comenius University, Mlynska dolina, 842 48 Bratislava, Slovakia (halicka@fmph.uniba.sk). The research of this author was supported in part by VEGA grant 1/7675/20 from the Slovak Scientific Grant Agency.

‡Faculty of Information Technology and Systems, Delft University of Technology, P.O. Box 5031, 2600 GA Delft, The Netherlands (E.deKlerk@ITS.TUDelft.nl).

§Department TWI/SSOR, Mekelweg 4, Delft University of Technology, P.O. Box 5031, 2628 CD Delft, The Netherlands (C.Roos@ITS.TUDelft.nl).

We will consider the following primal-dual pair of semidefinite programs in the standard form:

$$\text{(P)} \qquad \min_{X \in S^n} \{C \bullet X \ : \ A_i \bullet X = b_i \ (i = 1, \ldots, m) \ X \succeq 0\},$$

$$\text{(D)} \qquad \max_{y \in \mathbb{R}^m, S \in S^n} \left\{ b^T y \ : \ \sum_{i=1}^m A_i y_i + S = C, \ S \succeq 0 \right\},$$

where $A_i \in S^n$ $(i = 1, \ldots, m)$ and $C \in S^n$, $b \in \mathbb{R}^m$. We assume that A_i $(i = 1, \ldots, m)$ are linearly independent. The solutions X and (y, S) will be referred to as feasible solutions if they satisfy the primal and dual constraints, respectively.

We assume that both (P) and (D) satisfy the interior point condition; i.e., there exists (X^0, S^0, y^0) such that

$$A_i \bullet X^0 = b_i \ (i = 1, \ldots, m), \ X^0 \succ 0, \quad \text{and} \quad \sum_{i=1}^m A_i y_i^0 + S^0 = C, \ S^0 \succ 0.$$

The primal and dual feasible sets will be denoted by $\mathcal{P}$ and $\mathcal{D}$, respectively, and $\mathcal{P}^*$ and $\mathcal{D}^*$ will denote the respective optimal sets. It is well known that under our assumptions both $\mathcal{P}^*$ and $\mathcal{D}^*$ are nonempty and bounded. The optimality conditions for (P) and (D) are

$$\text{(1)} \qquad \begin{array}{rcll} A_i \bullet X & = & b_i, & X \succeq 0 \quad (i = 1, \ldots, m), \\ \sum_{i=1}^m A_i y_i + S & = & C, & S \succeq 0, \\ XS & = & 0. & \end{array}$$

A strictly complementary solution can be defined as an optimal solution pair (X, S) satisfying the rank condition: rank X + rank $S = n$. Contrary to linear optimization (LO), for SDO the existence of the strictly complementary solution is not generally ensured.

We now relax the optimality conditions (1) to

$$\text{(2)} \qquad \begin{array}{rcll} A_i \bullet X & = & b_i, & X \succeq 0 \quad (i = 1, \ldots, m), \\ \sum_{i=1}^m A_i y_i + S & = & C, & S \succeq 0, \\ XS & = & \mu I, & \end{array}$$

where I is the identity matrix and $\mu \geq 0$. It is easy to see that for $\mu = 0$ (2) gives (1), and hence it may have more than one solution. On the other hand, it is well known that for $\mu > 0$ system (2) has a unique solution, denoted by $(X(\mu), S(\mu), y(\mu))$ (see, e.g., [6]). As for LO, this solution is seen as the parametric representation of an analytic curve (the *central path*) in terms of the parameter $\mu > 0$.

It has been shown that the central path for SDO shares many properties with the central path for LO. First, the basic property was established that the central path restricted to $0 < \mu \leq \bar{\mu}$ for some $\bar{\mu} > 0$ is bounded, and thus it has limit points as $\mu \downarrow 0$ in the optimal set [9], [5]. Then it was shown that the limit points are in the relative interior of the optimal set [5], [1]. Finally, it was claimed by Goldfarb and Scheinberg [1] that the central path converges for $\mu \downarrow 0$ to the so-called analytic

center of the optimal solution set. Although this result has been widely cited in the recent literature, we will show in this paper that it is not correct in the absence of strict complementarity. Let us mention that the correct proofs of this fact—however, only under the assumption of strict complementarity—were given in [9] and later in [4].

Since the central path does not converge to the analytic center in general, it is natural to ask whether it always converges. The convergence property seems to be a "folkloric" result that is already mentioned on page 74 of the review paper [10] (without supplying references or a proof). In [7] the convergence of the central path for the linear complementarity problem (LCP) is proven with the help of some results from algebraic geometry. In [6], Kojima, Shindoh, and Hara mention that this proof for LCP can be extended to the monotone semidefinite complementarity problem (which is equivalent to SDO) without giving a formal proof. A more general result was shown in [2], where convergence is proven for a class of convex SDO problems that includes SDO.

We include a complete convergence proof in an appendix to this paper, which also uses some ideas from the theory of algebraic sets, but in a different manner from [7]. It is also much shorter, and requires fewer auxiliary results, than the proof in [2].

1.2. Analytic center of the optimal solution set. A pair of optimal solutions $(X, S) \in \mathcal{P}^* \times \mathcal{D}^*$ is called a *maximally complementary solution pair* to the pair of problems (P) and (D) if it maximizes rank (X)+rank (S) over all optimal solution pairs. The set of maximally complementary solutions coincides with the relative interior of $(\mathcal{P}^* \times \mathcal{D}^*)$. Another characterization is as follows: $(\bar{X}, \bar{S}) \in \mathcal{P}^* \times \mathcal{D}^*$ is maximally complementary if and only if

$$\mathcal{R}(\hat{X}) \subseteq \mathcal{R}(\bar{X}) \quad \forall \hat{X} \in \mathcal{P}^*, \quad \mathcal{R}(\hat{S}) \subseteq \mathcal{R}(\bar{S}) \quad \forall \hat{S} \in \mathcal{D}^*,$$

where $\mathcal{R}$ denotes the range space. For proofs of these characterizations see [5] and [1] and the references therein.

Let $\bar{X}$ and $\bar{S}$ be a pair of maximally complementary optimal solutions. Denote

$$|B| := \text{rank } \bar{X}, \quad \text{and} \quad |N| := \text{rank } \bar{S}.$$

Obviously, $|B| + |N| \leq n$. Without loss of generality (applying an orthonormal transformation of problem data, if necessary) we can assume that

$$\bar{X} = \begin{bmatrix} \bar{X}^B & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \bar{S} = \begin{bmatrix} \bar{0} & 0 & 0 \\ 0 & \bar{S}^N & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

where $\bar{X}^B \in S_{++}^{|B|}$ and $\bar{S}^N \in S_{++}^{|N|}$. Therefore, each optimal solution pair $(\hat{X}, \hat{S})$ is of the form

$$\hat{X} = \begin{bmatrix} \hat{X}^B & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \hat{S} = \begin{bmatrix} \hat{0} & 0 & 0 \\ 0 & \hat{S}^N & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

where $\hat{X}^B \in S_{+}^{|B|}$ and $\hat{S}^N \in S_{+}^{|N|}$, since $\mathcal{R}(\hat{X}) \subseteq \mathcal{R}(\bar{X})$ and $\mathcal{R}(\hat{S}) \subseteq \mathcal{R}(\bar{S})$.

In what follows we consider the partition of any $M \in S^n$ corresponding to the above optimal partition so that

$$M = \begin{bmatrix} M^B & M^{BN} & M^{BT} \\ M^{NB} & M^N & M^{NT} \\ M^{TB} & M^{TN} & M^T \end{bmatrix}.$$

We denote by $\mathcal{I} = \{B, BN, BT, NB, N, NT, TB, TN, T\}$ the index set corresponding to the optimal partition. If we refer to all the blocks of M except M^B, we will write M^i $(i \in \mathcal{I} - B)$.

Now, the optimal solutions sets can be characterized by using the block partition:

$$\mathcal{P}^* = \left\{ X \ : \ A_i^B \bullet X^B = b_i \ (\ i = 1, \ldots, n),\ X^B \in S_+^{|B|},\ X^k = 0 \ (k \in \mathcal{I} - B) \right\},$$

$$\mathcal{D}^* = \left\{ (S, y) \ : \ \sum_{i=1}^m A_i^N y_i + S^N = C^N,\ S^N \in S_+^{|N|}, \right.$$
$$\left. \sum_{i=1}^m A_i^k y_i = C^k,\ S^k = 0 \ (k \in \mathcal{I} - N) \right\}.$$

The analytic centers of these sets are defined as follows: $X^a \in \mathcal{P}^*$ is the analytic center of $\mathcal{P}^*$ if

$$(X^a)^B = \arg \max_{X^B \in S_{++}^{|B|}} \left\{ \ln \det X^B \ : \ A_i^B \bullet X^B = b_i,\ i = 1, \ldots, m \right\},$$

and $(y^a, S^a) \in \mathcal{D}^*$ is the analytic center of $\mathcal{D}^*$ if

$$\left(y^a, (S^a)^N \right) = \arg \max_{y \in \mathbb{R}^m, S^N \in S_{++}^{|N|}} \left\{ \ln \det S^N \ : \sum_{i=1}^m A_i^N y_i + S^N = C^N, \right.$$
$$\left. \sum_{i=1}^m A_i^k y_i = C^k,\ k \in \ \mathcal{I} - N \right\}.$$

We end this section with two known results about the central path.

LEMMA 1.1 (see [5]). *Any limit point (X^*, S^*) of the central path is a maximally complementary optimal solution; i.e., it satisfies*

$$X^{*B} \succ 0 \quad \textit{and} \quad S^{*N} \succ 0.$$

LEMMA 1.2 (see, e.g., [3, Lemma 2.3.2]). *For any $\mu > 0$ the central path $X(\mu), S(\mu), y(\mu)$ is the analytic center of the level set of the duality gap*

$$\left\{ (X, S, y) \ : \ A_i \bullet X = b_i \ (i = 1, \ldots, m),\ \sum_{i=1}^m A_i y_i + S = C, \right.$$
$$\left. C \bullet X - b^T y = \mu n,\ X \in S_+^n,\ S \in S_+^n \right\}.$$

As a corollary we see that the primal μ-center $X(\mu)$ is the analytic center of the set

$$\{X \ : \ C \bullet X = C \bullet X(\mu), \quad A_i \bullet X = b_i \ (i = 1, \ldots, m), \quad X \succ 0\}.$$

We will use this observation in the next section.

The last two lemmas make it plausible that the central path converges to the analytic center of the optimal set, but in the next section we show that this is not true.

2. Counterexamples. Let $n = 4$, $m = 4$, $b = [1\ 0\ 0\ 0]^T$, and

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad A_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad A_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

The primal problem (P) can be simplified to the following: Minimize x_{44} such that

$$X = \begin{bmatrix} 1 - x_{22} & x_{12} & x_{13} & x_{14} \\ x_{12} & x_{22} & -\frac{1}{2}x_{44} & -\frac{1}{2}x_{33} \\ x_{13} & -\frac{1}{2}x_{44} & x_{33} & 0 \\ x_{14} & -\frac{1}{2}x_{33} & 0 & x_{44} \end{bmatrix} \succeq 0.$$

The optimal set of (P) is given by all the positive semidefinite matrices of the form

$$X^* = \begin{bmatrix} 1 - x_{22} & x_{12} & 0 & 0 \\ x_{12} & x_{22} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3}$$

Solutions of the form X^* are clearly optimal, since $C \succeq 0$ and therefore $\mathrm{Tr}\,(CX) \geq 0$ $\forall X \in \mathcal{P}$.

The dual problem is to maximize y_1 such that

$$S = \begin{bmatrix} -y_1 & 0 & 0 & 0 \\ 0 & -y_1 & -y_3 & -y_2 \\ 0 & -y_3 & -y_2 & -y_4 \\ 0 & -y_2 & -y_4 & 1 - y_3 \end{bmatrix} \succeq 0.$$

Thus the dual problem has a unique optimal solution

$$y_i^* = 0 \quad (i = 1,2,3,4), \quad S^* = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4}$$

It is also easy to see from (3) and (4) that strict complementary does *not* hold. The central path is well defined for this example, since the matrices $A_1, \ldots, A_4$ are clearly linearly independent and strictly feasible solutions exist for both the primal and the dual problem. Indeed,

$$x_{22} = \frac{1}{2}, \quad x_{33} = x_{44} = \frac{1}{4}, \quad x_{ij} = 0 \ \ (i \neq j)$$

defines a positive definite feasible solution for (P), and $y_1 = -1$, $y_2 = -\frac{1}{2}$, and $y_3 = y_4 = 0$ defines a strictly feasible solution of (D).

The analytic center of $\mathcal{P}^*$ is obviously given by

$$\begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

However, we will show that the limit point of the primal central path satisfies

$$X(\mu) \to \begin{bmatrix} 0.4 & 0 & 0 & 0 \\ 0 & 0.6 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \text{ as } \mu \downarrow 0.$$

Due to the structure of feasible $S \in \mathcal{D}$ and the fact that $X(\mu) = \mu S(\mu)^{-1}$, the primal central path has the following structure:

$$X(\mu) = \begin{bmatrix} 1 - x_{22}(\mu) & 0 & 0 & 0 \\ 0 & x_{22}(\mu) & -\frac{1}{2}x_{44}(\mu) & -\frac{1}{2}x_{33}(\mu) \\ 0 & -\frac{1}{2}x_{44}(\mu) & x_{33}(\mu) & 0 \\ 0 & -\frac{1}{2}x_{33}(\mu) & 0 & x_{44}(\mu) \end{bmatrix}.$$

By Lemma 1.2, the point on the central path $X(\mu)$ is, for any $\mu > 0$, the analytic center of a level set. The level set is given by the primal feasibility and a level condition, which is $x_{44} = x_{44}(\mu) > 0$ in our case. This implies that $X(\mu)$ maximizes

$$\det \begin{bmatrix} 1 - x_{22} & 0 & 0 & 0 \\ 0 & x_{22} & -\frac{1}{2}x_{44}(\mu) & -\frac{1}{2}x_{33} \\ 0 & -\frac{1}{2}x_{44}(\mu) & x_{33} & 0 \\ 0 & -\frac{1}{2}x_{33} & 0 & x_{44}(\mu) \end{bmatrix} \tag{5}$$

under the conditions

$$x_{22} \in (0,1), \quad x_{33} > 0, \quad x_{22}x_{33}x_{44}(\mu) - \frac{x_{33}^3 + x_{44}^3(\mu)}{4} > 0.$$

Setting the gradient (with respect to x_{22} and x_{33}) of the determinant in (5) to zero, we obtain the two equations

$$x_{33}(\mu)x_{44}(\mu) - 2x_{22}(\mu)x_{33}(\mu)x_{44}(\mu) + \frac{1}{4}x_{44}(\mu)^3 + \frac{1}{4}x_{33}(\mu)^3 = 0, \tag{6}$$

$$(1 - x_{22}(\mu)) \left(x_{22}(\mu)x_{44}(\mu) - \frac{3}{4}x_{33}(\mu)^2 \right) = 0. \tag{7}$$

Using $x_{22}(\mu) \in (0,1)$, we deduce from (7) that

$$x_{33}(\mu) = \frac{2}{\sqrt{3}}\sqrt{x_{22}(\mu)x_{44}(\mu)}.$$

Substituting this expression in (6) and simplifying, we obtain

$$\frac{2}{\sqrt{3}}\sqrt{x_{22}(\mu)} - \frac{10}{3\sqrt{3}}x_{22}(\mu)^{3/2} + \frac{1}{4}x_{44}(\mu)^{3/2} = 0.$$

In the limit where $\mu \downarrow 0$, we have $x_{44}(\mu) \to 0$. Moreover, we can assume that $x_{22}(\mu)$ is positive in the limit, since the limit point of the central path is maximally complementary (Lemma 1.1). Denoting $\lim_{\mu \downarrow 0} x_{22}(\mu) := x_{22}(0) > 0$, we have

$$\frac{2}{\sqrt{3}}\sqrt{x_{22}(0)} - \frac{10}{3\sqrt{3}}x_{22}(0)^{3/2} = 0,$$

which implies $x_{22}(0) = 0.6$.

An example for the second order cone. The following example shows that the central path may already fail to converge to the analytic center of the optimal set in the special case of second order cone optimization.

Consider the problem of minimizing x_{12} subject to

$$\begin{bmatrix} x_{11} & x_{12} & 0 & 0 & 0 \\ x_{12} & x_{22} & 0 & 0 & 0 \\ 0 & 0 & x_{33} & x_{22} & 0 \\ 0 & 0 & x_{22} & x_{12} & 0 \\ 0 & 0 & 0 & 0 & 1 - (x_{11} + x_{33}) \end{bmatrix} \succeq 0.$$

Note that this problem is equivalent to a second order cone optimization problem: the semidefiniteness constraint is on a block-diagonal matrix with all blocks of size 1×1 or 2×2; it is also easy to check that the notions of analytic center and central path coincide whether the example is viewed as an SDO or as a second order cone problem.

The optimal set is given by all matrices of the above form where $x_{12} = x_{22} = 0$, and the analytic center of the optimal set is given by the optimal solution where $x_{11} = x_{33} = \frac{1}{3}$.

Using exactly the same technique as in the previous example, one can show that the limit point for the central path is $x_{11} = 2/7$, $x_{33} = 3/7$. However, the proof is more technical for this example due to the larger number of variables, and is therefore omitted.

3. Conclusions and future work. The purpose of this paper was twofold:

- to show that the central path in SDO may converge to an optimal solution which is not the analytic center of the optimal set (in the absence of strict complementarity);
- to give a simplified yet rigorous proof that the central path always converges for SDO.

The first result raises some questions:

- Can we give a "geometrical" characterization of the limit point of the central path?
- For which subclasses of SDO problems can one guarantee convergence of the central path to the analytic center of the optimal set?

We therefore hope that the observations in this paper will lead to a renewed interest in the limiting behavior of the central path in SDO.

Appendix. Convergence proof for the central path. In this appendix we give a proof of the convergence of the central path for SDO by using a result from algebraic geometry.

DEFINITION A.1 (algebraic set). *A subset $V \in R^k$ is called an algebraic set if V is the locus of common zeros of some collection of polynomial functions on $\mathbb{R}^k$.*

LEMMA A.2 (curve selection lemma). *Let $V \subset \mathbb{R}^k$ be a real algebraic set, and let $U \subset \mathbb{R}^k$ be an open set defined by finitely many polynomial inequalities:*

$$U = \left\{ x \in \mathbb{R}^k : g_1(x) > 0, \ldots, g_l(x) > 0 \right\}.$$

If $U \cap V$ contains points arbitrarily close to the origin, then there exists an $\epsilon > 0$ and a real analytic curve

$$p : [0, \epsilon) \mapsto \mathbb{R}^k$$

with $p(0) = 0$ and with $p(t) \in U \cap V$ for $t > 0$.

A proof of the curve selection lemma is given in [8, Lemma 3.1, p. 25].

THEOREM A.3. *The central path in semidefinite optimization always converges.*

Proof. Let (X^*, y^*, S^*) be any limit point of the central path of (P) and (D).

With reference to Lemma A.2, let the real algebraic set V be defined via

$$V = \left\{ (\bar{X}, \bar{S}, \bar{y}, \mu) \;\middle|\; \begin{array}{rcl} A_i \bullet \bar{X} & = & 0 \quad (i = 1, \ldots, m), \\ \sum_i (\bar{y}_i) A_i + \bar{S} & = & 0, \\ (\bar{X} + X^*)(\bar{S} + S^*) - \mu I & = & 0, \end{array} \right\}$$

and let the open set U be defined as the set of all $(\bar{X}, \bar{S}, \bar{y}, \mu)$ such that all principal minors of $(\bar{X} + X^*)$ and $(\bar{S} + S^*)$ are positive and $\mu > 0$.

Now $V \cap U$ corresponds to the central path excluding its limit points, in the sense that if $(\bar{X}, \bar{S}, \bar{y}, \mu) \in U \cap V$ then $X(\mu) = (\bar{X} + X^*)$ and $S(\mu) = (\bar{S} + S^*)$, where $X(\mu)$ (respectively, $S(\mu)$) denotes the μ-center of (P) (respectively, (D)) as before.

Moreover, the zero element is in the closure of $V \cap U$, by construction.

The required result now follows from the curve selection lemma. To see this, note that Lemma A.2 implies the existence of an $\epsilon > 0$ and an analytic function $p : [0, \epsilon) \mapsto \mathcal{S}^n \times \mathcal{S}^n \times \mathbb{R}^m \times \mathbb{R}$ such that

$$p(t) = \left(\bar{X}(t), \bar{S}(t), \bar{y}(t), \mu(t) \right) \to (0_{n \times n}, 0_{n \times n}, 0_m, 0) \text{ as } t \downarrow 0, \tag{8}$$

and if $t > 0$, $\big(\bar{X}(t), \bar{S}(t), \bar{y}(t), \mu(t)\big) \in U \cap V$, i.e.,

$$\begin{aligned} A_i \bullet \bar{X}(t) &= 0 \quad (i = 1, \ldots, m), \\ \textstyle\sum_i \bar{y}_i(t) A_i + \bar{S}(t) &= 0, \\ (\bar{X}(t) + X^*)(\bar{S}(t) + S^*) - \mu(t) I &= 0, \end{aligned} \tag{9}$$

and $\bar{X}(t) \succ 0$, $\bar{S}(t) \succ 0$, $\mu(t) > 0$.

Since the centrality system (2) has a unique solution, the system (9) also has a unique solution given by

$$\bar{X}(t) + X^* = X(\mu(t)), \quad \bar{S}(t) + S^* = S(\mu(t))$$

if $t > 0$. By (8), we therefore have

$$\lim_{t \downarrow 0} X(\mu(t)) = X^*, \quad \lim_{t \downarrow 0} S(\mu(t)) = S^*, \quad \lim_{t \downarrow 0} \mu(t) = 0.$$

Since $\mu(t) > 0$ on $(0, \epsilon)$, $\mu(0) = 0$, and μ is analytic on $[0, \epsilon)$, there exists an interval, say $(0, \epsilon')$, where $\frac{d\mu(t)}{dt} > 0$. Therefore the inverse function $\mu^{-1} : \mu(t) \mapsto t$ exists on the interval $(0, \mu(\epsilon'))$. Moreover, $\mu^{-1}(t) > 0 \; \forall \; t \in (0, \mu(\epsilon'))$ and $\lim_{t \downarrow 0} \mu^{-1}(t) = 0$.

This implies that

$$\lim_{t \downarrow 0} X(t) = \lim_{t \downarrow 0} X(\mu(\mu^{-1}(t))) = \lim_{t \downarrow 0} \bar{X}(\mu^{-1}(t)) + X^* = X^*.$$

Similarly, $\lim_{t \downarrow 0} S(t) = S^*$, which completes the proof. □

Acknowledgments. The authors are grateful to Osman Güler for suggesting the approach for the proof of convergence of the central path, and thank Jos Sturm for his assistance with the second order cone example. The first author would also like to thank P. Brunovsky for valuable discussions on the subjects of analytic sets and analytic functions. The authors also thank Luis Graña Drummond for bringing the paper [2] to their attention, and Allen Holder for useful comments.

REFERENCES

[1] D. Goldfarb and K. Scheinberg, *Interior point trajectories in semidefinite programming*, SIAM J. Optim., 8 (1998), pp. 871–886.

[2] L. M. Graña Drummond and Y. Peterzil, *The central path in smooth convex semidefinite programs*, Optimization, to appear.

[3] E. de Klerk, *Interior Point Methods for Semidefinite Programming*, Ph.D. thesis, Delft University of Technology, Delft, The Netherlands, 1999.

[4] E. de Klerk, C. Roos, and T. Terlaky, *Infeasible-start semidefinite programming algorithms via self dual embeddings*, in Topics in Semidefinite and Interior-Point Methods, Fields Inst. Commun. 18, P. M. Pardalos and H. Wolkowicz, eds., AMS, Providence, RI, 1998, pp. 215–236.

[5] E. de Klerk, C. Roos, and T. Terlaky, *Initialization in semidefinite programming via a self-dual, skew-symmetric embedding*, Oper. Res. Lett., 20 (1999), pp. 213–221.

[6] M. Kojima, S. Shindoh, and S. Hara, *Interior-point methods for the monotone semidefinite linear complementarity problem in symmetric matrices*, SIAM J. Optim., 7 (1997), pp. 86–125.

[7] M. Kojima, N. Megiddo, T. Noma, and A. Yoshise, *A Unified Approach to Interior Point Algorithms for Linear Complementarity Problems*, Lecture Notes in Comput. Sci. 538, Springer-Verlag, Berlin, Heidelberg, 1991.

[8] J. Milnor, *Singular Points of Complex Hypersurfaces*, Ann. Math. Stud., Princeton University Press, Princeton, NJ, 1968.

[9] Z.-Q. Luo, J. F. Sturm, and S. Zhang, *Superlinear convergence of a symmetric primal-dual path following algorithm for semidefinite programming*, SIAM J. Optim., 8 (1998), pp. 59–81.

[10] L. Vandenberghe and S. Boyd, *Semidefinite programming*, SIAM Rev., 38 (1996), pp. 49–95.

SIAM J. OPTIM.
Vol. 12, No. 4, pp. 1100–1125

A TRUNCATED NEWTON ALGORITHM FOR LARGE SCALE BOX CONSTRAINED OPTIMIZATION*

FRANCISCO FACCHINEI†, STEFANO LUCIDI†, AND LAURA PALAGI†

Abstract. A method for the solution of minimization problems with simple bounds is presented. Global convergence of a general scheme requiring the approximate solution of a single linear system at each iteration is proved and a superlinear convergence rate is established without requiring the strict complementarity assumption. The algorithm proposed is based on a simple, smooth unconstrained reformulation of the bound constrained problem and may produce a sequence of points that are not feasible. Numerical results and comparison with existing codes are reported.

Key words. bound constrained problem, penalty function, Newton method, conjugate gradient, nonmonotone line search

AMS subject classifications. 90C30, 90C06, 65K05, 65F10

PII. S1052623499359890

1. Introduction.

We are concerned with the solution of simple bound constrained minimization problems of the form

$$\text{(PB)} \qquad \min f(x) \qquad \text{s.t.} \qquad l \le x \le u,$$

where the objective function f is sufficiently smooth, l and u are constant vectors, and the inequalities are valid componentwise. In this paper we introduce a *globally and superlinearly* convergent algorithm that does not require strict complementarity and uses only matrix-vector products, thus being well suited for large scale cases.

The algorithms most widely used to solve problem (PB), when the dimension is small, fall into the *active set* category. At each iteration of methods in this class, a *working set* is defined that is supposed to approximate the set of active constraints at the solution and that is iteratively updated. In general, only a single constraint can be added to or deleted from the active set at each iteration, and this can slow down the convergence rate, especially when dealing with large scale problems. Note, however, that, for the special case of problem (PB), it is possible to envisage algorithms that update the working set more efficiently [17, 23], especially in the quadratic case [12]. Actually, a number of proposals have been made for designing algorithms that quickly identify the correct active set. With regard to (PB), the seminal work is [3] (see also [2]), in which it is shown that if the strict complementarity assumption holds, then it is possible, using a *projection* method, to add to or delete from the current estimated active set many constraints at each iteration and yet find an active set in a finite number of steps. This work has motivated a lot of additional studies on projection techniques, both for the general linearly constrained case and for the box constrained case (see, e.g., [5, 6, 7, 14] and [36, 37]).

Trust region-type algorithms for unconstrained optimization have been successfully extended to handle the presence of bounds on the variables. The global conver-

*Received by the editors August 6, 1999; accepted for publication (in revised form) October 16, 2001; published electronically April 19, 2002. This work was partially supported by project "Algorithms for Complete Systems Optimization"—MURST/COFIN.
http://www.siam.org/journals/siopt/12-4/35989.html
†Dipartimento di Informatica e Sistemistica "A. Ruberti," Università di Roma "La Sapienza," Via Buonarroti 12, 00185 Roma, Italy (facchinei@dis.uniroma1.it, lucidi@dis.uniroma1.it, palagi@dis.uniroma1.it).

gence theory thus developed is very robust [9, 22] and, under appropriate assumptions, it is possible to establish a superlinear convergence rate without requiring strict complementarity [22, 30, 32]. Furthermore, numerical results [10, 22, 32] show that these methods are effective. Another algorithm also based on a trust region philosophy, but in connection with a nonsmooth merit function, is proposed in [41]. A major difference between this latter algorithm and the techniques so far considered is that the iterates generated are not forced to remain feasible throughout.

We finally mention that interior point methods for the solution of problem (PB) are currently an active field of research and that some interesting theoretical results can be obtained in this framework. In particular, in [27] a local superlinearly convergent algorithm that does not require strict complementarity is proposed. Computational experience with this class of methods is still very limited (see [8, 27, 38]).

The method that we propose in this paper for the solution of (PB) does not fit into any of the categories considered above. At each iteration k we compute estimates L^k, U^k of the variables that we suppose will be, respectively, at their lower, upper bounds at the solution, and we also compute an estimate F^k of the variables we believe to be free. This partition of the variables obviously suggests performing an unconstrained minimization in the space of free variables, and this is the typical approach in active set methods. If one aims to develop a locally fast convergent method, an obvious choice for the unconstrained minimization algorithm in the subspace of free variables is the Newton method; this requires the (possibly inexact) solution of the Newton equation

$$\nabla^2 f(x^k)_{F^k F^k} d = -\nabla f(x^k)_{F^k}, \tag{1}$$

where the subscripts F^k attached to a vector or to a matrix denote the subvector or the principal submatrix corresponding to the indices in F^k. There are two main problems with the direction d^k so obtained. On the one hand, the point $x^k + d^k$ is not necessarily feasible; on the other hand, in general the algorithms based on this kind of considerations can be shown to be superlinearly convergent only if strict complementarity holds at the solution. The remedy usually adopted for the first problem is to "artificially" modify the Newton direction given by (1) so as to guarantee that $x^k + d^k$ is feasible. With respect to the second issue, we note that, with the exception of a few recent works [27, 32], superlinear convergence has been proved only under the strict complementarity assumption. The solution we propose to the aforementioned problems is the following. First of all, we observe that the difficulty in obtaining a superlinear convergence rate in the case of a solution which is not strictly complementary is due to the possible loss of curvature information that we have in the subspace of those variables that are active but with a zero multiplier. To overcome this problem we suggest modifying (1) by adding a "correction" term

$$\nabla^2 f(x^k)_{F^k F^k} d = -\nabla f(x^k)_{F^k} + \text{correction}, \tag{2}$$

which brings in the missing information. The correction term in (2) is simple to calculate and is eventually zero if the solution towards which the algorithm converges is strictly complementary or, more generally, if the estimates L^k, U^k, F^k eventually coincide with the sets they approximate (i.e., if exact identification of the active constraints occurs). (See [28, 29] for a similar approach.) The local Newton-type process defined by (2) is shown to be superlinearly convergent without the need for the strict complementarity assumption. However, we still have to face the first problem we mentioned above: the point $x^k + d^k$, where d^k is given by (2), may be infeasible.

Contrary to what is usually done, we prefer to leave the direction d^k untouched, since it is well known that the Newton direction is usually very good. Instead we give the algorithm the freedom to generate infeasible points. Obviously, in this case we cannot directly use the objective function $f(x)$ to measure progress towards optimality, as is usually done by most of the existing algorithms. Instead, we define a very simple differentiable exact penalty function that is used to assess the quality of the points generated by the algorithm. We remark that the penalty function has an extremely simple structure and requires just a few scalar products to be evaluated, so that the overhead for using the penalty function instead of the original objective function is usually negligible. We actually believe that the possibility of developing so-called infeasible-point algorithms for the solution of (PB) is an important contribution of this paper. The only possible disadvantage of our infeasible-point approach is that in some applications the objective function f might not be defined outside the feasible set. From this point of view, it may be important to note that the algorithm we propose allows the user to control the "degree of infeasibility" of the points generated. In fact, while the algorithm is intrinsically infeasible, it only generates points that are contained in a prescribed "enlargement" of the original feasible set of the type $(l - \alpha, u + \beta)$, where α and β are n-dimensional vectors of positive constants that are user-selected. It is then obvious that, in principle, we can force the algorithm to generate points that are only "slightly" infeasible. In any case, if the function f is defined on the whole space, the possibility of violating some of the constraints may give additional, beneficial flexibility.

The algorithm described in this paper is largely based on [18] and [19], where many of the theoretical results reported here were already outlined. The main novelty here is a complete theory for a truncated scheme, suitable for large scale problems, and a rather sophisticated implementation of the resulting algorithm along with extensive numerical results. Below we summarize some relevant features of the algorithm and of its implementation.

(a) A complete global convergence theory is established.
(b) It is shown that our general scheme does not prevent superlinear convergence, in the sense that if a step length of one along the search direction yields superlinear convergence, then, *without requiring strict complementarity*, the step length of one is eventually accepted.
(c) Rapid changes in the working set are allowed.
(d) The points generated by the algorithms at each iteration need not be feasible.
(e) The main computational burden per iteration is given by the approximate solution of a square linear system whose dimension is equal to the number of variables estimated to be nonactive.
(f) A particular truncated Newton-type algorithm is described which falls within the general scheme of point (a) and for which it is possible to establish, under the strong second order sufficient condition but without requiring strict complementarity, a superlinear convergence rate.
(g) Numerical results and comparison of an algorithm's performance with those of Lancelot [11] and Tron [32] are reported.

The paper is organized as follows. In the next section some basic definitions and assumptions are stated. In section 3 a detailed exposition of the local algorithm and of its convergence properties is given. In section 4 the conjugate gradient iterative scheme for the calculation of the direction is described. Section 5 contains the globalization scheme, which is based on a suitable merit function and on a nonmonotone

stabilization scheme. In particular, in section 5.1 the main properties of the differentiable merit function for (PB) are recalled, whereas in section 5.2 the nonmonotone stabilization algorithm is defined. Section 6 is dedicated to numerical experiments.

First we fix the notation. If M is an $n \times n$ matrix with rows M_i, $i = 1, \ldots, n$, and if I and J are index sets such that $I, J \subseteq \{1, \ldots, n\}$, we denote by M_I the $|I| \times n$ submatrix of M consisting of rows M_i, $i \in I$, and we denote by $M_{I,J}$ the $|I| \times |J|$ submatrix of M consisting of elements $M_{i,j}$, $i \in I$, $j \in J$. We indicate by E the $n \times n$ identity matrix. If w is an n vector, we denote by w_I the subvector with components w_i, $i \in I$. Given two n-dimensional vectors w, v, we denote by $w \circ v$ the Hadamard product of the two vectors, namely the vector whose ith component is $w_i v_i$, and by $\max[w, v]$ the componentwise max vector. Using a nonstandard notation that, however, simplifies the presentation, we denote by w^p the vector whose components are w_i^p.

A superscript k is used to indicate iteration numbers; furthermore, we often omit the arguments and write, for example, f^k instead of $f(x^k)$. Finally, by $\|\cdot\|$ we denote the Euclidean norm.

2. Problem formulation and preliminaries. For convenience we recall problem (PB)

$$\text{(PB)} \qquad \min f(x) \qquad \text{s.t.} \qquad l \le x \le u.$$

For simplicity we assume that the objective function $f : \mathbb{R}^n \to \mathbb{R}$ is three times continuously differentiable (even if weaker assumptions can be used; see Remark 5.1) and that $l_i < u_i$ for every $i = 1, \ldots, n$. Note that $-\infty$ and $+\infty$ are admitted values for l_i and u_i, respectively; i.e., we also consider the case in which some (possibly all) bounds are not present. In what follows we indicate by $\mathcal{F}$ the feasible set of (PB), that is, $\mathcal{F} = \{x \in R^n : l \le x \le u\}$.

Let $\alpha \in \mathbb{R}^n$ and $\beta \in \mathbb{R}^n$ be two fixed vectors of positive constants, and let x_a and x_b be two feasible points such that $f(x_a) < f(x_b)$. We define the following functions $a(x)$, $b(x)$, and $c(x)$:

$$a(x) = \alpha - l + x, \quad b(x) = \beta + u - x, \quad c(x) = f(x_b) - f(x). \tag{3}$$

The algorithm proposed in this paper generates a sequence of points which belong to the following open set:

$$\mathcal{S} = \{x \in \mathbb{R}^n : a(x) > 0,\ b(x) > 0,\ \ c(x) > 0\}.$$

Roughly speaking, x_b determines the maximum function value which can be taken by the objective function at the points generated by the algorithm, whereas x_a is used as the starting point. We remark that not every point produced by the algorithm is feasible; feasibility is only ensured in the limit. Note also that α and β are arbitrarily fixed before starting the algorithm and never changed during the minimization process.

To guarantee that no unbounded sequences are produced by the minimization process, we make the following standard assumption.

ASSUMPTION 1. *The set $\mathcal{S}$ is bounded.*

Assumption 1 is automatically satisfied if *either* of the following conditions hold:

- all the variables have finite lower and upper bounds;
- $f(x)$ is radially unbounded, that is, $\lim_{\|x\| \to \infty} f(x) = +\infty$.

For notational convenience, in this paper we consider in detail the results only for the case in which all the variables are box constrained, i.e., the case in which no l_i is $-\infty$ and no u_i is $+\infty$. The extension to the general case is trivial but cumbersome, and therefore we omit it. With this assumption, the KKT conditions for $\bar{x}$ to solve (PB) are

$$\begin{aligned} \nabla f(\bar{x}) - \bar{\lambda} + \bar{\mu} &= 0, \\ \bar{\lambda} \geq 0, \qquad (l - \bar{x})'\bar{\lambda} &= 0, \\ \bar{\mu} \geq 0, \qquad (\bar{x} - u)'\bar{\mu} &= 0, \\ l \leq \bar{x} \leq u, & \end{aligned} \tag{4}$$

where $\bar{\lambda} \in \mathbb{R}^n$ and $\bar{\mu} \in \mathbb{R}^n$ are the KKT multipliers. Strict complementarity is said to hold at the KKT point $(\bar{x}, \bar{\lambda}, \bar{\mu})$ if $\bar{x}_i = l_i$ implies $\bar{\lambda}_i > 0$ and $\bar{x}_i = u_i$ implies $\bar{\mu}_i > 0$. It is also possible to give second order sufficient conditions of optimality for (PB). The most common is the *KKT second order sufficient condition*; see, e.g., [2]. However, in order to prove a superlinear convergence rate without assuming strict complementarity, we shall employ a stronger condition, known as the *strong second order sufficient condition* (SSOSC). This condition has already been employed, with the same purpose, in [27, 30, 32] (see also [40]).

ASSUMPTION 2 (SSOSC). *Let $(\bar{x}, \bar{\lambda}, \bar{\mu})$ be a KKT triplet for* (PB). *We say that the* SSOSC *holds at $\bar{x}$ if*

$$z'\nabla^2 f(\bar{x})z > 0 \qquad \forall z \in \{z \in \mathbb{R}^n : z_i = 0, \text{ if } \bar{\lambda}_i > 0 \text{ or } \bar{\mu}_i > 0\}.$$

We note that the SSOSC boils down to the KKT second order sufficient condition if the strict complementarity assumption holds. In general, however, Assumption 2 is stronger than the KKT second order sufficient condition in that it requires the positive definiteness of the Hessian of the objective function on a larger region.

3. The local superlinearly convergent algorithm. In this section we define the local algorithm by the iteration

$$x^{k+1} = x^k + d^k, \tag{5}$$

and we show how to build the direction d^k. The calculation of d^k is based on an identification technique of the set of the active constraints and on the solution of KKT-like equations for (PB).

As regards the identification technique, following [20], we define the sets of indices L, U, F of the variables estimated to be active, respectively, at their lower bound, upper bound, or estimated to be free:

$$\begin{aligned} L(x) = L(x;\varsigma) &= \left\{ i : x_i \leq l_i + \min\left[\varsigma c(x) a_i(x) \lambda_i(x), \frac{u_i - l_i}{3} \right] \right\}, \\ U(x) = U(x;\varsigma) &= \left\{ i : x_i \geq u_i - \min\left[\varsigma c(x) b_i(x) \mu_i(x), \frac{u_i - l_i}{3} \right] \right\}, \\ F(x) = F(x;\varsigma) &= \{1, \ldots, n\} \setminus (L(x) \cup U(x)). \end{aligned} \tag{6}$$

Here ς is a positive constant; $a(x)$, $b(x)$, and $c(x)$ are the barrier functions defined by (3); and $\lambda(x), \mu(x)$ are two functions that satisfy, among others things, the following properties: (i) they are continuous; (ii) if $(\bar{x}, \bar{\lambda}, \bar{\mu})$ is a KKT triplet for (PB), then

$\lambda(\bar{x}) = \bar{\lambda}$ and $\mu(\bar{x}) = \bar{\mu}$. Such functions are called "multiplier functions;" see [20] and the references therein. In particular, we use the following multiplier functions that trivially satisfy these features and possess additional properties needed in the definition of the local algorithm:

$$\lambda(x) = [(u-x)^2 + (x-l)^2]^{-1} \circ (x-u)^2 \circ \nabla f(x), \tag{7}$$

$$\mu(x) = -[(u-x)^2 + (x-l)^2]^{-1} \circ (l-x)^2 \circ \nabla f(x). \tag{8}$$

The following theorem shows the main properties of $L(x), U(x), F(x)$ needed in what follows. The validity of this theorem immediately follows from Theorem 2.1 and Remark 2.1 in [20].

THEOREM 3.1. *Let $(\bar{x}, \bar{\lambda}, \bar{\mu})$ be a KKT triplet for* (PB). *Then, there exists a neighborhood Ω of $\bar{x}$ such that, for all $x \in \Omega$,*

$$\begin{array}{rcccl} \{i : \bar{\lambda}_i > 0\} & \subseteq & L(x) & \subseteq & \{i : l_i = \bar{x}_i\}, \\ \{i : \bar{\mu}_i > 0\} & \subseteq & U(x) & \subseteq & \{i : u_i = \bar{x}_i\}, \\ \{i : l_i < \bar{x}_i < u_i\} & \subseteq & F(x) & \subseteq & \{i : \bar{\lambda}_i = 0 \text{ and } \bar{\mu}_i = 0\}. \end{array} \tag{9}$$

Moreover, if the strict complementarity assumption holds, then, for all $x \in \Omega$,

$$L(x) = \{i : l_i = \bar{x}_i\}, \qquad U(x) = \{i : u_i = \bar{x}_i\}, \qquad F(x) = \{i : l_i < \bar{x}_i < u_i\}.$$

We observe that identification techniques have recently been proposed that allow one to identify exactly the active constraints at the solution without requiring strict complementarity [16]. However, the direction that can be obtained by using that partition of the variables does not allow us to obtain a globally convergent algorithm, at least in the present framework.

We are now ready to specify the direction d^k that we use in (5). We obtain d^k as an (approximate) solution of the following linear system:

$$\begin{bmatrix} \nabla^2 f(x^k)_{F^k} \\ E_{L^k} \\ E_{U^k} \end{bmatrix} d = - \begin{bmatrix} \nabla f^k_{F^k} \\ (x^k - l)_{L^k} \\ (x^k - u)_{U^k} \end{bmatrix}, \tag{10}$$

where we denote by L^k, U^k, and F^k the sets $L(x^k), U(x^k), F(x^k)$ evaluated at the current iterate x^k. From (10) we get immediately the components of d^k corresponding to indices in $L^k \cup U^k$:

$$d^k_{L^k} = -(x^k - l)_{L^k}, \tag{11}$$

$$d^k_{U^k} = -(x^k - u)_{U^k}. \tag{12}$$

Roughly speaking, the direction corresponding to indices in $L^k \cup U^k$ is such that $(x^k + d^k)_{L^k \cup U^k}$ "touches" the boundary of the box. By back-substituting (11) and (12) into (10), we see that $d^k_{F^k}$ can be found as the solution of

$$\nabla^2 f^k_{F^k, F^k} d_{F^k} = -\nabla f^k_{F^k} - \nabla^2 f^k_{F^k, L^k \cup U^k} d^k_{L^k \cup U^k}. \tag{13}$$

We can recognize here that the "correction" term introduced in (2) is given by $-\nabla^2 f^k_{F^k, L^k \cup U^k} d^k_{L^k \cup U^k}$.

As we are interested in the solution of large optimization problems, the exact solution at each iteration of (13), when $|F^k|$ is large, may be wasteful, if at all possible.

Then one can use only an *approximate* solution of system (13) and require that this solution become more and more accurate while approaching a KKT point. Indeed, we require that $d^k_{F^k}$ satisfy a system of the type

$$H^k d^k_{F^k} = -g^k + r^k,$$

where $H^k = \nabla^2 f^k_{F^k,F^k}$,

$$g^k = \nabla f^k_{F^k} + \nabla^2 f^k_{F^k,L^k\cup U^k} d^k_{L^k\cup U^k}, \tag{14}$$

and r^k represents a residual in the solution of system (13).

For the approximate computation of $d^k_{F^k}$ we use a conjugate gradient (CG)-type algorithm. The CG scheme NCCGA that we use differs from standard ones, and it is reported in the next section. Here we state only the properties that the direction $d^k_{F^k}$ obtained by NCCGA must satisfy to ensure convergence of the algorithm. In the next section we will show that the CG gradient algorithm NCCGA satisfies these assumptions.

ASSUMPTION 3. *The direction $d^k_{F^k}$ must satisfy the following properties:*

$$(g^k)' d^k_{F^k} \le -\hat{K}_1 \|g^k\|^2, \tag{15}$$

$$\|d^k_{F^k}\| \le \hat{K}_2 \|g^k\|, \tag{16}$$

$\hat{K}_1, \hat{K}_2$ being positive constant.

ASSUMPTION 4. *The direction $d^k_{F^k}$ must satisfy the following property:*

$$\lim_{k\to\infty} \frac{\|H^k d^k_{F^k} + g^k\|}{\|g^k\|} = 0. \tag{17}$$

Now we can prove the main features of the direction d^k obtained by (11), (12), and the application of the CG-type method to system (13).

THEOREM 3.2. *Assume that $d^k_{L^k}$, $d^k_{U^k}$ satisfy* (11), (12)*, and that $d^k_{F^k}$ satisfies Assumption* 3*. Let $\{x^k\}$ be a sequence of points such that $x^k \in \mathcal{S}$. If $\{d^k\} \to 0$, then every accumulation point $\bar{x}$ of $\{x^k\}$, together with $\lambda(\bar{x}), \mu(\bar{x})$, is a KKT point. Moreover, for every x^k belonging to $\mathcal{S}$, d^k is equal to zero if and only if $(x^k, \lambda(x^k), \mu(x^k))$ is a KKT point of* (PB).

Proof. Suppose that $x^k \in \mathcal{S}$ and $\{d^k\} \to 0$. Taking into account that the number of subsets of $\{1, \dots, n\}$ is finite, there exists a subsequence that, without loss of generality, we label again $\{x^k\}$ such that the index sets L^k, U^k, and F^k are constant, and hence we can write:

$$L(x^k) = L, \qquad U(x^k) = U, \qquad F(x^k) = F.$$

From (15) we can write

$$\hat{K}_1 \|g^k\|^2 \le |(g^k)' d^k_{F^k}| \le \|g^k\| \|d^k_{F^k}\|,$$

and using the definition (14) of g^k, we get

$$\hat{K}_1 \|\nabla f^k_{F^k} + \nabla^2 f^k_{F^k,L^k\cup U^k} d^k_{L^k\cup U^k}\| \le \|d^k_{F^k}\|.$$

Taking into account that d^k also satisfies (11), (12) and that x^k stays in a compact set, passing to the limit we get

$$\bar{x}_L = l_L, \qquad \bar{x}_U = u_U, \tag{18}$$

$$\nabla f_F(\bar{x}) = 0. \tag{19}$$

By (7), (8), (19) we have

$$\lambda_F(\bar{x}) = 0, \qquad \mu_F(\bar{x}) = 0.$$

Then, by the definition (6) of the index set F, we have

$$l_F \leq \bar{x}_F \leq u_F, \tag{20}$$

so that, recalling (18), we conclude that $\bar{x}$ is feasible.

By the definition (6) of L and (7), (8), (18), we have

$$\lambda_L(\bar{x}) = \nabla f_L(\bar{x}) \geq 0, \qquad \mu_L(\bar{x}) = 0. \tag{21}$$

Analogously, by the definition (6) of U, (7), (8), and (18), we also have

$$\mu_U(\bar{x}) = -\nabla f_U(\bar{x}) \geq 0, \qquad \lambda_U(\bar{x}) = 0. \tag{22}$$

Now, noting that (18)–(22) imply (4), we have that $\bar{x}$ is a KKT point.

As regards the second statement of the theorem, the *only if* part is a direct consequence of the proof above; hence we have to prove the *if* part. Suppose that x^k together with $\lambda(x^k), \mu(x^k)$ is a KKT point for (PB). Since x^k is feasible, we have, by the definition (6),

$$(x^k - l)_{L^k} = -d_{L^k} = 0,$$
$$(x^k - u)_{U^k} = -d_{U^k} = 0.$$

Furthermore, by the first equation of (4) and again by (9), we have that

$$\nabla f^k_{F^k} = 0.$$

This means, by definition, that $g^k = 0$, and hence from (16) we get $d_{F^k} = 0$. □

The preceding theorem shows that if x^k is not a KKT point, then d^k is not zero, and hence the algorithm (5) is well defined.

We now analyze the local properties of the algorithm. We show that, if convergence towards a point satisfying the SSOSC occurs, then the convergence rate is superlinear.

THEOREM 3.3. *Let $(\bar{x}, \bar{\lambda}, \bar{\mu})$ be a KKT triplet of* (PB) *satisfying the SSOSC, and suppose that the directions $d^k_{L^k}$, $d^k_{U^k}$ satisfy* (11), (12), *and $d^k_{F^k}$ satisfies Assumptions* 3 *and* 4. *Then, there exists a neighborhood Ω of $\bar{x}$ such that if $x^0 \in \Omega$, the sequence $\{x^k\}$ produced by iteration* (5) *is well defined and converges superlinearly to $\bar{x}$.*

Proof. We observe that the direction d^k is the same direction considered in [20] with reference to a local algorithm for the solution of inequality constrained problems of general type. Hence we can obtain the result by applying [20, Theorem 3.1] and classical results of truncated Newton methods [13]. However, to give a better insight into the algorithm, we sketch a direct proof here. The index sets (L^k, F^k, U^k) eventually can belong to only a finite number of triplets of index sets (L^h, F^h, U^h) all satisfying (9), i.e.,

$$\begin{aligned} \{i : \bar{\lambda}_i > 0\} &\subseteq L^h \subseteq \{i : l_i = \bar{x}_i\}, \\ \{i : \bar{\mu}_i > 0\} &\subseteq U^h \subseteq \{i : u_i = \bar{x}_i\}, \\ \{i : l_i < \bar{x}_i < u_i\} &\subseteq F^h \subseteq \{i : \bar{\lambda}_i = 0 \text{ and } \bar{\mu}_i = 0\}. \end{aligned}$$

Hence the direction d^k is a (truncated) Newton direction for one of finitely many systems of the type

$$\begin{bmatrix} \nabla f_{F^h}(x) \\ (x-l)_{L^h} \\ (x-u)_{U^h} \end{bmatrix} = 0$$

all having the solution $\bar{x}$. Under the SSOSC, the Jacobians of these systems

$$\begin{bmatrix} \nabla^2 f_{F^h,F^h}(\bar{x}) & \nabla^2 f_{F^h,L^h\cup U^h}(\bar{x}) \\ 0 & E_{L^h\cup U^h} \end{bmatrix}$$

are all invertible. The desired convergence property then easily follows by standard results of (truncated) Newton methods [13]. □

We note that we can obtain a superlinear convergence rate without assuming strict complementarity and with a very simple iterative scheme. We also remark that we do not impose feasibility of the iteration, and this gives the iteration more freedom.

4. The truncated scheme for d_{F^k}. We consider in this section a CG-type method for the solution of system (13), i.e.,

$$\nabla^2 f^k_{F^k,F^k} d^k_{F^k} = -\nabla f^k_{F^k} - \nabla^2 f^k_{F^k,L^k\cup U^k} d^k_{L^k\cup U^k}.$$

Essentially, the computation of $d^k_{F^k}$ is based on the use of a CG-type algorithm for the minimization of the quadratic model

$$\phi(d) = f^k + (g^k)'d + \frac{1}{2}d'H^k d,$$

where again $H^k = \nabla^2 f^k_{F^k,F^k}$ and $g^k = \nabla f^k_{F^k} + \nabla^2 f^k_{F^k,L^k\cup U^k} d^k_{L^k\cup U^k}$.

Standard CG methods generate sequences $\{p^i\}, \{s^i\}$, where $\{p^i\}$ approximates iteratively the solution $d^k_{F^k}$, and where $\{s^i\}$ are the conjugate directions. The general scheme is

$$\begin{aligned} p^{i+1} &= p^i + \lambda^i s^i, \\ s^{i+1} &= r^{i+1} + \beta^i s^i, \end{aligned}$$

where λ^i minimizes the quadratic function ϕ along the direction s^i, and β^i is chosen so as to maintain conjugacy between the directions s^i. Usually the CG scheme terminates either if the residual $r^i = -\nabla\phi(p^i)$ is in the norm below a prescribed tolerance or if a negative curvature direction $(s^i)'H^k s^i \le 0$ is encountered. The CG scheme that we use is derived from [25] and differs from the more standard CG algorithms outlined above in that if a negative curvature is found, the algorithm does not stop, but continues to generate conjugate directions. In other words, it tries to determine a good approximation of the Newton direction even if H^k is not positive definite. To this aim, the CG algorithm also generates a new sequence $\{d^i\}$ that differs from $\{p^i\}$ only if a negative curvature direction s^i is found. Indeed, in this case the update rule is $d^{i+1} = d^i - \lambda^i s^i$; that is, the opposite direction $-s^i$ is taken. The algorithm then stops either if the residual $\|r^i\|$ is below a given tolerance or if $(s^i)'Hs^i = 0$. The direction $d^k_{F^k}$ is set either to p^i or to d^i, depending on the outcome of a suitable angle condition test.

The truncated scheme is outlined below. In the description of the algorithm, we eliminated the dependencies from the iteration k when this does not produce

confusion. Hence $H = H^k$, $g = g^k$, and $d^k_{F^k} = d_F$. Moreover, $i_{\rm pos}$ and $i_{\rm neg}$ count the number of iterations where directions of positive and negative curvature, respectively, are generated.

Negative curvature conjugate gradient algorithm (NCCGA).
Data: $\eta > 0$, $\sigma \in (0,1)$, $c \in (0,1)$. Define tol $= \eta\|g\|$.

Step **0:** Set $p^0 = 0$, $d^0 = 0$, $r^0 = -g$, $s^0 = r^0$, $i = i_{\rm neg} = i_{\rm pos} = 0$.

Step **1:** If $|(s^i)'Hs^i| \le \sigma\|s^i\|^2$, set $d_F = \begin{cases} -g & \text{if } i = 0 \\ d^i & \text{if } i > 0 \end{cases}$ and stop.

Step **2:** Compute $\lambda^i = \dfrac{(s^i)'r^i}{(s^i)'Hs^i}$,

$$p^{i+1} = p^i + \lambda^i s^i, \qquad r^{i+1} = r^i - \lambda^i H s^i,$$

$$d^{i+1} = \begin{cases} d^i - \lambda^i s^i, & i_{\rm neg} = i_{\rm neg} + 1 \quad \text{if } (s^i)'Hs^i < -\sigma\|s^i\|^2, \\ d^i + \lambda^i s^i & i_{\rm pos} = i_{\rm pos} + 1 \quad \text{if } (s^i)'H^k s^i > \sigma\|s^i\|^2. \end{cases}$$

Step **3:** If $\|r^{i+1}\| >$tol, compute $\beta^i = \dfrac{\|r^{i+1}\|^2}{\|r^i\|^2}$,

$$s^{i+1} = r^{i+1} + \beta^i s^i,$$

set $i = i + 1$, and go to Step 1; otherwise, go to Step 4.

Step **4:** If ($i_{\rm neg} = 0$ or $i_{\rm pos} = 0$), set $d_F = d^{i+1}$ and stop; otherwise, set

$$p = \begin{cases} p^{i+1} & \text{if } g'p^{i+1} \le 0, \\ -p^{i+1} & \text{if } g'p^{i+1} > 0. \end{cases}$$

If $|\, g'p \,| \ge c\, \|g\|^2$, set $d_F = p$; otherwise, set $d_F = d^{i+1}$ and stop. ∎

REMARK 4.1. We recall that Theorem 2.2(c) of [25] ensures that there exist positive constants $\hat{K}_1$ and $\hat{K}_2$ such that the direction $d^k_{F^k}$ produced by Algorithm NCCGA satisfies Assumption 3, i.e.,

$$\begin{aligned} (g^k)'d^k_{F^k} &\le -\hat{K}_1\|g^k\|^2, \\ \|d^k_{F^k}\| &\le \hat{K}_2\|g^k\|. \end{aligned}$$

Moreover, Assumption 4 is satisfied if we take $\eta^k \to 0$ in the stopping criterion $\|r^{i+1}\| \le \eta^k\|g^k\|$ of the NCCGA above.

5. Globalization scheme. In this section we define a *globally and superlinearly convergent* algorithm for problem (PB). In section 3 we have defined a local algorithm. However, far from a solution, we have to tackle the following problems:

(i) the direction d^k may be a "bad" direction;
(ii) global convergence must be enforced by using a stabilization technique;
(iii) superlinear convergence must be retained.

We introduce a general algorithm model for the solution of (PB), which is based on the nonmonotone stabilization technique proposed in [26] and on a merit function $P(x;\varepsilon)$ studied in [18]. Nonmonotone algorithms are known to be very efficient in

forcing global convergence while preserving a fast convergence rate. The use of a penalty function is needed, as explained in the introduction, since we do not force feasibility of the iterates, and thus the objective function f cannot be used to gauge progress towards optimality.

A key point in establishing the properties of the algorithm is the existence of some relation between the search direction and the penalty function.

To be more specific, the algorithm model is an iterative process of the form

$$x^{k+1} = x^k + \rho^k d^k, \tag{23}$$

where d^k is the search direction defined in section 3 and ρ^k is a *stepsize*. The algorithm model that we use includes different strategies for enforcing global convergence without requiring a monotonic reduction of the merit function. This may be reasonable in many situations. For example, if the sequence $\{d^k\}$ goes to zero, then, by Theorem 3.2, the corresponding sequence of points $\{x^k\}$ is converging to a KKT point. Then an effective criterion to control if convergence is taking place is to check whether the norm of the direction is decreasing. In this case the unit step size ($\rho^k = 1$) can be accepted *without computing the merit function.* Otherwise, a check on the merit function value is made, and the algorithm may perform a nonmonotone Armijo-type linesearch procedure [24] with $P(x;\varepsilon)$ as a merit function.

To assist the reader, we split the presentation of the new algorithm into two parts. In the next subsection we briefly recall the expression of the merit function and its main exactness properties, and we prove that a positive scalar γ exists such that the condition

$$\nabla P(x^k;\varepsilon)' d^k \leq -\gamma \|d^k\|^2 \tag{24}$$

holds for sufficiently small values of the penalty parameter. Then, in subsection 5.2 we introduce a general algorithm model for the minimization of $P(x;\varepsilon)$, which uses the direction d^k, and we show that this algorithm is globally convergent to KKT points of (PB).

5.1. The merit function for (PB). The merit function P is a particular case of the class of continuously differentiable exact penalty functions introduced in [18]. We report here only some of its basic features; the interested reader is referred to [18] for a more complete discussion. The penalty function makes use of the multiplier functions $\lambda(x)$ and $\mu(x)$ defined in (7) and (8), and it is defined as follows:

$$P(x;\varepsilon) = f(x) + \lambda(x)' r(x;\varepsilon) + \mu(x)' s(x;\varepsilon) + \frac{1}{\varepsilon c(x)} \left[(r(x;\varepsilon)^2)' a^{-1}(x) + (s(x;\varepsilon)^2)' b^{-1}(x) \right],$$

with

$$\begin{aligned} r(x;\varepsilon) &= \max\left[l - x, -\frac{\varepsilon}{2} c(x) a(x) \circ \lambda(x) \right], \\ s(x;\varepsilon) &= \max\left[x - u, -\frac{\varepsilon}{2} c(x) b(x) \circ \mu(x) \right], \end{aligned} \tag{25}$$

where $a(x)$, $b(x)$, and $c(x)$ are the shifted barrier functions defined by (3), and x_b is the point defined in section 2.

The penalty function depends on a positive parameter ε; furthermore, it is defined only in the open set $\mathcal{S}$ defined in section 2. It can be proved that the level sets of the penalty function $\{x \in \mathcal{S} : P(x;\varepsilon) \leq P(x_a;\varepsilon)\}$ are compact [18]. In particular,

this implies that a minimization algorithm, starting from a point $x^0 \in \mathcal{S}$ such that $P(x^0;\varepsilon) \le P(x_a;\varepsilon)$, applied to the penalty function will never generate unbounded sequences. A detailed study of the properties of $P(x;\varepsilon)$ can be found in [18]. It can be proved that, *for sufficiently small values of the penalty parameter* $\varepsilon > 0$, there is a one-to-one correspondence between (unconstrained) stationary and minimum points of the penalty function on $\mathcal{S}$ and KKT and minimum points of (PB) with a function value smaller than $f(x_b)$.

The properties of the penalty function outlined above clearly show that we can solve (PB) by performing a *single unconstrained* minimization of $P(x;\varepsilon)$, provided that ε is small enough. From this point of view, another important feature of the penalty function $P(x;\varepsilon)$ is that, in spite of the terms (25), it is continuously differentiable in $\mathcal{S}$ (see [18]), so that standard efficient methods for unconstrained smooth minimization can be employed. However, if one wants to develop a practical algorithm, at least two important questions have to be answered: how to calculate a suitable value of ε so that, as discussed above, the unconstrained minimization of the penalty function is equivalent to the solution of (PB), and which unconstrained optimization algorithm to employ for the minimization of the penalty function. In [18] a very general scheme for updating ε has been proposed that, coupled with practically any standard unconstrained minimization algorithm, allows us to solve (PB). This scheme has, however, a drawback in that it does not exploit the structure either of (PB) or of the minimization algorithm employed. The algorithm that we present avoids in a novel and innovative way the two drawbacks mentioned above. In particular, the unconstrained minimization algorithm we use is a nonmonotone linesearch scheme which uses the search direction defined in section 3. Since this algorithm is so tailored to the structure of the problem, we can use a rule for updating the penalty parameter different from that proposed in [18] and which, for the problem at hand, seems much more efficient from a practical point of view.

As a first step in the definition of the algorithm, we have to show the relationship between the direction d^k defined in section 3 and the merit function $P(x;\varepsilon)$.

To this end we define the constant ς that appears in (6) as $\varsigma = \varepsilon/2$; that is, we have

$$L(x) = L(x;\varepsilon) := \left\{ i : x_i \le l_i + \min\left[\frac{\varepsilon c(x)}{2} a_i(x)\lambda_i(x), \frac{u_i - l_i}{3} \right] \right\}, \tag{26}$$

$$U(x) = U(x;\varepsilon) := \left\{ i : x_i \ge u_i - \min\left[\frac{\varepsilon c(x)}{2} b_i(x)\mu_i(x), \frac{u_i - l_i}{3} \right] \right\}, \tag{27}$$

and $F(x) = F(x;\varepsilon) = \{1,\dots,n\}\backslash(L(x) \cup U(x))$, where ε is the penalty parameter.

We prove that the direction d^k obtained by (11), (12), and (13), solved by algorithm NCCGA with L^k, U^k, F^k given by (26) and (27), satisfies the descent condition (24) for sufficiently small values of the penalty parameter ε.

THEOREM 5.1. *There exists an* $\bar{\varepsilon} > 0$ *such that for every* $\varepsilon \in (0, \bar{\varepsilon}]$ *and* $x^k \in \{x \in \mathcal{S} : P(x;\varepsilon) \le P(x_a;\varepsilon)\}$ *the following relation holds:*

$$\nabla P(x^k;\varepsilon)' d^k \le -\gamma \|d^k\|^2 \tag{28}$$

for some positive γ, *where* d^k *is obtained by* (11), (12), *and* (13) *solved by NCCGA, with* L^k, U^k, F^k *given by* (26) *and* (27).

Proof. First we report the expression of the gradient[1] of $P(x;\varepsilon)$ (see [18]):

$$\begin{aligned}\nabla P(x;\varepsilon) =& -\frac{1}{\varepsilon c(x)} a^{-1}(x) \circ \left[2e + r(x,\varepsilon) \circ a^{-1}(x)\right] \circ r(x;\varepsilon) + \nabla\lambda(x) r(x;\varepsilon) \\ &+ \frac{1}{\varepsilon c(x)} b^{-1}(x) \circ \left[2e + s(x,\varepsilon) \circ b^{-1}(x)\right] \circ s(x;\varepsilon) + \nabla\mu(x) s(x;\varepsilon) \\ &+ \frac{1}{\varepsilon c(x)^2} \left[(r(x;\varepsilon)^2)' a^{-1}(x) + (s(x;\varepsilon)^2)' b^{-1}(x)\right] \nabla f(x),\end{aligned} \tag{29}$$

where $\nabla\lambda, \nabla\mu$ are the gradients of the twice continuously differentiable multiplier functions and $e \in R^n$ is a vector of all ones.

The proof is by contradiction. Assume that the theorem is false; then there exist sequences $\{x^k\}$, $\{\varepsilon^k\}$, and $\{\gamma^k\}$ such that

$$\begin{aligned}&\varepsilon^k \downarrow 0, \quad \gamma^k \downarrow 0, \quad x^k \in \{x \in \mathcal{S} : P(x;\varepsilon^k) \le P(x_a;\varepsilon^k)\}, \\ &\nabla P(x^k;\varepsilon^k)' d^k > -\gamma^k \|d^k\|^2.\end{aligned} \tag{30}$$

Furthermore, we shall assume, without loss of generality, that the index sets L^k, F^k, and U^k are constant, so that we can write

$$L(x^k) = L, \qquad F(x^k) = F, \qquad U(x^k) = U.$$

By (29) we can write

$$\begin{aligned}(\nabla P^k)' d^k =& -\frac{1}{\varepsilon^k c^k} (r^k)' \left[2e + (a^k)^{-1} \circ r^k\right] \circ (a^k)^{-1} \circ d^k + (r^k)' (\nabla\lambda^k)' d^k \\ &+ \frac{1}{\varepsilon^k c^k} (s^k)' \left[2e + (b^k)^{-1} \circ s^k\right] \circ (b^k)^{-1} \circ d^k + (s^k)' (\nabla\mu^k)' d^k \\ &+ \frac{1}{\varepsilon^k (c^k)^2} \left[(r^k)' \left((a^k)^{-1} \circ r^k\right) + (s^k)' \left((b^k)^{-1} \circ s^k\right)\right] (\nabla f^k)' d^k.\end{aligned} \tag{31}$$

Recalling that d^k satisfies (11), (12), the definitions (25), (26), (27), and that $\mathcal{S}$ is compact by Assumption 1, we have, for ε^k small enough,

$$\begin{aligned}&i \in L \implies r_i^k = (l - x^k)_i = d_i^k, && s_i^k = -\frac{\varepsilon^k c^k}{2} b_i^k \mu_i^k, \\ &i \in F \implies r_i^k = -\frac{\varepsilon^k c^k}{2} a_i^k \lambda_i^k && s_i^k = -\frac{\varepsilon^k c^k}{2} b_i^k \mu_i^k, \\ &i \in U \implies r_i^k = -\frac{\varepsilon^k c^k}{2} a_i^k \lambda_i^k, && s_i^k = (x^k - u)_i = -d_i^k;\end{aligned} \tag{32}$$

hence, rearranging terms, (31) becomes

$$\begin{aligned}(\nabla P^k)' d^k =& -\frac{1}{\varepsilon^k c^k} (d_L^k)' [2e + (l - x^k) \circ (a^k)^{-1}]_L \circ ((a^k)^{-1} \circ d^k)_L \\ &- \frac{1}{\varepsilon^k c^k} (d_U^k)' [2e + (x^k - u) \circ (b^k)^{-1}]_U \circ ((b^k)^{-1} \circ d^k)_U \\ &+ (\lambda_F^k - \mu_F^k)' d_F^k + \frac{\varepsilon^k c^k}{4} (\lambda_F^k)' (\lambda_F^k \circ d_F^k) - \frac{\varepsilon^k c^k}{4} (\mu_F^k)' (\mu_F^k \circ d_F^k)\end{aligned}$$

[1] We remark that the terms of the gradient have been rearranged using the expression of the multiplier functions λ, μ, so that the expression of $\nabla f(x)$ does not appear explicitly in the above formula.

$$
\begin{aligned}
&+\frac{1}{2}(\lambda_U^k)'\left[2e-\frac{\varepsilon^k c^k}{2}\lambda^k\right]_U\circ d_U^k-\frac{1}{2}(\mu_L^k)'\left[2e-\frac{\varepsilon^k c^k}{2}\mu^k\right]_L\circ d_L^k\\
&+(d_L^k)'[(\nabla\lambda^k)'d^k]_L-(d_U^k)'[(\nabla\mu^k)'d^k]_U\\
&-\frac{\varepsilon^k c^k}{2}(a^k\circ\lambda^k)_F'[(\nabla\lambda^k)'d^k]_F-\frac{\varepsilon^k c^k}{2}(a^k\circ\lambda^k)_U'[(\nabla\lambda^k)'d^k]_U\\
&-\frac{\varepsilon^k c^k}{2}(b^k\circ\mu^k)_L'[(\nabla\mu^k)'d^k]_L-\frac{\varepsilon^k c^k}{2}(b^k\circ\mu^k)_F'[(\nabla\mu^k)'d^k]_F\\
&+\frac{1}{\varepsilon^k(c^k)^2}\left[((r^k)^2)'(a^k)^{-1}+((s^k)^2)'(b^k)^{-1}\right](\nabla f^k)'d^k.
\end{aligned}
\tag{33}
$$

We now make the following readily verifiable observations.

(i) Each element of the vectors $[2e+(l-x^k)\circ(a^k)^{-1}]_L$ and $[2e+(x^k-u)\circ(b^k)^{-1}]_U$ is greater than 1.

(ii) Recalling the definition (7) and (8) of the multiplier functions λ and μ, we have the following:

(1)

$$(\lambda_F^k-\mu_F^k)'d_F^k=(\nabla f_F^k)'d_F^k.$$

By using conditions (15) and (16) and the definition (14) we can write

$$(\nabla f_{F^k}^k)'d_{F^k}^k\le-\frac{\hat K_1}{\hat K_2^2}\|d_{F^k}^k\|^2-(d_{F^k}^k)'\nabla^2 f_{F^k,L^k\cup U^k}^k d_{L^k\cup U^k}^k,$$

and, since $\|\nabla^2 f^k\|$ is bounded, we have

$$(\nabla f_{F^k}^k)'d_{F^k}^k\le-K_1\|d_{F^k}^k\|^2+K_2\big(\|d_{F^k}^k\|\|d_{L^k}^k\|+\|d_{F^k}^k\|\|d_{U^k}^k\|\big).\tag{34}$$

Hence

$$(\lambda_F^k-\mu_F^k)'d_F^k\le-K_1\|d_F^k\|^2+K_2\big(\|d_F^k\|\|d_L^k\|+\|d_F^k\|\|d_U^k\|\big)$$

for some positive constants K_1, K_2.

(2) By using (15) we have $\hat K_1\|g^k\|^2\le|(g^k)'d_{F^k}^k|\le\|g^k\|\|d_{F^k}^k\|$, and from the definitions (7), (8), and (14) we get

$$\begin{aligned}
\|\lambda_F^k\|&\le\|\nabla f_F^k\|\le K_3\big(\|d_F^k\|+\|d_U^k\|+\|d_L^k\|\big),\\
\|\mu_F^k\|&\le\|\nabla f_F^k\|\le K_3\big(\|d_F^k\|+\|d_U^k\|+\|d_L^k\|\big).
\end{aligned}$$

(3) By (7), (8), and (32),

$$\begin{aligned}
(\lambda^k)_i&=\frac{(d^k)_i^2}{(x^k-u)_i^2+(l-x^k)_i^2}\nabla f(x^k)_i,\qquad i\in U,\\
(\mu^k)_i&=-\frac{(d^k)_i^2}{(x^k-u)_i^2+(l-x^k)_i^2}\nabla f(x^k)_i,\qquad i\in L.
\end{aligned}$$

(iii) The quantities $\|x^k-l\|$, $\|x^k-u\|$, $\|\lambda(x^k)\|$, $\|\mu(x^k)\|$, $\|\nabla\lambda(x^k)\|$, $\|\nabla\mu(x^k)\|$, and $\|\nabla f(x^k)\|$ are bounded.

Then, taking into account (33) and points (i)–(iii) above, we can assert that, for ε^k small enough,

$$
\begin{aligned}
(\nabla P^k)'d^k \;\le\; & -\frac{K_1}{\varepsilon^k c^k}\|d_L^k\|^2 - K_2\|d_F^k\|^2 - \frac{K_3}{\varepsilon^k c^k}\|d_U^k\|^2 + K_4\|d_L^k\|^2 + K_5\|d_U^k\|^2 \\
& + K_6\|d_L^k\|\|d_F^k\| + K_7\|d_L^k\|\|d_U^k\| + K_8\|d_F^k\|\|d_U^k\| \\
& + \frac{1}{\varepsilon^k (c^k)^2}[(d_L^k)'((a^k)^{-1}\circ d_L^k) + (d_U^k)'((b^k)^{-1}\circ d_U^k)](\nabla f^k)'d^k \\
& + \varepsilon^k K_9\|d^k\|^2,
\end{aligned} \tag{35}
$$

where $K_1,\ldots,K_9$ are positive constants. Equations (30) and (35) imply that for k sufficiently large we can write

$$
\begin{aligned}
0 \le\; & \gamma^k\|d^k\|^2 - \frac{K_1}{\varepsilon^k c^k}\|d_L^k\|^2 - K_2\|d_F^k\|^2 - \frac{K_3}{\varepsilon^k c^k}\|d_U^k\|^2 + K_4\|d_L^k\|^2 \\
& + K_5\|d_U^k\|^2 + K_6\|d_L^k\|\|d_F^k\| + K_7\|d_L^k\|\|d_U^k\| + K_8\|d_F^k\|\|d_U^k\| \\
& + \frac{1}{\varepsilon^k (c^k)^2}[(d_L^k)'((a^k)^{-1}\circ d_L^k) + (d_U^k)'((b^k)^{-1}\circ d_U^k)](\nabla f^k)'d^k + \varepsilon^k K_9\|d^k\|^2 \\
\le\; & -(\|d_L^k\|,\|d_F^k\|,\|d_U^k\|)Q^k(\|d_L^k\|,\|d_F^k\|,\|d_U^k\|)' \\
& + \underbrace{\frac{1}{\varepsilon^k (c^k)^2}[(d_L^k)'((a^k)^{-1}\circ d_L^k) + (d_U^k)'((b^k)^{-1}\circ d_U^k)](\nabla f^k)'d^k}_{*},
\end{aligned} \tag{36}
$$

where Q^k is the symmetric matrix defined by

$$
Q^k = \begin{pmatrix}
\frac{K_1}{\varepsilon^k c^k} - K_4 - \gamma^k - \varepsilon^k K_9 & -\frac{K_6}{2} & -\frac{K_7}{2} \\
-\frac{K_6}{2} & K_2 - \gamma^k - \varepsilon^k K_9 & -\frac{K_8}{2} \\
-\frac{K_7}{2} & -\frac{K_8}{2} & \frac{K_3}{\varepsilon^k c^k} - K_5 - \gamma^k - \varepsilon^k K_9
\end{pmatrix}.
$$

For ε^k and γ^k small enough (in particular, $\gamma^k = K_2/2$ works for all small ε^k), it is easily seen that Q^k is a positive definite matrix with eigenvalues uniformly bounded away from zero. We now note that we can assume, without loss of generality, that the bounded quantity $\|d^k\|$ admits a limit and that, by Proposition 3.4 in [18], $x^k \to \bar{x} \in \mathcal{F}\cap\mathcal{S}$. By (9) we have that $\|d_L^k\| \to 0$ and $\|d_U^k\| \to 0$. We consider now two different cases: (a) d^k converges to 0, (b) d^k converges to a vector different from 0.

(a) In this case, by Theorem 3.2, $\bar{x}$ is a KKT point. We show that $c^k \not\to 0$. First we note that for all k we have $P(x^k;\varepsilon^k) \le P(x_a;\varepsilon^k) \le f(x_a)$, where the last inequality holds because x_a is feasible. Moreover, we have also $P(x^k;\varepsilon^k) \ge f(x^k) + \lambda(x^k)'r(x^k;\varepsilon^k) + \mu(x^k)'s(x^k;\varepsilon^k)$. Now, assume by contradiction that $c^k \to 0$ (i.e., $f(\bar{x}) = f(x_b)$). Then, recalling the expression of P, we must have that $s(x^k;\varepsilon^k), r(x^k;\varepsilon^k) \to 0$. Hence we have $f(x_b) = f(\bar{x}) \le \liminf_{k\to\infty} P(x^k;\varepsilon^k) \le f(x_a)$. This leads to $f(x_b) \le f(x_a)$, which is a contradiction with the assumption that $f(x_a) < f(x_b)$. Hence the expression (*) in (36) is dominated by the quadratic term $(\|d_L^k\|,\|d_F^k\|,\|d_U^k\|)Q^k(\|d_L^k\|,\|d_F^k\|,\|d_U^k\|)'$, so that we have a contradiction from (36).

(b) In this case, as we already observed, $d_L^k \to 0$ and $d_U^k \to 0$, so that we must have $d_F^k \to \tilde{d}_F \ne 0$. Since we can write $(\nabla f^k)'d^k = (\nabla f_L^k)'d_L^k + (\nabla f_F^k)'d_F^k + (\nabla f_U^k)'d_U^k$,

then, using (34), we have that

$$\lim_{k\to\infty}(\nabla f^k)'d^k = \lim_{k\to\infty}(\nabla f_F^k)'d_F^k \le -K_1\|\tilde{d}_F\|^2 < 0,$$

which leads, for k large enough, to $(\nabla f^k)'d^k < 0$, so that the term (*) is nonpositive. Since the quadratic term in (36) tends to a negative quantity, again we have a contradiction from (36) and the proof is complete. □

Theorem 5.1 states that the direction d^k defined in section 3 satisfies suitable "descent" conditions with respect to the merit function P if ε is smaller than a threshold value $\bar{\varepsilon}$. However the value $\bar{\varepsilon}$ generally is not known in advance and therefore has to be determined during the minimization process.

5.2. A nonmonotone algorithm for the minimization of $P(x;\varepsilon)$. In this section we describe the NMSB algorithm for the solution of problem (PB). The algorithm is basically a nonmonotone line search algorithm [26] for the unconstrained minimization of the function $P(x;\varepsilon)$, coupled with a simple updating scheme of the parameter ε. We show that every limit point of the sequence produced by the algorithm is a KKT point of (PB) and that if one of the limit points satisfies the SSOSC, then the whole sequence converges with a superlinear convergence rate.

In order to help the reader understand the different roles played by the merit function and by the nonmonotone scheme, we first give a brief description of a simple *monotone* version of the algorithm. Everywhere in this section we assume that the direction d^k is obtained by (11), (12), and (13) solved by NCCGA. The solution of the linear system by NCCGA represents the major computational burden at each iteration in the algorithm, but usually requires a limited number of CG inner iterations.

In the monotone version, starting from a point $x^0 \in \mathcal{S}$ and such that $P(x^0;\varepsilon) \le P(x_a;\varepsilon)$, the algorithm generates the point $x^{k+1} = x^k + \rho^k d^k$, where ρ^k is obtained by means of a *monotone linesearch* described below. The algorithm stops if a KKT point is found (or, equivalently, if $\|d^k\| = 0$).

monotone linesearch: Given $\zeta \in (0,1)$ and $\sigma \in (0,1/2)$,

If $\nabla P(x^k;\varepsilon)'d^k \le -\zeta\varepsilon\|d^k\|^2$, then

find the smallest integer from $i = 0, 1, \ldots$ such that

$x^k + 2^{-i}d^k \in \mathcal{S}$,

$P(x^k + 2^{-i}d^k;\varepsilon) \le P(x^k;\epsilon) + \sigma 2^{-i}\nabla P(x^k;\varepsilon)'d^k$,

set $x^{k+1} = x^k + 2^{-i}d^k$;

Otherwise (**update** ε)

set $\varepsilon = 0.5\varepsilon$ and $x^0 = \begin{cases} x^k & \text{if } P(x^k,\varepsilon) \le P(x_a;\varepsilon), \\ x_a & \text{otherwise,} \end{cases}$

set $k = 0$ and restart the monotone algorithm.

Endif

Basically, the *linesearch* procedure is divided into two main parts. The first consists of checking whether condition (28) of Theorem 5.1 holds. If not, the value of ε is reduced. Theorem 5.1 guarantees that after a finite number of reductions the value of ε settles down and condition (28) is always satisfied. Note also that when a reduction of ε takes place, we can restart the minimization process with x^k or x_a, depending on which of the two gives a better value of the penalty function. If the current value of ε appears to be sufficiently small, a stepsize ρ^k is found such

that the new point $x^{k+1} = x^k + \rho^k d^k$ is in the set $\mathcal{S}$ where the function $P(x;\varepsilon)$ is defined, and an Armijo condition is satisfied. We note that, thanks to the form of the constraints, it is possible to find analytically the value of the step such that $l - \alpha < x^{k+1} < u + \beta$, and any further reduction of ρ^k maintains the feasibility of the new point. The only part that must be performed iteratively is finding the value of ρ^k such that $f(x^{k+1}) < f(x_b)$. This procedure is very simple and is similar to analogous procedures in interior point methods.

The monotone scheme enforces the reduction of the merit function at every step and performs a linesearch to determine an appropriate stepsize. This traditional approach can be improved by allowing the merit function value to increase in a controlled manner and by using a criterion based on the size of the search direction to assess the acceptability of the stepsize of one. These more refined strategies, which do not even require the evaluation of the merit function at each step, are based on [24, 26, 33, 34], and there is currently wide numerical experience indicating that they are often beneficial from the computational point of view. This is even more true in the case of the minimization of penalty functions, which can easily have narrow curved valleys.

We now pass to the description of this nonmonotone version of the algorithm. The changes with respect to the latter algorithm, although extensive, concern only the criteria for the choice of the stepsize ρ^k. Since in this version we do not necessarily evaluate the merit function at each iteration, we introduce the new counters j, which is increased every time we evaluate the merit function, and $\ell(j)$, which denotes the iteration at which the merit function has been evaluated.

Nonmonotone stabilization algorithm for box constrained problems (NMSB)

Data: Choose $\varepsilon > 0$, x_a, $x_b \in \mathcal{F}$ s.t. $f(x_a) < f(x_b)$,
$\alpha, \beta \in R^n$ with $\alpha, \beta > 0$,
$\delta_0 \geq 0$, $\theta \in (0,1)$, an integer $\mathcal{N} \geq 0$.

Initialization: Set $x^0 = x_a$, $k = 0$, $j = 0$, $\ell(j) = 0$, and $\delta = \delta_0$.
Compute $P(x^0;\varepsilon)$ and set $\mathcal{R}^j = P(x^0;\varepsilon)$.

Iteration: If $(\|d^k\| = 0)$, return x^k and stop.
If $k \neq \ell(j) + \mathcal{N}$,
find x^{k+1} by a **standard-step**;
Otherwise,
find x^{k+1} by a **function-step**.
Endif
Set $k = k + 1$, and repeat **Iteration**.

The algorithm performs two different kind of steps: *standard*-steps and *function*-steps. Standard-steps usually account for the majority of the steps; more precisely, we always perform standard steps except every $\mathcal{N}$ iterations, where $\mathcal{N}$ is a nonnegative integer chosen by the user (with usual values between 5 and 20).

Standard-step: If $\|d^k\| \leq \delta$,
Find a ρ^k such that $x^k + \rho^k d^k \in \mathcal{S}$;
set $x^{k+1} = x^k + \rho^k d^k$, $\delta = \theta\delta$.
Otherwise,
find x^{k+1} by a **function-step**.
End if

As we discussed briefly at the beginning of section 5, the main criterion of the

algorithm for assessing progress towards optimality is the magnitude of the search direction d^k. In a standard-step we therefore check whether $\|d^k\|$ is smaller than a quantity that we drive to zero while the algorithm progresses. If this simple test is passed, the algorithm does not even compute the merit function, and only a control on the fact that the new iterate x^{k+1} belongs to $\mathcal{S}$ is performed. If, instead, the direction is not sufficiently small, the merit function is computed and a nonmonotone Armijo-type linesearch procedure is performed, proceeding essentially as in a function-step described below.

Function-step: Compute $P(x^k;\varepsilon)$.
If $P(x^k;\varepsilon) \geq \mathcal{R}^j$,
backtrack: replace x_k by $x_{\ell(j)}$, set $k = \ell(j)$,
find x^{k+1} by a **nonmonotone linesearch**;
Otherwise,
update $\mathcal{R}^j$, set $\ell(j) = k$ and $j = j + 1$.
If $\|d^k\| \leq \delta$, then
find a ρ^k such that $x^k + \rho^k d^k \in \mathcal{S}$;
set $x^{k+1} = x^k + \rho^k d^k$, $\delta = \theta\delta$.
Otherwise,
find x^{k+1} by a **nonmonotone linesearch**.
Endif
Endif

Function-steps occur at least every $\mathcal{N}$ iterations and should be regarded as a safeguard. In these steps the merit function is always computed, and its value is compared with the reference value $\mathcal{R}^j$. If the value of the merit function is smaller than the reference value, the algorithm proceeds as in a standard-step; i.e., the stepsize of one is accepted if the direction is sufficiently small, and otherwise a linesearch is performed. Otherwise, if the value of the merit function is larger than $\mathcal{R}^j$, that is, if it is "too large," the algorithm "backtracks" by restoring the vector of variables to the last point at which the objective function was smaller than the reference value $\mathcal{R}^j$.

The *nonmonotone linesearch* procedure used in NMSB differs from the monotone one considered before only in the fact that when performing the Armijo line search we do not enforce a decrease with respect to the current value of the merit function $P(x^k;\varepsilon)$, but rather with respect to a *reference value* $\mathcal{R}^j$ that can be larger than $P(x^k;\varepsilon)$.

nonmonotone linesearch: Given $\zeta \in (0,1)$ and $\sigma \in (0,1/2)$,

If $\nabla P(x^k;\varepsilon)' d^k \leq -\zeta\varepsilon\|d^k\|^2$, then
find the smallest integer from $i = 0, 1, \ldots$ such that
$x^k + 2^{-i} d^k \in \mathcal{S}$,
$P(x^k + 2^{-i} d^k;\varepsilon) \leq \mathcal{R}^j + \sigma 2^{-i} \nabla P(x^k;\varepsilon)' d^k$,
set $x^{k+1} = x^k + 2^{-i} d^k$, $\ell(j) = k + 1$ and **update** $\mathcal{R}^j$;
otherwise (**update** ε),
set $\varepsilon = 0.5\varepsilon$ and $x^0 = \begin{cases} x^k & \text{if } P(x^k,\varepsilon) \leq P(x_a;\varepsilon), \\ x_a & \text{otherwise,} \end{cases}$
set $k = 0$ and restart the NMSB algorithm.

Note that the reference value $\mathcal{R}^j$ is updated only when the merit function value is calculated. To complete the description of the algorithm we have to specify only the updating rule for the reference value $\mathcal{R}^j$. To this end we recall that the index j is increased every time we set $\ell(j) = k$, i.e., every time the penalty function is evaluated. Therefore $\{x^{\ell(j)}\}$ is the sequence of points at which the merit function is evaluated, and $\{\mathcal{R}^j\}$ is the sequence of reference values. The reference value is initially set to $P(x^0;\varepsilon)$. Whenever a point $x^{\ell(j)}$ is generated such that $P(x^{\ell(j)},\varepsilon) < \mathcal{R}^j$, the reference value is updated by taking into account a fixed number $m(j) \le \bar{m}$ of previous values of the penalty function. To be precise, the updating rule for $\mathcal{R}^j$ is the following.

Update $\mathcal{R}^j$: Given $\bar{m} \ge 0$, set $m(0) = 0$ and let $m(j+1)$ be such that

$$m(j+1) \le \min[m(j)+1, \bar{m}].$$

Choose the value $\mathcal{R}^{j+1}$ to satisfy

$$P(x^{\ell(j+1)};\varepsilon) \le \mathcal{R}^{j+1} \le \max_{0\le i\le m(j+1)} P(x^{\ell(j+1-i)};\varepsilon). \tag{37}$$

The NMSB algorithm is a very general scheme and encompasses many possible extensions of unconstrained algorithms.

For example, if we set $\bar{m} = 0$, $\delta_0 = 0$, and $\mathcal{N} = 0$, we obtain the Armijo monotone stabilization algorithm described at the beginning. If we set $\bar{m} > 0$, $\delta_0 = 0$, and $\mathcal{N} = 0$, we obtain the box constrained version of the nonmonotone algorithm proposed in [24].

The following result holds.

THEOREM 5.2. *Suppose that Assumption* 1 *holds. Let $\{x^k\}$ be the sequence generated according to the NMSB algorithm described above. Then*

(i) *after a finite number of iterations the penalty parameter ε stays fixed;*
(ii) *there exists at least one limit point of the sequence $\{x^k\}$;*
(iii) *every limit point of the sequence $\{x^k\}$ is a KKT of* (PB)*;*
(iv) *every limit point $\bar{x}$ of the sequence $\{x^k\}$ is such that $f(\bar{x}) \le f(x_a)$.*

Point (i) follows from the test $\nabla P(x^k;\varepsilon)'d^k \le -\zeta\varepsilon\|d^k\|^2$ performed during the linesearch in the NMSB algorithm. In fact, Theorem 5.1 guarantees that after a finite number of reductions the value of ε settles down and condition (28) is always satisfied. The proof of points (ii)–(iv), albeit conceptually very similar to that in [26], is tedious and not particularly illuminating. We refer the reader to [21] for a detailed proof of the theorem.

In the statement of Theorem 5.2 we have stressed the properties of the algorithm in terms of the properties of problem (PB). However, we can equivalently see algorithm NMSB as an algorithm for the minimization of the penalty function P. From this point of view, we can also see that every accumulation point of the sequence generated by the algorithm is a stationary point of the penalty function. This easily follows by the fact that every KKT point of (PB) is a stationary point of $P(x;\varepsilon)$ for every positive value of ε; see [18].

We now pass to analyzing the local properties of the algorithm.

THEOREM 5.3. *Suppose that the sequence $\{x^k\}$ produced by the algorithm converges to a point $\bar{x}$ satisfying the SSOSC. Then, eventually $x^{k+1} = x^k + d^k$ (i.e., the stepsize of one is accepted eventually), and the convergence rate is superlinear.*

Proof. We first observe that the gradient of P is semismooth according to the definition of [35, 39]. This follows easily by the expression (29) and the facts that the

composite of semismooth functions is semismooth, that the *max* operator is semismooth, and that smooth functions are also semismooth [35]. Now, taking into account that $\mathcal{S}$ is compact by Assumption 1 and that (11), (12), (16) hold, we have that the direction d^k is bounded. Moreover, from Theorem 5.2, ε settles after a finite number of iterations. Since $x^k \to \bar{x}$, a KKT point, this implies that $\nabla P(x^k;\varepsilon) \to \nabla P(\bar{x};\varepsilon) = 0$, and hence from (28) we have that $d^k \to 0$. Now, since $f(\bar{x}) < f(x_b)$, then eventually $x^k + d^k \in \mathcal{S}$, and the Armijo rule eventually accepts stepsize one by [15, Theorem 3.2]. Now superlinear convergence follows from Theorem 3.3. ∎

REMARK 5.1. It may be interesting to note that at the beginning of the paper we made, for simplicity, the blanket assumption that f is three times continuously differentiable. However, it is possible to show that to establish global convergence it is sufficient to assume continuous differentiability of the objective function, while to prove the superlinear convergence rate of the algorithm it is enough to assume that the Hessian of f is semismooth. Furthermore, these differentiability assumptions are only needed on $\mathcal{S}$.

REMARK 5.2. In Theorem 5.3, for simplicity, we made the assumption that $\{x^k\} \to \bar{x}$. We remark, however, that it is standard to prove that if one of the limit points of the sequence $\{x^k\}$ generated by the algorithm satisfies the strong second order sufficient condition, then the whole sequence converges to this point.

6. Numerical experiments. In this section we analyze the behavior of an implementation of NMSB algorithm. The following choices were made.

- Given a user-supplied starting point x_u, we generate the two points x_a, x_b required by the algorithm in the following simple way. First, we project x_u onto the feasible region, thus obtaining a point $\tilde{x}$. Then we generate a new feasible point $\hat{x}$ by moving from $\tilde{x}$ along the gradient of the objective function with a prefixed stepsize

$$\delta^0 = \min\left(\frac{2\max(|f(\tilde{x})|, 10^{-3})}{\max(10^{-20}, \|\nabla f(\tilde{x})\|^2)}, 1\right)$$

  and by then projecting the resulting point onto the feasible region, namely,

$$\hat{x} = \operatorname{mid}\left[l, \tilde{x} - \delta^0 \nabla f(\tilde{x}), u\right],$$

  where mid is the componentwise median of the three vectors in the arguments. Finally, if $f(\tilde{x}) < f(\hat{x})$, we set $x_a = \tilde{x}$ and $x_b = \hat{x}$; otherwise, we set $x_a = \hat{x}$ and $x_b = \tilde{x}$.
- The starting point of the minimization process is $x^0 = x_a$.
- The barrier parameters α and β are chosen according to the rule given below:

$$\alpha_i = \max[0, l_i - (x_u)_i] + \operatorname{mid}[0.1, u_i - l_i, 10],$$
$$\beta_i = \max[0, (x_u)_i - u_i] + \operatorname{mid}[0.1, u_i - l_i, 10].$$

  The mid term in the definition of α and β makes the wideness of $\mathcal{S}$ proportional in a safeguarded way to the wideness of the box. The first term instead ensures that the user-supplied point x_u belongs to $\mathcal{S}$.
- The initial value of the penalty parameter is given by

$$\varepsilon_0 = 10^{-3} \min\left[10^{-2}, \frac{1}{f(x_b)}, \frac{1}{\|\nabla f(x_a)\|^2}\right],$$

  where if one of the denominators is zero, we understand that the corresponding term is $+\infty$.

- We set the remaining parameters in NMSB to the following values:

$$\mathcal{N} = 20, \quad \bar{m} = 20, \quad \delta_0 = 10^3, \quad \sigma = 0.5, \quad \theta = 0.5, \quad \zeta = 10^{-3}.$$

We note that in our experience the choice of the barrier parameters α, β and of the initial penalty parameter ε is not crucial. In particular, the updating rule for the penalty parameter allows one to easily recover from an unsuitable initial choice of ε^0.

Regarding the stopping rule, we employed two criteria: we terminated the algorithm if either

$$\begin{aligned} &\|\nabla_P f(x^k)\| \le 10^{-5}, \\ &\|\max[l - x^k, 0]\| + \|\max[x^k - u, 0]\| \le 10^{-5}(1 + \|x^k\|), \end{aligned} \tag{38}$$

or

$$\|d^k\| \le 10^{-9},$$

where $\nabla_P f(x)$ is the projected gradient defined by

$$\nabla_P f(\bar{x})_i = \begin{cases} \min[0, \nabla f(\bar{x})_i] & \text{if} \quad \bar{x}_i = l_i, \\ \nabla f(\bar{x})_i & \text{if} \quad l_i < \bar{x}_i < u_i, \\ \max[0, \nabla f(\bar{x})_i] & \text{if} \quad \bar{x}_i = u_i. \end{cases}$$

The first conditions are the measure of the violation of the classical KKT conditions; the second one is validated by Theorem 3.2. All the runs were made on a Pentium II with 128 MB RAM, using Fortran90 with the default optimization compiling option. We tested the algorithm on box constrained problems from the CUTE collection [4], and we selected those problems whose number of variables n can be set by the user. We considered the use of a preconditioner for the CG-type method NCCGA, and we refer to the implementation of the NMSB algorithm that uses the preconditioner as P-NMSB, and to the nonpreconditioned version as NMSB. From a theoretical point of view, the choice of the preconditioning matrix is not significant; however, the numerical behavior may be heavily affected by this choice. In particular, we used the incomplete Cholesky factorization (ICFS) code described in [31]. The ICFS routine has the advantage of requiring an amount of memory that can be fixed in advance, and it is equal to $n \cdot p$; we used $p = 5$. In our opinion, a possible drawback to the use of such a preconditioner is the fact that we need to explicitly evaluate and store, although in sparse format, the Hessian matrix $\nabla^2_{FF} f$. When $|F|$ is large and the matrix is dense, this can be a heavy task that may not always be compensated by the minor computational costs of the CG scheme. However, on the problems of the CUTE collections, the use of the preconditioner significantly improved the behavior of the NMSB algorithm, mainly in terms of inner iterations of the CG scheme.

We summarize in Table 1 the results obtained by the NMSB and the P-NMSB codes. As measures of performance, we report the number of function evaluations, Hessian evaluations, inner CG iterations, and cpu time in seconds. The first part of the table contains the number of "wins" and "ties" between NMSB and P-NMSB. We say that an algorithm wins against the other when the measure of the corresponding performance index is better by at least 10% than that of the other algorithm. The algorithms are tied if their performance with respect to a given criterion differs by at most 10%. We are aware that this scoring (as any other kind of global comparison measure) is questionable; on the other hand, we think it is useful to give a concise

TABLE 1
Wins and cumulative results of NMSB versus P-NMSB.

	Wins		Ties	Cumulative results	
	NMSB	P-NMSB		NMSB	P-NMSB
function evl.	4	8	20	872	756
Hessian evl.	4	8	20	790	659
inner CG its.	1	30	1	15223	1010
cpu time	7	13	12	385.32	293.55

idea of what the behavior of the algorithm is. The second part of Table 1 reports the overall results (the sum of the measure of performance on all the problems).

On one problem the two versions of the algorithm did not converge to the same solution. The results in Table 1 do not take into account performance on this problem.

Table 1 shows that the use of a preconditioner strongly influences the number of inner iterations of the CG scheme. On the other performance criteria the two versions NMSB and P-NMSB are tied on many problems (72%), and on the remaining ones P-NMSB performs slightly better than NMSB. Hence we use the preconditioned version P-NMSB for the comparison with other algorithms.

In particular, we compared P-NMSB with LANCELOT [11] and TRON [32]. TRON uses the preconditioner ICFS with the same value of the parameter $p = 5$. LANCELOT implements different types of preconditioners that can be set by the user. We used the default preconditioner that corresponds to a banded preconditioner. For each code (P-NMSB, TRON, LANCELOT) we recorded the number of function evaluations (nf), Hessian evaluations (nh), inner CG iterations (ncg), and cpu time in seconds (time). As regards the stopping criteria for TRON and LANCELOT, we also used for these algorithms the conditions (38). The results obtained by P-NMSB, TRON, and LANCELOT on the problems of the CUTE collections are reported in Table 2. On seven problems P-NMSB did not converge to the same solution of either TRON or LANCELOT. We put an $*$ near the names of these problems. We note, however, that in four cases out of these seven, P-NMSB reaches a point with a lower function value.

In order to better understand the comparative behavior of the codes, we summarize the results in Tables 3 and 4. The problems where the algorithms do not converge to the same solution are not taken into account in these two tables. In particular, Table 3 summarizes the number of wins, in terms of function evaluations, Hessian evaluations, inner CG iterations, and runtime, of P-NMSB versus TRON and of P-NMSB versus LANCELOT. As above, we say that an algorithm wins when the measure of the performance index is at least 10% better than that of the other algorithm, whereas the algorithms tie if their performance with respect to a given criterion differs by at most 10%.

In Table 4, we report the cumulative results, i.e., the sum of function evaluations, Hessian evaluations, inner CG iterations, and runtime on all the problems.

From the tables above it seems fair to say that P-NMSB and TRON behave similarly on most problems with regard to function evaluations, Hessian evaluations, and runtime, while TRON is slightly better in terms of inner iterations. However, since the two algorithms use the same preconditioner, we think that the cpu time is the main important performance criterion, and with respect to that the two algorithms have the same performance. This evidence supports the claim that P-NMSB has no heavy "internal" operations and, in particular, that the use of the penalty function

TABLE 2
Performance of P-NMSB, TRON, and LANCELOT on CUTE problems.

		P-NMSB				TRON				LANCELOT			
Problem	n	nf	nh	ncg	time	nf	nh	ncg	time	nf	nh	ncg	time
BDEXP	15000	12	10	10	3.49	11	11	10	3.88	10	10	12	3.99
CVXBQP1	15000	7	3	3	0.90	2	2	0	0.33	1	1	5	1.33
JNLBRNG1	15625	29	26	31	13.31	26	26	28	12.93	24	24	2029	123.30
JNLBRNG2	15625	20	17	18	8.64	16	16	18	7.75	14	14	898	56.12
JNLBRNG3	15625	14	12	12	6.14	6	6	6	2.74	6	6	219	15.46
JNLBRNGA	15625	25	23	26	10.51	23	23	24	10.60	21	21	1584	86.33
JNLBRNGB	15625	12	10	10	4.70	9	9	11	3.90	8	8	419	23.42
MCCORMCK	15000	6	4	4	1.62	7	6	6	1.61	4	4	4	2.19
NCVXBQP1	15000	6	3	3	1.86	2	2	0	0.31	4	4	2	2.45
NCVXBQP2 *	15000	35	33	37	7.3	8	8	6	1.49	6	6	333	4.78
NCVXBQP3 *	15000	113	101	107	20.65	11	11	8	2.31	7	7	129	2.89
NOBNDTOR	14884	38	36	64	17.02	38	38	51	18.73	36	36	1386	93.47
NONSCOMP	15000	8	6	6	1.64	8	8	7	1.58	8	8	8	2.73
OBSTCLAE *	15625	8	6	16	3.97	27	27	37	11.56	5	5	7452	591.44
OBSTCLAL *	15625	26	25	35	12.01	25	25	28	10.08	24	24	604	33.44
OBSTCLBL *	15625	24	20	32	10.85	19	19	25	9.91	18	18	2088	145.
OBSTCLBM *	15625	15	10	19	6.15	9	9	12	4.71	5	5	1378	110.36
OBSTCLBU *	15625	25	20	30	10.76	20	20	24	9.5	19	19	621	43.03
SSC	15625	9	7	23	6.27	8	8	18	6.31	1	1	111	14.82
TORSION1	14884	40	38	62	16.20	39	39	48	16.07	37	37	1148	66.2
TORSION2	14884	27	23	47	11.29	23	23	31	10.76	14	14	2026	123.88
TORSION3	14884	21	19	27	7.42	20	20	22	6.87	19	19	332	17.0
TORSION4	14884	26	22	40	9.48	24	24	30	9.22	14	14	634	26.83
TORSION5	14884	12	10	13	3.89	11	11	11	3.35	9	9	93	5.36
TORSION6	14884	17	13	25	5.87	15	15	18	5.50	8	8	159	7.15
TORSIONA	14884	41	39	64	19.	39	39	48	17.64	37	37	1147	74.79
TORSIONB	14884	25	22	50	12.01	23	23	33	11.78	15	15	2079	141.52
TORSIONC	14884	21	19	27	8.39	20	20	22	7.65	19	19	332	19.71
TORSIOND	14884	26	22	38	10.52	24	24	30	10.25	14	14	659	31.85
TORSIONE	14884	12	10	13	4.34	11	11	11	3.8	9	9	93	6.15
TORSIONF	14884	17	13	24	6.57	15	15	18	6.06	7	7	166	8.45
TORSIONG	14884	65	63	113	33.40	63	63	82	32.59	61	61	2338	176.64
TORSIONH	14884	9	7	18	4.68	7	7	14	4.26	1	1	102	9.03

TABLE 3
Wins of P-NMSB versus TRON, and of P-NMSB versus P-LANCELOT.

	Wins		Ties	Wins		Ties
	P-NMSB	TRON		P-NMSB	P-LANCELOT	
function evl.	1	11	14	0	19	7
Hessian evl.	5	3	18	2	12	12
inner its.	2	19	5	24	1	1
cpu time	1	6	19	26	0	0

TABLE 4
Cumulative results of P-NMSB, TRON, and LANCELOT.

	Cumulative results		
	P-NMSB	TRON	LANCELOT
function evl.	545	490	401
Hessian evl.	477	489	401
inner its.	771	597	17985
cpu time	229.16	216.47	1140.17

does not affect the computational time of the algorithm, as was expected.

As regards LANCELOT, we observe that it almost always performs better than P-NMSB (and also of TRON) in terms of function and Hessian evaluations. However, the number of inner iterations and the cpu time are significantly worse than those of P-NMSB. Since the preconditioner of LANCELOT is different from the one used by P-NMSB and TRON, it is not possible to draw any definitive general conclusion.

In summary, we think that the NMSB algorithm is reliable and efficient and that its performance is comparable with that of TRON and, at least on the problems of the CUTE collection, is better than that of LANCELOT.

Acknowledgments. We wish to thank an anonymous referee and Dr. Nick Gould for useful comments and suggestions that led to a significant improvement of the paper.

REFERENCES

[1] B. M. AVERICK, R. G. CARTER, J. J. MORÉ, AND G.-L. XUE, *The MINPACK-2 Test Problem Collection*, preprint MCS-P153-0694, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1992.

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

[3] D. P. BERTSEKAS, *Projected Newton methods for optimization problems with simple constraints*, SIAM J. Control Optim., 20 (1982), pp. 221–246.

[4] I. BONGARTZ, A. R. CONN, N. I. M. GOULD, AND PH. L. TOINT, *CUTE: Constrained and unconstrained testing environment*, ACM Trans. Math. Software, 21 (1995), pp. 123–160.

[5] J. V. BURKE AND J. J. MORÉ, *On the identification of active constraints*, SIAM J. Numer. Anal., 25 (1988), pp. 1197–1211.

[6] P. CALAMAI AND J. MORÉ, *Projected gradient for linearly constrained problems*, Math. Programming, 39 (1987), pp. 93–116.

[7] T. COLEMAN AND L. HULBERT, *A direct active set algorithm for large sparse quadratic programs with simple bounds*, Math. Programming, 45 (1987), pp. 373–406.

[8] T. F. COLEMAN AND Y. LI, *An interior trust region approach for nonlinear minimization subject to bounds*, SIAM J. Optim., 6 (1996), pp. 418–445.

[9] A. R. CONN, N. I. M. GOULD, AND PH. 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.

[10] A. Conn, N. Gould, and Ph. Toint, *Testing a class of methods for solving minimization problems with simple bounds on the variables*, Math. Comp., 50 (1988), pp. 399–430.

[11] A. Conn, N. Gould, and Ph. Toint, *LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A)*, Springer-Verlag, Berlin, New York, Heidelberg, 1992.

[12] R. Cottle and M. Goheen, *A special class of large quadratic programs*, in Nonlinear Programming 3, O. L. Mangasarian, R. R. Meyer, and S. Robinson, eds., Academic Press, New York, 1978, pp. 361–390.

[13] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.

[14] J. C. Dunn, *Global and asymptotic convergence rate estimates for a class of projected gradient processes*, SIAM J. Control Optim., 19 (1981), pp. 368–400.

[15] F. Facchinei, *Minimization of SC^1 functions and the Maratos effect*, Oper. Res. Lett., 17 (1995), pp. 131–137.

[16] F. Facchinei, A. Fischer, and C. Kanzow, *On the accurate identification of active constraints*, SIAM J. Optim., 9 (1998), pp. 14–32.

[17] F. Facchinei, J. Júdice, and J. Soares, *An active set Newton algorithm for large-scale nonlinear programs with box constraints*, SIAM J. Optimization, 8 (1998), pp. 158–186.

[18] F. Facchinei and S. Lucidi, *A class of penalty functions for optimization problems with bound constraints*, Optimization, 26 (1992), pp. 239–259.

[19] F. Facchinei and S. Lucidi, *A Class of Methods for Optimization Problems with Simple Bounds. Part 2: Algorithms and Numerical Results*, Technical report R.336, IASI-CNR, Roma, Italy, 1992.

[20] F. Facchinei and S. Lucidi: *Quadratically and superlinearly convergent algorithms for the solution of inequality constrained minimization problems*, J. Optim. Theory Appl., 85 (1995), pp. 265–289.

[21] F. Facchinei, S. Lucidi, and L. Palagi, *A Truncated Newton Algorithm for Large Scale Box Constrained Optimization*, TR 15-99 DIS, Università di Roma "La Sapienza," Rome, Italy, 1999.

[22] A. Friedlander, J. M. Martinez, S. A. Santos, *A new trust region algorithm for bound constrained minimization*, Appl. Math. Optim., 30 (1994), pp. 235–266.

[23] P. Gill, W. Murray, and M. Wright, *Practical Optimization*, Academic Press, New York, 1981.

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

[25] L. Grippo, F. Lampariello, and S. Lucidi, *A truncated Newton method with nonmonotone linesearch for unconstrained minimization*, J. Optim. Theory Appl., 60 (1989), pp. 401–419.

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

[27] M. Heinkenschloss, M. Ulbrich, and S. Ulbrich, *Superlinear and quadratic convergence of affine-scaling interior-point Newton methods for problems with simple bounds without strict complementarity assumption*, Math. Programming, 86 (1999), pp. 615–635.

[28] C. Kanzow, *Strictly feasible equation-based methods for mixed complementarity problems*, Numer. Math., 89 (2001), pp. 135–160.

[29] C. Kanzow, *An active set-type Newton method for constrained nonlinear systems*, in Complementarity: Applications, Algorithms and Extensions, M. C. Ferris, O. L. Mangasarian, J.-S. Pang, eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, 2001, pp. 179–200.

[30] M. Lescrenier, *Convergence of trust region algorithms for optimization with bounds when strict complementarity does not hold*, SIAM J. Numer. Anal., 28 (1991), pp. 476–495.

[31] C.-J. Lin and J. J. Moré, *Incomplete Cholesky factorizations with limited memory*, SIAM J. Sci. Comput., 21 (1999), pp. 24–45.

[32] C.-J. Lin and J. J. Moré, *Newton's method for large bound-constrained optimization problems*, SIAM J. Optim., 9 (1999), pp. 1100–1127.

[33] S. Lucidi and M. Roma, *Numerical experiences with new truncated Newton methods in large scale unconstrained optimization*, Comput. Optim. Appl., 7 (1997), pp. 71–87.

[34] S. Lucidi, F. Rochetich, and M. Roma, *Curvilinear stabilization techniques for truncated Newton methods in large scale unconstrained optimization*, SIAM J. Optim., 8 (1998), pp. 916–939.

[35] R. Mifflin, *Semismooth and semiconvex functions in constrained optimization*, SIAM J. Control Optim., 15 (1977), pp. 959–972.

[36] J. Moré and G. Toraldo, *Algorithms for bound constrained quadratic programming problems*, Numer. Math., 55 (1989), pp. 377–400.

[37] J. J. Moré and G. Toraldo, *On the solution of large quadratic programming problems with bound constraints*, SIAM J. Optim., 1 (1991), pp. 93–113.
[38] S. Nash and A. Sofer, *A barrier method for large-scale constrained optimization*, ORSA J. Computing, 5 (1993), pp. 40–53.
[39] L. Qi and J. Sun, *A nonsmooth version of Newton's method*, Math. Programming, 58 (1993), pp. 353–368.
[40] S. M. Robinson: *Generalized equations*, in Mathematical Programming: The State of the Art, A. Bachem, M. Groetschel, and B. Korte, eds., Springer-Verlag, Berlin, 1983, pp. 346–367.
[41] S. Wright, *Algorithms for Minimization Subject to Bounds*, Technical report MCS-P32-1288, Argonne National Laboratory, Mathematics and Computer Science Division, Chicago, IL, 1988.

SIAM J. OPTIM.
Vol. 12, No. 4, pp. 1126–1146

A MULTIPLE-CUT ANALYTIC CENTER CUTTING PLANE METHOD FOR SEMIDEFINITE FEASIBILITY PROBLEMS*

KIM-CHUAN TOH†, GONGYUN ZHAO†, AND JIE SUN‡

Abstract. We consider the problem of finding a point in a nonempty bounded convex body Γ in the cone of symmetric positive semidefinite matrices $\mathcal{S}^m_+$. Assume that Γ is defined by a separating oracle, which, for any given $m \times m$ symmetric matrix $\hat{Y}$, either confirms that $\hat{Y} \in \Gamma$ or returns several selected cuts, i.e., a number of symmetric matrices $A_i,\ i = 1, \ldots, p,\ p \le p_{\max}$, such that Γ is in the polyhedron $\{Y \in \mathcal{S}^m_+ \mid A_i \bullet Y \le A_i \bullet \hat{Y}, i = 1, \ldots, p\}$. We present a multiple-cut analytic center cutting plane algorithm. Starting from a trivial initial point, the algorithm generates a sequence of positive definite matrices which are approximate analytic centers of a shrinking polytope in $\mathcal{S}^m_+$. The algorithm terminates with a point in Γ within $O(m^3 p_{\max}/\epsilon^2)$ Newton steps (to leading order), where ϵ is the maximum radius of a ball contained in Γ.

Key words. analytic center, cutting plane methods, multiple cuts, semidefinite programming

AMS subject classifications. 65K05, 90C25

PII. S1052623400370503

1. Introduction.

Let $\mathcal{S}^m$ be the set of $m \times m$ symmetric matrices, and let $\mathcal{S}^m_+$ be its subset of symmetric positive semidefinite matrices. We consider the problem of finding a point in a convex subset Γ of $\mathcal{S}^m_+$. We assume that Γ contains a full-dimensional closed ball with radius $\epsilon > 0$. The set Γ is implicitly defined by a separating oracle, which, for any given $m \times m$ symmetric matrix $\hat{Y}$, either confirms that $\hat{Y} \in \Gamma$ or returns several cuts, i.e., a number of symmetric matrices $A_i,\ i = 1, \ldots, p,\ p \le p_{\max}$, such that Γ is in the polyhedron $\{Y \in \mathcal{S}^m_+ \mid A_i \bullet Y \le A_i \bullet \hat{Y}, i = 1, \ldots, p\}$. Here $p_{\max}$ is the maximum number of cuts admitted in each iteration.

In a recent paper [8], we presented an analytic center cutting plane method for the case $p_{\max} = 1$, in which a single cut is added in each iteration. The method was shown to have a worst-case complexity of $O(m^3/\epsilon^2)$ (to leading order). However, to make a cutting plane algorithm practically efficient, adding multiple cuts is often necessary. The purpose of this paper is to propose and analyze an analytic cutting plane method that uses multiple cuts for solving the convex semidefinite feasibility problem mentioned above. In admitting multiple cuts in an analytic center cutting plane method, we face some new theoretical problems that are different from the single-cut situation; these include (a) the problem of finding a feasible starting point for the Newton iteration after several new cuts have been added, (b) the estimation of the number of Newton steps needed to obtain a new approximate center through estimating the changes in the primal-dual potential function.

Our paper extends the multiple-cut schemes of Goffin and Vial [2], Luo [5], and Ye [10] from $\mathbb{R}^m_+$ to $\mathcal{S}^m_+$. Such extensions not only broaden the applications of cutting

*Received by the editors April 4, 2000; accepted for publication (in revised form) October 29, 2001; published electronically April 26, 2002.
http://www.siam.org/journals/siopt/12-4/37050.html

†Department of Mathematics, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119620 (mattohkc@math.nus.edu.sg, matzgy@math.nus.edu.sg). The research of the first author was supported in part by the Singapore-MIT Alliance.

‡Department of Decision Sciences, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119620 (jsun@nus.edu.sg). The research of this author was supported by National University of Singapore (NUS) Academic Research Grant R314-000-009/026-112 and the Singapore-MIT Alliance.

plane methods but also extend several classical theoretical results for nonnegative vectors to positive semidefinite matrices. We note that for our multiple-cut analytic center cutting plane algorithm, the complexity analysis on the number of Newton iterations per oracle call follows the approach in [3]. For the complexity analysis on the number of oracle calls, we follow the approach in [10], but we simplify the proofs of some results analogous to those in [10] by considering all the added cuts simultaneously instead of inductively.

In this paper we will show that, starting from a trivial initial point, the multiple-cut algorithm generates a sequence of positive definite matrices which are approximate analytic centers of a shrinking polytope in $\mathcal{S}^m_+$. The algorithm will stop with a solution in at most $O(m^3 p_{\max}/\epsilon^2)$ (to leading order) Newton steps. Our analysis shows that when the problem is specialized to the space of positive semidefinite diagonal matrices (which is equivalent to the nonnegative orthant R^m_+), the complexity bound is reduced to $O(m^2 p_{\max}/\epsilon^2)$. This complexity bound is lower than the existing bound of $O(m^2 p^2_{\max}/\epsilon^2)$ obtained in [2] and [10], where the same cuts are considered. Our bound appears to be better than that obtained in [5]. (Note that the proof for the bound appearing in [5] is incomplete, and, to the best of our knowledge, a provable bound should be $O(m^2 p^2_{\max}/\epsilon^2)$.) Furthermore, the analysis in [5] is carried out only for the so-called shallow cuts, which are placed at some distance away from the current testing point and hence may not be as efficient as our proposed algorithm, where the cuts pass through the testing point.

We are able to obtain better complexity results than existing ones even when the problem is specialized to $\mathbb{R}^m_+$, because in each oracle call we admit only cuts that are sufficiently good. We shall not give the precise definition of "goodness" here but refer the reader to section 4. Roughly speaking, based on our criteria, the admitted cuts $A_i,\ i = 1, \ldots, p$, in each oracle call are effective in reducing the size of the polytope in the sense that each should be able to delete a sizable portion of the current polytope that cannot be otherwise deleted by the other admitted cuts. One obvious advantage of having such a selection criterion is that the number of cuts added in each iteration is reduced, since only effective cuts are admitted, and this translates into savings in the computational cost in each Newton step.

We will now introduce some notations. For matrices $A, Y \in \mathcal{S}^m$, we define

$$A \bullet Y := \mathrm{Tr}(A^T Y) = \sum_{i,j=1}^{m} A_{ij} Y_{ij},$$

where "T" stands for the transpose, and "Tr" denotes the trace. We write $Y \succ 0$ and $Y \succeq 0$ if Y is positive definite and positive semidefinite, respectively. For $Y \succeq 0$, we denote its symmetric square root by $Y^{1/2}$. The 2-norm of a vector x is denoted by $\|x\|$, and the matrix 2-norm of a matrix A is denoted by $\|A\|$. For $A \in \mathcal{S}^m$, we write

$$\|A\|_F := (A \bullet A)^{1/2}, \quad \lambda(A) := (\lambda_1(A), \ldots, \lambda_m(A))^T,$$

where $\lambda_1(A), \ldots, \lambda_m(A)$ are the eigenvalues of A. Note that $\|A\|_F = \|\lambda(A)\|$ and $\|A\| = \|\lambda(A)\|_\infty$. We will use these facts in the paper without explicitly mentioning them. For a positive vector $x \in \mathbb{R}^n$, we write

$$\ln x := \sum_{i=1}^{n} \ln x_i.$$

We use diag(x) to denote the diagonal matrix whose diagonal is the vector x. For a positive vector x, we will use x^{-1} to denote the vector obtained from x by inverting all of its components.

Generally, we use capital letters for matrices, lower case letters for vectors, and Greek letters for scalars. For convenience, we let $\bar{m} = m(m+1)/2$.

Let **svec** be an isometry identifying $\mathcal{S}^m$ with $\mathbb{R}^{\bar{m}}$, so that $K \bullet L = \mathbf{svec}(K)^T \mathbf{svec}(L)$, and let **smat** be the inverse of **svec**. Given any $G \in \mathcal{S}^m$, we let $G \circledast G \in \mathbb{R}^{\bar{m} \times \bar{m}}$ be the unique symmetric matrix such that

$$(G \circledast G)\, \mathbf{svec}(M) \;=\; \mathbf{svec}(GMG) \quad \forall\, M \in \mathcal{S}^m.$$

It is easy to see that if G is positive definite, then $G \circledast G$ is positive definite and $(G \circledast G)^{1/2} = G^{1/2} \circledast G^{1/2}$. If G is nonsingular, then $(G \circledast G)^{-1} = G^{-1} \circledast G^{-1}$.

Throughout, we make the following assumptions:

A1. Γ is a convex subset of $\mathcal{S}^m_+$.

A2. $\Gamma \subset \Omega_0$, where $\Omega_0 := \{Y \in \mathcal{S}^m \mid 0 \preceq Y \preceq I\}$.

A3. Γ contains a full-dimensional ball of radius $\epsilon > 0$. That is, there exists $Y^c \in \mathcal{S}^m$ such that $\{Y \in \mathcal{S}^m : \|Y - Y^c\|_F \le \epsilon\} \subset \Gamma$.

Note that Assumption A2 is made for convenience. It can be satisfied by scaling if the original convex set $\hat{\Gamma}$ is bounded. That is, suppose there exists a constant $\gamma > 0$ such that for all $Y \in \hat{\Gamma}$, $\|Y\| \le \gamma$. Then the scaled set $\Gamma = \{Y/\gamma \mid Y \in \hat{\Gamma}\}$ satisfies A2.

The organization of the paper is as follows. In section 2, we describe our multiple-cut analytic center cutting plane algorithm for semidefinite feasibility problems. Section 3 is devoted to the analysis of the computation of an approximate analytic center for a working set. In particular, we establish the number of Newton steps required to compute an approximate analytic center in terms of the number of cuts added. In section 4, we establish the dual potential increment when the current working set is changed to the next working set. Subsequently, we establish complexity results for our multiple-cut cutting plane algorithm.

2. A multiple-cut analytic center cutting plane method. We first define the analytic center and then propose a multiple-cut analytic center cutting plane method at the end of this section.

Let $A_i \bullet Y \le c_i$, $i = 1, \ldots, n_k$, be all the cuts defining the kth working set Ω_k. Define

$$\mathcal{A} := (\mathbf{svec}A_1, \mathbf{svec}A_2, \ldots, \mathbf{svec}A_{n_k}), \quad c := (c_1, c_2, \ldots, c_{n_k})^T.$$

Then the set Ω_k can be represented as

$$\Omega_k = \{Y \in \Omega_0 \mid \mathcal{A}^T \mathbf{svec} Y \le c\}.$$

We define the following *potential function* on the set Ω_k:

$$\phi_k(Y) \;=\; -\sum_{i=1}^{n_k} \ln(c_i - A_i \bullet Y) - \ln(\det Y) - \ln(\det(I - Y)),$$

where "det" denotes the determinant. We let

$$\phi_k(\Omega) := \min\{\phi_k(Y) \mid Y \in \Omega\}.$$

The unique minimizer of $\phi_k(Y)$ over Ω_k is known as the analytic center of Ω_k.

It is easy to see that the analytic center of the initial working set Ω_0 is $I/2$, where I is the identity matrix. As a matter of fact,

$$\begin{aligned}\phi_0(Y) &= -\ln(\det Y) - \ln(\det(I-Y)) \\ &= -\ln\prod_{i=1}^{m}\lambda_i(Y) - \ln\prod_{i=1}^{m}\lambda_i(I-Y) \\ &= -\sum_{i=1}^{m}\ln\left[\lambda_i(Y)(1-\lambda_i(Y))\right].\end{aligned}$$

The minimum of $\phi_0(Y)$ must satisfy $\lambda_1(Y) = \cdots = \lambda_m(Y) = 1/2$, and hence $Y = I/2$.

It is known [7, Proposition 5.4.5] that ϕ_k is a strongly 1-self-concordant function on Ω and

$$\nabla\phi_k(Y) = \mathbf{svec}\left(\sum_{i=1}^{n_k}\frac{A_i}{c_i - A_i \bullet Y} - Y^{-1} + (I-Y)^{-1}\right),$$

$$\nabla^2\phi_k(Y) \;=\; \mathcal{A}S^{-2}\mathcal{A}^T + Y^{-1} \circledast Y^{-1} + (I-Y)^{-1} \circledast (I-Y)^{-1},$$

where $S = \operatorname{diag}(s)$ and $s = c - \mathcal{A}^T\mathbf{svec}(Y) > 0$. Strictly speaking, $\nabla\phi_k(Y)$ should be the $m \times m$ matrix within the round brackets. However, we have identified the $m \times m$ matrix with a vector in $\mathbb{R}^{\bar{m}}$ through the linear isometry **svec**. Similarly, $\nabla^2\phi_k(Y)$ is identified with an $\mathbb{R}^{\bar{m}} \times \mathbb{R}^{\bar{m}}$ matrix.

The optimality conditions for minimizing ϕ_k are

$$\begin{aligned} Sx &= e, \quad (e \text{ denotes the vector of ones}) \\ YZ &= I, \\ (I-Y)V &= I, \\ \mathcal{A}^T\mathbf{svec}Y + s &= c, \\ \mathcal{A}x - \mathbf{svec}Z + \mathbf{svec}V &= 0, \\ I \succ Y \succ 0, \quad Z, V \succ 0, \quad & s, x > 0. \end{aligned} \tag{2.1}$$

With a slight abuse of language, we also call the solution $(\bar{Y}, \bar{s}, \bar{x}, \bar{Z}, \bar{V})$ of (2.1) the analytic center of Ω_k.

DEFINITION 2.1. *Given a point $(Y, s, x, Z, V) \in \mathcal{S}^m \times \mathbb{R}^{n_k} \times \mathbb{R}^{n_k} \times \mathcal{S}^m \times \mathcal{S}^m$, with $0 \prec Y \prec I$, we define*

$$\eta(Y, s, x, Z, V) \;=\; \sqrt{\|Sx - e\|^2 + \|\lambda(YZ) - e\|^2 + \|\lambda((I-Y)V) - e\|^2}. \tag{2.2}$$

We call (Y, s, x, Z, V) an η-approximate (analytic) center of Ω_k if $\eta(Y, s, x, Z, V) \le \eta$, all the linear equalities in (2.1) *are satisfied, and $x, s > 0$, $Z, V \succ 0$. Obviously, a 0-approximate center is exactly the analytic center of Ω.*

DEFINITION 2.2. *Given $Y \in \mathcal{S}^m$ such that $0 \prec Y \prec I$, and $s = c - \mathcal{A}^T\mathbf{svec}(Y) > 0$, we define*

$$\delta_k(Y) = \sqrt{\nabla\phi_k(Y)^T\left[\nabla^2\phi_k(Y)\right]^{-1}\nabla\phi_k(Y)}\,. \tag{2.3}$$

It was shown [8] that the following lemma holds.

LEMMA 2.3. *Given $Y \in \mathcal{S}^m$ such that $0 \prec Y \prec I$, let $s = c - \mathcal{A}^T\mathbf{svec}(Y)$. We have*

$$\begin{aligned}\delta_k(Y) &= \eta(Y, s, x_Y, Z_Y, V_Y)\\ &= \min\{\eta(Y, s, x, Z, V) \,:\, \mathcal{A}x - \mathbf{svec}(Z) + \mathbf{svec}(V) = 0,\ x \in \mathbb{R}^k,\ Z, V \in \mathcal{S}^m\}. \qquad (2.4)\end{aligned}$$

Remark. Given $Y \in \mathcal{S}^m$ such that $0 \prec Y \prec I$, $s = c - \mathcal{A}^T\mathbf{svec}(Y) > 0$, and $\delta_k(Y) < \eta < 1$, the minimizer (x_Y, Z_Y, V_Y) of (2.4) satisfies $x_Y > 0$, and $Z_Y, V_Y \succ 0$. For such a Y, we will call Y an η-approximate center of Ω_k in the sense that the point (Y, s, x_Y, Z_Y, V_Y) is an η-approximate center.

We will now describe our algorithm.

A multiple-cut analytic center cutting plane algorithm.

Step 0. Select $\eta \in (0, 1-\sqrt{3}/2)$, and pick $\bar{\delta} \in (\eta, 1)$. Set $k = 0$. Let Ω_0 be the initial working set, and let $Y_0 = I/2$ be the initial point.

Step 1. At the kth iteration, call the oracle to either confirm that Y_k is a feasible point of Γ or return p_k matrices $A_{n_k+1}, \ldots, A_{n_k+p_k} \in \mathcal{S}^m$ with $\|A_{n_k+i}\|_F = 1$. If $Y_k \in \Gamma$, stop; otherwise, construct the new working set

$$\Omega_{k+1} = \{Y \in \Omega_k \,:\, A_{n_k+i} \bullet Y \le A_{n_k+i} \bullet Y_k,\ i = 1, \ldots, p_k\}.$$

Step 2. Find a point $\tilde{Y}$ in the interior of Ω_{k+1} (discussed in section 3).

Step 3. (Recentering) Starting with the point $Y = \tilde{Y}$ in Step 2, perform the dual Newton method:

3.1. If $\delta_{k+1}(Y) < \eta$, set $Y_{k+1} = Y$, $k := k+1$; go to Step 1.

3.2. Otherwise, set

$$Y_+ = Y - \bar{\alpha}\,\mathbf{smat}\left([\nabla^2\phi_{k+1}(Y)]^{-1}\nabla\phi_{k+1}(Y)\right),$$

where $\bar{\alpha}$ is determined as follows: if $\delta_{k+1}(Y) \ge \bar{\delta}$, $\bar{\alpha} = \frac{1}{1+\delta_{k+1}(Y)}$; else, $\bar{\alpha} = 1$. Set $Y = Y_+$. Go to Step 3.1.

Note that we need the restriction $\eta < 1 - \sqrt{3}/2$ in order to construct the point $\tilde{Y}$ in Step 2.

3. Restoration of centrality. In our cutting plane algorithm, approximate analytic centers are found by using the dual Newton method. Our aim in this section is to estimate the number of Newton steps required to find an approximate analytic center for a newly constructed working set. We do so by estimating the amount of potential value we should reduce for the new set. The mechanics are as follows. Since the potential function is 1-self-concordant, each Newton step can reduce the potential function by a constant amount. Thus to estimate the number of Newton steps needed to find an approximate analytic center for a new working set, all we need is to estimate the amount of potential value we should reduce for the new set.

To find an approximate analytic center for a new working set, we would ideally want the Newton method to start with the preceding approximate analytic center Y_k. However, Y_k is not in the interior of the new working set Ω_{k+1}, since the new cuts pass through this point. Thus our immediate task is to find an interior point in Ω_{k+1} and then use this point as the starting point for the Newton method.

Let n_k be the number of cuts defining the set Ω_k. Suppose that p_k new cuts are added to form the new set Ω_{k+1}. Recall that

$$\begin{aligned}\mathcal{A} &:= (\mathbf{svec}A_1, \mathbf{svec}A_2, \ldots, \mathbf{svec}A_{n_k}), \qquad c := (c_1, c_2, \ldots, c_{n_k})^T,\\ \mathcal{B}_k &:= (\mathbf{svec}A_{n_k+1}, \mathbf{svec}A_{n_k+2}, \ldots, \mathbf{svec}A_{n_k+p_k}), \qquad d := \mathcal{B}_k^T\mathbf{svec}Y_k.\end{aligned}$$

Then the sets Ω_k and Ω_{k+1} can be written as

$$\Omega_k = \{Y \in \Omega_0 \mid \mathcal{A}^T \mathbf{svec} Y \leq c\}, \qquad \Omega_{k+1} = \{Y \in \Omega_k \mid \mathcal{B}_k^T \mathbf{svec} Y \leq d\}.$$

Let $H_k = \nabla^2 \phi_k(Y_k)$ and

$$M_k := \mathcal{B}_k^T H_k^{-1} \mathcal{B}_k.$$

Suppose $(Y_k, s^k, x^k, Z_k, V_k)$ is an η-approximate center with $\eta < 1 - \sqrt{3}/2$. (Note that, by Lemma 2.3, $\delta_k(Y_k) \leq \eta(Y_k, s^k, x^k, Z_k, V_k) \leq \eta$.) We will now construct a point $(\tilde{Y}, \tilde{s}, \tilde{x}, \tilde{Z}, \tilde{V})$ that is in the interior of Ω_{k+1}, using a procedure similar to that in Goffin and Vial [2]. To this end, consider the following convex minimization problem:

$$\begin{array}{ll} \min & p_k \omega^T M_k \, \omega \, - \, \ln \omega \\ \text{such that} & \omega = (\omega_1, \ldots, \omega_{p_k})^T > 0. \end{array}$$

Evidently, the above problem has a unique solution that is also the unique solution to the KKT-conditions:

$$M_k \, \omega = \xi, \tag{3.1a}$$

$$2 p_k \, \omega_i \, \xi_i = 1, \quad \omega_i, \ \xi_i \ > 0, \quad i = 1, \ldots, p_k. \tag{3.1b}$$

Let $(\tilde{\omega}, \tilde{\xi})$ be an approximate solution of the above KKT-conditions, where (3.1a) is satisfied exactly and $\max\{|2 p_k \tilde{\omega}_i \tilde{\xi}_i - 1| : i = 1, \ldots, p_k\} \leq 1/2$. Note that, in this case,

$$\tilde{\omega}^T M_k \, \tilde{\omega} \ \leq \ \frac{3}{4}. \tag{3.2}$$

Note that to find such a pair $(\tilde{\omega}, \tilde{\xi})$, we can apply Newton's method to (3.1a) and (3.1b), where the computational work for each Newton iteration is $O(p_k^3)$. In general, this constitutes only a very small fraction of the total computational work involved in finding an approximate analytic center for Ω_{k+1}. In order not to lengthen the paper unnecessarily, we shall not establish the complexity of the Newton method for finding $(\tilde{\omega}, \tilde{\xi})$ in this paper. Interested readers can refer to [3] for such results.

Let $U_k = I - Y_k$ and

$$\Delta Y \ = \ -\mathbf{smat}(H_k^{-1} \mathcal{B}_k \tilde{\omega}), \quad \Delta s = -\mathcal{A}^T \mathbf{svec} \Delta Y, \tag{3.3}$$

$$\Delta x = S_k^{-2} \mathcal{A}^T \mathbf{svec} \Delta Y, \quad \Delta Z = -Y_k^{-1} (\Delta Y) Y_k^{-1}, \quad \Delta V = U_k^{-1} (\Delta Y) U_k^{-1}. \tag{3.4}$$

Define

$$\tilde{Y} \ = \ Y_k + \Delta Y, \qquad \tilde{s} \ = \ \begin{pmatrix} s^k + \Delta s \\ \tilde{\xi} \end{pmatrix}, \tag{3.5}$$

$$\tilde{x} = \begin{pmatrix} x^k + \Delta x \\ \tilde{\omega} \end{pmatrix}, \quad \tilde{Z} = Z_k + \Delta Z, \quad \tilde{V} = V_k + \Delta V. \tag{3.6}$$

We refer the reader to [3] for an illuminating discussion on the motivation for considering the optimization problem (3.1a)–(3.1b) in constructing the strictly interior point of Ω_{k+1} above.

It is readily shown that the following result holds:

$$\begin{aligned}&\|S_k^{-1}\mathcal{A}^T\mathbf{svec}(\Delta Y)\|^2 \;+\; \|Y_k^{-1/2}(\Delta Y)Y_k^{-1/2}\|_F^2 \;+\; \|U_k^{-1/2}(\Delta Y)U_k^{-1/2}\|_F^2\\ &= \mathbf{svec}(\Delta Y)^T H_k\,\mathbf{svec}(\Delta Y) \;=\; \tilde{\omega}^T M_k\tilde{\omega} \;\le\; \frac{3}{4}.\end{aligned} \tag{3.7}$$

LEMMA 3.1. *For any vector $q = (q_1, \ldots, q_n)^T$ with $\|q\| < 1$, the following inequality holds:*

$$-\ln(e-q) \;\le\; e^T q + \frac{\|q\|^2}{2(1-\|q\|)}.$$

Proof. For this proof, we refer to [11]. □

LEMMA 3.2. *Suppose $(Y_k, s^k, x^k, Z_k, V_k)$ is an η-approximate center with $\eta < 1$. Then the following inequalities hold:*

$$\|X_k^{-1}\Delta x\| \le \frac{1}{1-\eta}\|S_k^{-1}\Delta s\|,$$

$$\|Z_k^{-1/2}(\Delta Z)Z_k^{-1/2}\|_F \le \frac{1}{1-\eta}\|Y_k^{-1/2}(\Delta Y)Y_k^{-1/2}\|_F,$$

$$\|V_k^{-1/2}(\Delta V)V_k^{-1/2}\|_F \le \frac{1}{1-\eta}\|U_k^{-1/2}(\Delta Y)U_k^{-1/2}\|_F,$$

where $X_k = \mathrm{diag}(x^k)$.

Proof. We shall omit the proof of the first inequality, as it is easy. Now we proceed with the proof of the second one. We have

$$\begin{aligned}\|Z_k^{-1/2}(\Delta Z)Z_k^{-1/2}\|_F^2 &= \sum_{i=1}^m \lambda_i\left(Z_k^{-1/2}Y_k^{-1/2}(Y_k^{-1/2}\Delta Y Y_k^{-1/2})Y_k^{-1/2}Z_k^{-1/2}\right)^2\\ &= \sum_{i=1}^m \theta_i^2\,\lambda_i\left(Y_k^{-1/2}\Delta Y Y_k^{-1/2}\right)^2\\ &\le \left(\max_{1\le i\le m}\theta_i^2\right)\,\|Y_k^{-1/2}\Delta Y Y_k^{-1/2}\|_F^2,\end{aligned}$$

where we have used a theorem of Ostrowski [4, p. 225] in the second equality above, and the θ_i's are scalars such that

$$\lambda_{\min}(Z_k^{-1/2}Y_k^{-1}Z_k^{-1/2}) \;\le\; \theta_i \;\le\; \lambda_{\max}(Z_k^{-1/2}Y_k^{-1}Z_k^{-1/2}).$$

Noting that $\lambda_{\max}(Z_k^{-1/2}Y_k^{-1}Z_k^{-1/2}) \le 1/(1-\eta)$, we have proved the required inequality. The last inequality in the lemma can be proved similarly. □

THEOREM 3.3. *Suppose $(Y_k, s^k, x^k, Z_k, V_k)$ is an η-approximate center with $\eta < 1 - \sqrt{3}/2$. Then the point $(\tilde{Y}, \tilde{s}, \tilde{x}, \tilde{Z}, \tilde{V})$ constructed in* (3.5)–(3.6) *satisfies the last three conditions in* (2.1).

Proof. First, we show that $\tilde{s} > 0$ and $0 \prec \tilde{Y} \prec I$. We have

$$s^k + \Delta s \;=\; s^k - \mathcal{A}^T\mathbf{svec}(\Delta Y) = S_k\left[e - S_k^{-1}\mathcal{A}^T\mathbf{svec}(\Delta Y)\right] > 0,$$

since $\|S_k^{-1}\mathcal{A}^T\mathbf{svec}(\Delta Y)\| \le \sqrt{3}/2 < 1$ from (3.7). On the other hand, we also have

$$\tilde{Y} \;=\; Y_k^{1/2}(I + Y_k^{-1/2}\Delta Y Y_k^{-1/2})Y_k^{1/2} \succ 0,$$

since $\|Y_k^{-1/2}\Delta Y Y_k^{-1/2}\|_F \le \sqrt{3}/2 < 1$. That $\tilde{Y} \prec I$ can be shown similarly. Furthermore,

$$\left[\begin{array}{c} \mathcal{A}^T\mathbf{svec}Y \\ \mathcal{B}_k^T\mathbf{svec}Y \end{array}\right] + s \;=\; \left[\begin{array}{c} c \\ d + \mathcal{B}_k^T\mathbf{svec}(\Delta Y) + \tilde{\xi} \end{array}\right] \;=\; \left[\begin{array}{c} c \\ d \end{array}\right],$$

where we used the fact that, from (3.1a), $\mathcal{B}_k^T\mathbf{svec}(\Delta Y) = -M_k\tilde{\omega} = -\tilde{\xi}$.

Next we show that $\tilde{x} > 0$ and $\tilde{Z}, \tilde{V} \succ 0$. We have

$$\tilde{Z} \;=\; Z_k^{1/2}(I + Z_k^{-1/2}(\Delta Z)Z_k^{-1/2})Z_k^{1/2} \;\succ\; 0,$$

since, by Lemma 3.2,

$$\|Z_k^{-1/2}(\Delta Z)Z_k^{-1/2}\|_F \;\le\; \frac{1}{(1-\eta)}\|Y_k^{-1/2}(\Delta Y)Y_k^{-1/2}\|_F \;\le\; \frac{\sqrt{3}}{2(1-\eta)} \;<\; 1.$$

Furthermore,

$$\begin{aligned}
&[\mathcal{A} \;\; \mathcal{B}_k]\,\tilde{x} - \mathbf{svec}\tilde{Z} + \mathbf{svec}\tilde{V} \\
&\quad = \mathcal{A}\,x^k + \mathcal{A}\,\Delta x + \mathcal{B}_k\tilde{\omega} - \mathbf{svec}Z_k - \mathbf{svec}\Delta Z + \mathbf{svec}V_k + \mathbf{svec}\Delta V \\
&\quad = \mathcal{A}\,\Delta x + \mathcal{B}_k\tilde{\omega} - \mathbf{svec}\Delta Z + \mathbf{svec}\Delta V \\
&\quad = \mathcal{A}\,S_k^{-2}\mathcal{A}^T\mathbf{svec}(\Delta Y) + Y_k^{-1} \circledast Y_k^{-1}\mathbf{svec}(\Delta Y) + U_k^{-1} \circledast U_k^{-1}\mathbf{svec}(\Delta Y) + \mathcal{B}_k\tilde{\omega} \\
&\quad = H_k\mathbf{svec}(\Delta Y) + \mathcal{B}_k\tilde{\omega} \;=\; 0. \qquad \square
\end{aligned}$$

Up to this point, we have succeeded in finding in the interior of Ω_{k+1} a point $\tilde{Y}$ that is derived from Y_k. Our next task is to estimate the potential value of the new point in Ω_{k+1}.

LEMMA 3.4. *Suppose $\delta_k(Y_k) \le \eta$. Then the potential value $\phi_{k+1}(\tilde{Y})$ satisfies the following inequality:*

$$\phi_{k+1}(\tilde{Y}) \;\le\; \phi_k(Y_k) + \frac{\sqrt{3}}{2}\eta + \frac{3}{4(2-\sqrt{3})} - \ln\tilde{\xi}. \tag{3.8}$$

Proof. Let $\tilde{U} = I - \tilde{Y}$ and $U_k = I - Y_k$. We have

$$\phi_{k+1}(\tilde{Y}) \;=\; -\ln\tilde{s} - \ln(d - \mathcal{B}_k^T\mathbf{svec}(\tilde{Y})) - \ln(\det\tilde{Y}) - \ln(\det\tilde{U}) \;=\; \phi_k(\tilde{Y}) - \ln\tilde{\xi}. \tag{3.9}$$

Note that we used the fact that $d - \mathcal{B}_k^T\mathbf{svec}(\tilde{Y}) = -\mathcal{B}_k^T\mathbf{svec}(\Delta Y) = \tilde{\xi}$. Now

$$\begin{aligned}
\phi_k(\tilde{Y}) \;&=\; -\ln(s^k + \Delta s) - \ln\det(Y_k + \Delta Y) - \ln\det(U_k - \Delta Y) \\
&= \phi_k(Y_k) - \ln(e + S_k^{-1}\Delta s) - \ln\det(I + Y_k^{-1/2}\Delta Y\, Y_k^{-1/2}) \\
&\qquad - \ln\det(I - U_k^{-1/2}\Delta Y\, U_k^{-1/2})
\end{aligned}$$

$$
\begin{aligned}
&= \phi_k(Y_k) - \ln(e + S_k^{-1}\Delta s) - \ln\Big(e + \lambda(Y_k^{-1/2}(\Delta Y)Y_k^{-1/2})\Big) \\
&\qquad - \ln\Big(e - \lambda(U_k^{-1/2}(\Delta Y)U_k^{-1/2})\Big) \\
&= \phi_k(Y_k) - \ln(e - q),
\end{aligned} \tag{3.10}
$$

where

$$
q \;=\; \begin{pmatrix} -S_k^{-1}\Delta s \\ -\lambda(Y_k^{-1/2}(\Delta Y)Y_k^{-1/2}) \\ \lambda(U_k^{-1/2}(\Delta Y)U_k^{-1/2}) \end{pmatrix}. \tag{3.11}
$$

Note that $e^T q = \nabla\phi_k(Y_k)^T \mathbf{svec}\Delta Y$ and $\|q\|^2 = \mathbf{svec}(\Delta Y)^T H_k\, \mathbf{svec}(\Delta Y) \le 3/4$.

By applying Lemma 3.1 to (3.10), we have

$$
\begin{aligned}
\phi_k(\tilde{Y}) - \phi_k(Y_k) &\le e^T q + \frac{\|q\|^2}{2(1-\|q\|)} \\
&= \nabla\phi_k(Y_k)^T \mathbf{svec}\Delta Y + \frac{\mathbf{svec}(\Delta Y)^T H_k\, \mathbf{svec}(\Delta Y)}{(2-\sqrt{3})} \\
&\le \delta_k(Y_k)\sqrt{\tilde{\omega}^T M_k \tilde{\omega}} \;+\; \frac{\tilde{\omega}^T M_k \tilde{\omega}}{(2-\sqrt{3})} \\
&\le \frac{\sqrt{3}}{2}\eta \;+\; \frac{3}{4(2-\sqrt{3})}.
\end{aligned} \tag{3.12}
$$

Note that in the next to last inequality above, we used the Cauchy inequality to derive the result: $\nabla\phi_k(Y_k)^T \mathbf{svec}(\Delta Y) \le \delta_k(Y_k)\sqrt{\tilde{\omega}^T M_k \tilde{\omega}}$.

Substituting the result in (3.12) into (3.9), we prove the lemma. □

From Lemma 3.4, we see that the upper bound for the dual potential value $\phi_{k+1}(\tilde{Y})$ contains the term $-\ln\tilde{\xi}$. If we were to consider the dual potential value alone, then finding an upper bound for $-\ln\tilde{\xi}$ would be necessary. But we have found that finding a tight upper bound for this term is difficult. As a result, we have decided to consider the primal-dual potential value, for which finding an upper bound for $-\ln\tilde{\xi}$ is not necessary. To this end, let us define the primal potential function associated with Ω_k. For any $\psi_k(x, Z, V) \in \mathbb{R}^{n_k}_{++} \times \mathcal{S}^m_{++} \times \mathcal{S}^m_{++}$ that satisfies $\mathcal{A}x - \mathbf{svec}(Z) + \mathbf{svec}(V) = 0$, the primal potential of (x, Z, V) is defined by

$$
\psi_k(x, V, Z) \;=\; c^T x + I \bullet V - \ln x - \ln\det Z - \ln\det V. \tag{3.13}
$$

The primal-dual potential function associated with Ω_k is

$$
\Lambda_k(Y, x, Z, V) \;=\; \phi_k(Y) + \psi_k(x, Z, V).
$$

We should emphasize that the primal-dual potential function is introduced solely for the purpose of estimating the potential value of $(\tilde{Y}, \tilde{s}, \tilde{x}, \tilde{Z}, \tilde{V})$. It is not needed in our cutting plane algorithm described in section 2.

Now we shall proceed to establish an analogue of Lemma 3.4 for the primal potential function. Before doing that, we need the following lemma.

LEMMA 3.5. *For the directions* $(\Delta x, \Delta Z, \Delta V)$ *given in* (3.4)*, the following inequality holds:*

$$\left|c^T\Delta x - e^T X_k^{-1}\Delta x - Z_k^{-1}\bullet\Delta Z - V_k^{-1}\bullet\Delta V + I\bullet\Delta V + d^T\tilde{\omega}\right| \;\le\; \frac{\eta}{1-\eta}\frac{\sqrt{3}}{2}. \tag{3.14}$$

Proof. Noting that $d = \mathcal{B}_k^T\mathbf{svec}(Y_k)$ and $H_k\mathbf{svec}(\Delta Y) = -\mathcal{B}_k\tilde{\omega}$, we have

$$d^T\tilde{\omega} = -\mathbf{svec}(Y_k)^T\left(\mathcal{A}\,\Delta x + \mathbf{svec}[Y_k^{-1}(\Delta Y)Y_k^{-1}] + \mathbf{svec}[U_k^{-1}(\Delta Y)U_k^{-1}]\right).$$

Let $X_k = \mathrm{diag}(x^k)$ and $S_k = \mathrm{diag}(s^k)$. Then

$$\begin{aligned}
&\left|c^T\Delta x + d^T\tilde{\omega} - e^T X_k^{-1}\Delta x - Z_k^{-1}\bullet\Delta Z - V_k^{-1}\bullet\Delta V + I\bullet\Delta V\right| \\
&= \left|e^T(S_k - X_k^{-1})\Delta x \;+\; (Z_k^{-1} - Y_k)\bullet(Y_k^{-1}\Delta Y Y_k^{-1}) \;+\; (U_k - V_k^{-1})\bullet(U_k^{-1}\Delta Y U_k^{-1})\right| \\
&= \left|(e - X_k^{-1}(s^k)^{-1})^T S_k^{-1}\Delta s\right| \;+\; \left|(Y_k^{-1/2}Z_k^{-1}Y_k^{-1/2} - I)\bullet(Y_k^{-1/2}\Delta Y Y_k^{-1/2})\right| \\
&\quad + \left|(I - U_k^{-1/2}V_k^{-1}U_k^{-1/2})\bullet(U_k^{-1/2}\Delta Y U_k^{-1/2})\right| \\
&\le \|e - X_k^{-1}(s^k)^{-1}\|\,\|S_k^{-1}\Delta s\| \;+\; \|Y_k^{-1/2}Z_k^{-1}Y_k^{-1/2} - I\|_F\,\|Y_k^{-1/2}\Delta Y Y_k^{-1/2}\|_F \\
&\quad + \|U_k^{-1/2}V_k^{-1}U_k^{-1/2} - I\|_F\,\|U_k^{-1/2}\Delta Y U_k^{-1/2}\|_F \\
&\le \left(\|e - X_k^{-1}(s^k)^{-1}\|^2 \;+\; \|Y_k^{-1/2}Z_k^{-1}Y_k^{-1/2} - I\|_F^2 \;+\; \|U_k^{-1/2}V_k^{-1}U_k^{-1/2} - I\|_F^2\right)^{1/2} \\
&\quad \times \left(\|S_k^{-1}\Delta s\|^2 \;+\; \|Y_k^{-1/2}\Delta Y Y_k^{-1/2}\|_F^2 \;+\; \|U_k^{-1/2}\Delta Y U_k^{-1/2}\|_F^2\right)^{1/2} \\
&\le \eta\left(Y_k^{-1}, (s^k)^{-1}, (x^k)^{-1}, Z_k^{-1}, V_k^{-1}\right)\left(\mathbf{svec}(\Delta Y)^T H_k\mathbf{svec}(\Delta Y)\right)^{1/2} \\
&\le \frac{\eta}{1-\eta}\frac{\sqrt{3}}{2}.
\end{aligned}$$

Note that, in the last inequality above, we used (3.7) and the fact that

$$\eta(Y^{-1}, (s)^{-1}, (x)^{-1}, Z^{-1}, V^{-1}) \;\le\; \frac{\eta}{1-\eta}. \qquad \square$$

LEMMA 3.6. *For the point* $(\tilde{x}, \tilde{Z}, \tilde{V})$ *constructed in* (3.6)*, the following inequality holds:*

$$\psi_{k+1}(\tilde{x}, \tilde{Z}, \tilde{V}) \le \psi_k(x^k, Z_k, V_k) + \frac{3}{4(1-\eta)(2-2\eta-\sqrt{3})} + \frac{\eta}{1-\eta}\frac{\sqrt{3}}{2} - \ln\tilde{\omega}. \tag{3.15}$$

Proof. We have

$$\begin{aligned}
\psi_{k+1}(\tilde{x}, \tilde{Z}, \tilde{V}) &= c^T x^k + c^T\Delta x - \ln x^k - \ln(e + X_k^{-1}\Delta x) - \ln\det Z_k \\
&\quad - \ln\det(I + Z_k^{-1/2}(\Delta Z)Z_k^{-1/2}) - \ln\det V_k \\
&\quad - \ln\det(I + V_k^{-1/2}(\Delta V)V_k^{-1/2}) + I\bullet V_k + I\bullet\Delta V + d^T\tilde{\omega} - \ln\tilde{\omega} \\
&= \psi_k(x^k, Z_k, V_k) + c^T\Delta x + I\bullet\Delta V + d^T\tilde{\omega} - \ln\tilde{\omega} - \ln(e+p),
\end{aligned} \tag{3.16}$$

where

$$p = \begin{pmatrix} X_k^{-1}\Delta x \\ \lambda(Z_k^{-1/2}(\Delta Z)Z_k^{-1/2}) \\ \lambda(V_k^{-1/2}(\Delta V)V_k^{-1/2}) \end{pmatrix}.$$

Note that $e^T p = e^T X_k^{-1}\Delta x + Z_k^{-1} \bullet \Delta Z + V_k^{-1} \bullet \Delta V$, and by Lemma 3.2,

$$\begin{aligned} \|p\|^2 &= \|X_k^{-1}\Delta x\|^2 + \|Z_k^{-1/2}(\Delta Z)Z_k^{-1/2}\|_F^2 + \|V_k^{-1/2}(\Delta V)V_k^{-1/2}\|_F^2 \\ &\le \frac{1}{(1-\eta)^2}\left(\|S_k^{-1}\Delta s\|^2 + \|Y_k^{-1/2}(\Delta Y)Y_k^{-1/2}\|_F^2 + \|U_k^{-1/2}(\Delta Y)U_k^{-1/2}\|_F^2\right) \\ &= \frac{1}{(1-\eta)^2}\,\mathbf{svec}(\Delta Y)^T H_k\,\mathbf{svec}(\Delta Y) \;\le\; \frac{1}{(1-\eta)^2}\,\frac{3}{4}. \end{aligned} \tag{3.17}$$

By Lemma 3.1 and (3.17), we get from (3.16),

$$\begin{aligned} \psi_{k+1}(\tilde{x},\tilde{Z},\tilde{V}) &\le \psi_k(x^k,Z_k,V_k) + c^T\Delta x + I \bullet \Delta V + d^T\tilde{\omega} - \ln\tilde{\omega} - e^T p + \frac{\|p\|^2}{2(1-\|p\|)} \\ &\le \psi_k(x^k,Z_k,V_k) + c^T\Delta x + I \bullet \Delta V + d^T\tilde{\omega} - \ln\tilde{\omega} - e^T p + \frac{3}{4(1-\eta)(2-2\eta-\sqrt{3})}. \end{aligned}$$

By applying Lemma 3.5 and (3.7), we prove the lemma. ∎

The next lemma is an analogue of Lemma 3.4 for the primal-dual potential function.

LEMMA 3.7. *Suppose $\eta(Y_k, s^k, x^k, Z_k, V_k)$ is an η-approximate center with $\eta < 1 - \sqrt{3}/2$. Then*

$$\Lambda_{k+1}(\tilde{Y},\tilde{x},\tilde{Z},\tilde{V}) \;\le\; \Lambda_k(Y_k,x^k,Z_k,V_k) + \beta(\eta) + p_k\left(\frac{3}{4} + \ln 2p_k\right),$$

where

$$\beta(\eta) \;=\; \eta\frac{\sqrt{3}}{2} + \frac{3}{4(2-\sqrt{3})} + \frac{\eta}{1-\eta}\frac{\sqrt{3}}{2} + \frac{3}{4(1-\eta)(2-2\eta-\sqrt{3})}. \tag{3.18}$$

Proof. Combining the results in Lemmas 3.4 and 3.6, we have

$$\Lambda_{k+1}(\tilde{Y},\tilde{x},\tilde{Z},\tilde{V}) \;\le\; \Lambda_k(Y_k,x^k,Z_k,V_k) + \beta(\eta) - \ln\tilde{\omega}\tilde{\xi}. \tag{3.19}$$

Note that

$$\begin{aligned} -\ln\tilde{\omega}\tilde{\xi} &= p_k\ln 2p_k \;+\; \sum_{i=1}^{p_k} -\ln\left(1-(1-2p_k\tilde{\omega}_i\tilde{\xi}_i)\right) \\ &\le p_k\ln 2p_k \;+\; \sum_{i=1}^{p_k}\left[(1-2p_k\tilde{\omega}_i\tilde{\xi}_i) + \frac{|1-2p_k\tilde{\omega}_i\tilde{\xi}_i|^2}{2(1-|1-2p_k\tilde{\omega}_i\tilde{\xi}_i|)}\right] \\ &\le p_k\ln 2p_k \;+\; \frac{3}{4}p_k. \end{aligned} \tag{3.20}$$

By substituting (3.20) into (3.19), the lemma is proved. ∎

With Lemma 3.7, we can finally establish an explicitly known upper bound for the primal-dual potential value $\Lambda_{k+1}(\tilde{Y}, \tilde{x}, \tilde{Z}, \tilde{V})$.

THEOREM 3.8. *Suppose that* $(\bar{Y}_{k+1}, \bar{x}^{k+1}, \bar{Z}_{k+1}, \bar{V}_{k+1})$ *is the analytic center of* Ω_{k+1}*, and* $(\tilde{Y}, \tilde{x}, \tilde{Z}, \tilde{V})$ *is the point constructed in* (3.5)–(3.6)*. Then*

$$\Lambda_{k+1}(\tilde{Y}, \tilde{x}, \tilde{Z}, \tilde{V}) - \Lambda_{k+1}(\bar{Y}_{k+1}, \bar{x}^{k+1}, \bar{Z}_{k+1}, \bar{V}_{k+1}) \;\le\; p_k \left(\ln 2p_k - \frac{1}{4} \right) + \beta(\eta) + \frac{2\eta^2}{1-\eta^2}, \tag{3.21}$$

where $\beta(\eta)$ *is the constant given in* (3.18).

Proof. Suppose that $(Y_k, s^k, x^k, Z_k, V_k)$ is an η-approximate center of Ω_k with $\eta < 1 - \sqrt{3}/2$. We have

$$\begin{aligned}
&\Lambda_{k+1}(\tilde{Y}, \tilde{x}, \tilde{Z}, \tilde{V}) - \Lambda_{k+1}(\bar{Y}_{k+1}, \bar{x}^{k+1}, \bar{Z}_{k+1}, \bar{V}_{k+1}) \\
&= \Lambda_{k+1}(\tilde{Y}, \tilde{x}, \tilde{Z}, \tilde{V}) - \Lambda_k(Y_k, x^k, Z_k, V_k) \;+\; \Lambda_k(\bar{Y}_k, \bar{x}^k, \bar{Z}_k, \bar{V}_k) \\
&\quad - \Lambda_{k+1}(\bar{Y}_{k+1}, \bar{x}^{k+1}, \bar{Z}_{k+1}, \bar{V}_{k+1}) \\
&\quad + \Lambda_k(Y_k, x_k, Z_k, V_k) - \Lambda_k(\bar{Y}_k, \bar{x}^k, \bar{Z}_k, \bar{V}_k).
\end{aligned} \tag{3.22}$$

It is readily shown that

$$\begin{aligned}
&\Lambda_k(\bar{Y}_k, \bar{x}^k, \bar{Z}_k, \bar{V}_k) - \Lambda_{k+1}(\bar{Y}_{k+1}, \bar{x}^{k+1}, \bar{Z}_{k+1}, \bar{V}_{k+1}) \\
&\qquad = (n_k + 2m) - (n_k + p_k + 2m) \;=\; -p_k.
\end{aligned} \tag{3.23}$$

Next we need to get an upper bound for the term $\Lambda_k(Y_k, x^k, Z_k, V_k) - \Lambda_k(\bar{Y}_k, \bar{x}^k, \bar{Z}_k, \bar{V}_k)$ in (3.22). By following the proof of Lemma 2.1 in [1] and using the quadratic convergence result in [8], it is readily shown that

$$\phi_k(Y_k) \;\le\; \phi_k(\bar{Y}_k) + \frac{\eta^2}{1-\eta^2}. \tag{3.24}$$

Similarly, it can be shown that

$$\psi_k(x^k, Z_k, V_k) \;\le\; \psi_k(\bar{x}^k, \bar{Z}_k, \bar{V}_k) + \frac{\eta^2}{1-\eta^2}. \tag{3.25}$$

Combining (3.24) and (3.25), we get

$$\Lambda_k(Y_k, x^k, Z_k, V_k) - \Lambda_k(\bar{Y}_k, \bar{x}^k, \bar{Z}_k, \bar{V}_k) \;\le\; \frac{2\eta^2}{1-\eta^2}. \tag{3.26}$$

By putting the results of Lemma 3.7, (3.23), and (3.26) into (3.22), the theorem is proved. ∎

With the estimate of $\Lambda_{k+1}(\tilde{Y}, \tilde{x}, \tilde{Z}, \tilde{V})$ in Theorem 3.8, we are now ready to estimate the number of dual Newton steps required to find an approximate analytic center for Ω_{k+1} by using the point $\tilde{Y}$ as the initial point.

THEOREM 3.9. *Given an* η*-approximate center* Y_k *of* Ω_k*, with* $\eta < 1 - \sqrt{3}/2$*, the total number of dual Newton steps required to find an* η*-approximate center* Y_{k+1} *of* Ω_{k+1} *is*

$$O\left(p_k \ln p_k\right),$$

where the constant $O(1)$ is independent of k.

Proof. By Theorem 2.2.3 in [7], each dual Newton step reduces Λ_{k+1} by a positive constant $\gamma = \bar{\delta} - \ln(1+\bar{\delta})$, as long as a point $\hat{Y}$ with $\delta_{k+1}(\hat{Y}) < \bar{\delta} < 1$ is not yet found, while keeping the primal iterate fixed. Now, starting at $(\tilde{Y}, \tilde{s}, \tilde{x}, \tilde{Z}, \tilde{V})$, the total value of Λ_{k+1} which needs to be reduced is not more than $\Lambda_{k+1}(\tilde{Y}, \tilde{x}, \tilde{Z}, \tilde{V}) - \Lambda_{k+1}(\bar{Y}, \bar{x}, \bar{Z}, \bar{V})$; thus Theorem 3.8 implies that at most

$$\frac{1}{\gamma}\left[p_k\left(\ln p_k + \ln 2 - \frac{1}{4}\right) + \beta(\eta) + \frac{2\eta^2}{1-\eta^2}\right]$$

Newton steps are required to reach a point $\hat{Y}$ with $\delta_{k+1}(\hat{Y}) \le \bar{\delta}$. From $\hat{Y}$ onwards, by Lemma 4.3 in [8], quadratic convergence can be achieved, and thus it needs at most $\ln(\ln(\bar{\delta}/\eta))$ additional full Newton steps to find a point Y_{k+1} satisfying $\delta_{k+1}(Y_{k+1}) \le \eta$. (We can choose, for example, $\bar{\delta} = 0.9$ and $\eta = 0.1$; then $\ln(\ln(\bar{\delta}/\eta)) \le 4$.) ∎

4. Potential changes and complexity. Recall that $\Omega_k = \{Y \in \Omega_0 \mid \mathcal{A}^T \mathbf{svec} Y \le c\}$. Suppose that Y_k is an η-approximate analytic center of Ω_k with $\eta < 1 - \sqrt{3}/2$. Let

$$\mathcal{B}_k \;=\; (\mathbf{svec} A_{n_k+1}, \ldots, \mathbf{svec} A_{n_k+p_k}), \qquad d \;=\; \mathcal{B}_k^T \mathbf{svec}(Y_k).$$

Then

$$\Omega_{k+1} = \{Y \in \Omega_k \mid \mathcal{B}_k^T \mathbf{svec} Y \;\le\; d\}.$$

Let $\bar{Y}_k$ and $\bar{Y}_{k+1}$ be the analytic centers of Ω_k and Ω_{k+1}, respectively. Let

$$\bar{r}_k = \sqrt{\lambda_{\max}(\mathcal{B}_k^T \bar{H}_k^{-1} \mathcal{B}_k)}, \tag{4.1}$$

where $\bar{H}_k = \nabla^2 \phi(\bar{Y}_k)$.

In this section, we estimate the amount that the dual potential will increase when the working set changes from Ω_k to Ω_{k+1}. To this end, we first establish a lemma that is an extension of a result in [10].

LEMMA 4.1. *Suppose that n, p are positive integers and v is a positive n-vector with $e^T v = n$. Then for any positive constant η the following inequality holds:*

$$(\|v - e\| + \eta)^p \prod_{i=1}^{n} v_i \;\le\; p^{p+1}\theta^p,$$

where θ is a positive constant no greater than $1.3 + \eta$.

Proof. We need to consider only the case in which $n \ge 2$, as the inequality holds trivially when $n = 1$. Consider the maximization problem

$$\begin{aligned} \max \qquad & f(v) \;:=\; \|v - e\|^p \prod_{i=1}^{n} \alpha_i \\ \text{such that} \quad & e^T v = n. \end{aligned}$$

It is shown in [10] that the maximizer v has the form

$$v_1 = \gamma, \quad v_2 = \cdots = v_n = \frac{n-\gamma}{n-1}, \quad \text{where } \gamma > 1,$$

and

$$f(v) \le \left(\frac{n}{n-1}\right)^{p/2} (p+1)^{p+1} \exp\left(\frac{-p(p+2)}{p+1}\right).$$

Thus

$$\begin{aligned}(\|v-e\|+\eta)^p \prod_{i=1}^n v_i &= \left(\|v-e\| \prod_{i=1}^n v_i^{1/p} + \eta \prod_{i=1}^n v_i^{1/p}\right)^p \\ &\le \left[\left(\|v-e\|^p \prod_{i=1}^n v_i\right)^{1/p} + \eta\right]^p, \qquad \left(\text{since } \textstyle\prod_{i=1}^n v_i \le (\frac{e^T v}{n})^n = 1\right) \\ &\le \left[\left(\frac{n}{n-1}\right)^{1/2} (p+1)^{(p+1)/p} \exp\left(\frac{-(p+2)}{p+1}\right) + \eta\right]^p \\ &\le p^{p+1}\theta^p,\end{aligned}$$

where

$$\theta = \max_{n\ge 2, p\ge 1} \left\{ \left(\frac{n}{n-1}\right)^{1/2} \left(1+\frac{1}{p}\right)^{1+1/p} \exp\left(\frac{-1}{p+1} - 1\right) + \frac{\eta}{p} p^{-1/p} \right\} \le 1.3+\eta. \qquad \square$$

LEMMA 4.2. *Suppose Y_k is an approximate analytic center of Ω_k with $\delta_k(Y_k) \le \eta < 1-\sqrt{3}/2$. Then*

$$\phi_{k+1}(\Omega_{k+1}) \ \ge \ \phi_k(\Omega_k) - \frac{p_k}{2} \ln(p_k\, \bar{r}_k^2\, \theta^2) - \ln p_k, \tag{4.2}$$

where θ is a constant depending only on η.

Proof. For simplicity, we will drop the subscripts k and $k+1$ in our notations in this proof and denote, for example, Ω_k and Ω_{k+1} by Ω and Ω_+, respectively.

Let $\bar{U} = I - \bar{Y}$, $\bar{U}_+ = I - \bar{Y}_+$, and

$$\bar{s}^+ \ = \ c - \mathcal{A}^T \mathbf{svec}(\bar{Y}_+), \qquad \bar{s} \ = \ c - \mathcal{A}^T \mathbf{svec}(\bar{Y}), \qquad \bar{t} \ = \ d - \mathcal{B}^T \mathbf{svec}(\bar{Y}_+).$$

Let

$$\bar{G} \ = \ [\mathcal{A}\bar{S}^{-1}, \ -\bar{Y}^{-1/2} \circledast \bar{Y}^{-1/2}, \ \bar{U}^{-1/2} \circledast \bar{U}^{-1/2}].$$

Note that $\bar{H} = \bar{G}\bar{G}^T$.

First, we establish an upper bound for $\ln \prod_{j=1}^p \bar{t}_j$. We have

$$\begin{aligned}\bar{t} = \mathcal{B}^T(\mathbf{svec}Y - \mathbf{svec}\bar{Y}_+) \ &= \ \mathcal{B}^T \bar{H}^{-1} \bar{G} \left(\bar{G}^T \mathbf{svec}Y - \bar{G}^T \mathbf{svec}\bar{Y}_+\right) \\ &= (\bar{G}^T \bar{H}^{-1} \mathcal{B})^T \left(\bar{G}^T \mathbf{svec}(Y - \bar{Y}) - \bar{G}^T \mathbf{svec}(\bar{Y}_+ - \bar{Y})\right).\end{aligned}$$

Thus

$$\|\bar{t}\| \le \|\bar{G}^T \bar{H}^{-1} \mathcal{B}\| \left(\|\bar{G}^T \mathbf{svec}(Y - \bar{Y})\| + \|\bar{G}^T \mathbf{svec}(\bar{Y}_+ - \bar{Y})\|\right).$$

By part (iii) of Theorem 2.2.2 in [7], we have

$$\|\bar{G}^T \mathbf{svec}(Y - \bar{Y})\| \ = \ \eta(\bar{x}, s, Y, \bar{Z}, \bar{V}) \le \frac{1 - [1 - 3\delta(Y)]^{1/3}}{[1 - 3\delta(Y)]^{1/3}} \le 3\delta(Y) \ \le \ 3\eta.$$

Thus

$$\|\bar{t}\| \le \bar{r}\ \left(3\eta + \|\bar{G}^T\mathbf{svec}(\bar{Y}_+ - \bar{Y})\|\right).$$

Hence

$$\ln\prod_{j=1}^{p}\bar{t}_j = \frac{p}{2}\left(\frac{1}{p}\sum_{j=1}^{p}\ln\bar{t}_j^2\right) \ \le\ \frac{p}{2}\ln\left(\frac{\sum_{j=1}^{p}\bar{t}_j^2}{p}\right) \ =\ \frac{p}{2}\ln\|\bar{t}\|^2 - \frac{p}{2}\ln p$$

$$\le p\ln\left(3\eta + \|\bar{G}^T\mathbf{svec}(\bar{Y}_+ - \bar{Y})\|\right) + p\ln\bar{r} - \frac{p}{2}\ln p, \tag{4.3}$$

and the desired upper bound is established.

Now observe that

$$\phi_+(\Omega_+) - \phi(\Omega) \ =\ -\ln\prod_{j=1}^{p}\bar{t}_j - \ln\left(\prod_{i=1}^{n}\frac{\bar{s}_i^+}{\bar{s}_i}\ \frac{\det\bar{Y}_+}{\det\bar{Y}}\ \frac{\det\bar{U}_+}{\det\bar{U}}\right).$$

Using the bound in (4.3), we have

$$\phi(\Omega_+) - \phi(\Omega) \ \ge\ \frac{p}{2}\ln p - p\ln\bar{r} - \ln\left(3\eta + \|\bar{G}^T\mathbf{svec}(\bar{Y}_+ - \bar{Y})\|\right)^p\prod_{i=1}^{n}\frac{\bar{s}_i^+}{\bar{s}_i}\ \frac{\det\bar{Y}_+}{\det\bar{Y}}\ \frac{\det\bar{U}_+}{\det\bar{U}}. \tag{4.4}$$

The inequality (4.2) follows, once we have shown that

$$\left(3\eta + \|\bar{G}^T\mathbf{svec}(\bar{Y}_+ - \bar{Y})\|\right)^p\prod_{i=1}^{n}\frac{\bar{s}_i^+}{\bar{s}_i}\ \frac{\det\bar{Y}_+}{\det\bar{Y}}\ \frac{\det\bar{U}_+}{\det\bar{U}} \ \le\ p^{p+1}\,\theta^p. \tag{4.5}$$

Note that

$$\begin{aligned}
&\|\bar{G}^T\mathbf{svec}(\bar{Y}_+ - \bar{Y})\|^2\\
&\quad= \mathbf{svec}(\bar{Y}_+ - \bar{Y})^T\left[\mathcal{A}\bar{S}^{-2}\mathcal{A}^T + \bar{Y}^{-1}\circledast\bar{Y}^{-1} + \bar{U}^{-1}\circledast\bar{U}^{-1}\right]\mathbf{svec}(\bar{Y}_+ - \bar{Y})\\
&\quad= (\bar{s} - \bar{s}^+)^T\bar{S}^{-2}(\bar{s} - \bar{s}^+) \ +\ \mathbf{svec}(\bar{Y}_+ - \bar{Y})^T(\bar{Y}^{-1}\circledast\bar{Y}^{-1})\mathbf{svec}(\bar{Y}_+ - \bar{Y})\\
&\qquad+\mathbf{svec}(\bar{U} - \bar{U}_+)^T(\bar{U}^{-1}\circledast\bar{U}^{-1})\mathbf{svec}(\bar{U} - \bar{U}_+)\\
&\quad= \left\|\begin{pmatrix} e - \bar{S}^{-1}\bar{s}^+\\ e - \lambda(\bar{Y}^{-1/2}\bar{Y}_+\bar{Y}^{-1/2})\\ e - \lambda(\bar{U}^{-1/2}\bar{U}_+\bar{U}^{-1/2})\end{pmatrix}\right\|^2,
\end{aligned}$$

and, by using (2.1), we have

$$\begin{aligned}
&e^T\bar{S}^{-1}\bar{s}^+ \ +\ e^T\lambda(\bar{Y}^{-1/2}\bar{Y}_+\bar{Y}^{-1/2}) \ +\ e^T\lambda(\bar{U}^{-1/2}\bar{U}_+\bar{U}^{-1/2})\\
&\quad= \bar{x}^T(c - \mathcal{A}^T\mathbf{svec}\bar{Y}_+) + \bar{Z}\bullet\bar{Y}_+ + \bar{V}\bullet\bar{U}_+\\
&\quad= \bar{x}^Tc + \bar{V}\bullet I\\
&\quad= \bar{x}^T(c - \mathcal{A}^T\mathbf{svec}\bar{Y}) + \bar{Z}\bullet\bar{Y} + \bar{V}\bullet\bar{U}\\
&\quad= \bar{x}^T\bar{s} + \bar{Z}\bullet\bar{Y} + \bar{V}\bullet\bar{U}\\
&\quad= n + 2m.
\end{aligned}$$

By Lemma 4.1, (4.5) is proved. ∎

The complexity analysis is based on the following idea. For the sequence of working sets Ω_k, we can establish upper and lower bounds on $\phi(\Omega_k)$. The upper bound is approximately $n_k \ln \epsilon^{-1}$, which is a consequence of the assumption that Γ contains a ball of radius ϵ and the fact that Ω_k is defined by n_k cuts. The lower bound is obtained by estimating $-\sum_{i=0}^{k-1} p_i \ln \bar{r}_i$, which is a consequence of Lemma 4.2. A sophisticated estimation of $\bar{r}_k$ gives rise to a lower bound that is proportional to $n_k \ln(n_k/m^3)$. The algorithm must terminate before the lower and upper bounds conflict with each other.

We first establish an upper bound for $\phi_k(\Omega_k)$.

LEMMA 4.3. *Let $\Omega_k \supset \Gamma$ be defined by n_k linear inequalities and the positive semidefinite constraint. Suppose Assumptions* A1–A3 *hold. Then*

$$\phi_k(\Omega_k) \leq -(n_k + 2m) \ln \epsilon.$$

Proof. Assumptions A1–A3 imply that there exists a point $Y^c \in \Gamma$ such that

(i) all eigenvalues of Y^c and $I - Y^c$ are greater than or equal to ϵ;

(ii) for any $A \in \mathcal{S}^m$ with $\|A\|_F = 1$ and $\alpha \in \mathbb{R}$, if $\Gamma \subset \{Y \mid A \bullet Y \leq \alpha\}$, then $\alpha - A \bullet Y^c \geq \epsilon$.

We will briefly describe how to prove $\lambda(Y_c) \geq \epsilon e$, before continuing with the proof of the lemma. Suppose that λ_j is an eigenvalue of Y_c, and v_j is a corresponding unit eigenvector. Consider the matrix $\hat{Y}_c := Y_c - \lambda_j v_j v_j^T$. Since this matrix has a zero eigenvalue, it lies on the boundary of Ω_0, and by Assumption A3, we have

$$\epsilon \;\leq\; \|\hat{Y}_c - Y_c\|_F \;=\; \|\lambda_j v_j v_j^T\|_F \;=\; \lambda_j \|v_j\|^2 \;=\; \lambda_j.$$

The fact that $\lambda(Y_c) \leq (1 - \epsilon)e$ can be proved similarly.

Now we continue with the proof of the lemma. Since $\Gamma \subset \Omega_k$,

$$\phi_k(\Omega_k) \;\leq\; \phi_k(Y^c) \;=\; -\sum_{i=1}^{n_k} \ln(c_i - A_i \bullet Y^c) - \ln \det Y^c - \ln \det(I - Y^c).$$

Noting that $\|A_i\|_F = 1$, $c_i - A_i \bullet Y^c \geq \epsilon$, and

$$\det Y^c \;=\; \prod_{i=1}^{m} \lambda_i(Y^c) \geq \epsilon^m, \qquad \det(I - Y^c) \;=\; \prod_{i=1}^{m} \lambda_i(I - Y^c) \geq \epsilon^m,$$

we have the desired inequality. ∎

Now we turn to finding a lower bound for $\phi_k(\Omega_k)$. By Lemma 4.2, we have

$$\phi_k(\Omega_k) \geq \phi_0(\Omega_0) - \sum_{i=0}^{k-1} p_i \ln \bar{r}_i - \frac{1}{2} \sum_{i=0}^{k-1} p_i \ln p_i - \sum_{i=0}^{k-1} p_i \ln \theta - \sum_{i=0}^{k-1} \ln p_i. \tag{4.6}$$

Obviously, we need to estimate $\bar{r}_i$ for each i. We first seek to bound $\bar{H}_i^{-1}$ by D_i^{-1}, where D_i is defined as follows. Let $D_0 = 8I$, where I is the identity matrix of order $\bar{m} \times \bar{m}$. For $i = 1, 2, \ldots$, let

$$D_i = D_0 + \frac{1}{m} \sum_{j=0}^{i-1} \mathcal{B}_j \mathcal{B}_j^T,$$

where $\mathcal{B}_j = [\mathbf{svec}A_{n_j+1}, \ldots, \mathbf{svec}A_{n_j+p_j}]$.

LEMMA 4.4. *Let A_{n_i+j} (with $\|A_{n_i+j}\|_F = 1$), $j = 1, \ldots, p_i$, be the cuts generated from the approximate analytic center $Y_i \in \Omega_i$, $i = 0, \ldots, k-1$. Let $c_{n_i+j} = \mathbf{svec}(A_{n_i+j})^T \mathbf{svec}(Y_i)$, $j = 1, \ldots, p_i$, $i = 1, \ldots, k$. For any point $Y \in \Omega_k$, let $s = c - \mathcal{A}^T \mathbf{svec}Y$, where*

$$\mathcal{A} \;=\; [\mathcal{B}_0 \;\; \mathcal{B}_1 \;\; \cdots \;\; \mathcal{B}_{k-1}], \qquad c \;=\; \begin{bmatrix} \mathcal{B}_0^T \mathbf{svec}(Y_0) \\ \vdots \\ \mathcal{B}_{k-1}^T \mathbf{svec}(Y_{k-1}) \end{bmatrix}.$$

Then

$$s_{n_i+j} \;\le\; \sqrt{m} \quad \forall \;\; j = 1, \ldots, p_i, \;\; i = 1, \ldots, k, \qquad \nabla^2 \phi_k(Y) \;\succeq\; D_k.$$

In particular, $\bar{H}_k = \nabla^2 \phi_k(\bar{Y}_k) \succeq D_k$.

Proof. We first estimate s_{n_i+j}. We have

$$\begin{aligned} s_{n_i+j} &= c_{n_i+j} - \mathbf{svec}(A_{n_i+j})^T \mathbf{svec}(Y) \;=\; (\mathbf{svec}A_{n_i+j})^T (\mathbf{svec}Y_i - \mathbf{svec}Y) \\ &\le \|\mathbf{svec}A_{n_i+j}\| \, \|\mathbf{svec}Y_{i-1} - \mathbf{svec}Y\| \\ &= \|\mathbf{svec}Y_i - \mathbf{svec}Y\| \;=\; \|Y_i - Y\|_F \\ &= \left[\sum_{j=1}^{m} \lambda_j^2 (Y_i - Y) \right]^{1/2} \;\le\; \sqrt{m}\,. \end{aligned} \tag{4.7}$$

The last inequality holds because, by Assumption A2,

$$I \;\succeq\; Y_i \;\succeq\; Y_i - Y \;\succeq\; -Y \;\succeq\; -I,$$

implying that $e \ge \lambda(Y_i - Y) \ge -e$.

Next, let $U = I - Y$ and $S_i = \mathrm{diag}(s_{n_i+1}, \ldots, s_{n_i+p_i})$. Then

$$\begin{aligned} \nabla^2 \phi_k(Y) &= Y^{-1} \circledast Y^{-1} + U^{-1} \circledast U^{-1} + \mathcal{A} S^{-2} \mathcal{A}^T \\ &= Y^{-1} \circledast Y^{-1} + U^{-1} \circledast U^{-1} + \sum_{i=0}^{k-1} \mathcal{B}_i \, S_i^{-2} \, \mathcal{B}_i^T \\ &\succeq 8I + \frac{1}{m} \sum_{i=0}^{k-1} \mathcal{B}_i \, \mathcal{B}_i^T \;=\; D_k. \end{aligned} \tag{4.8}$$

Note that in deriving (4.8) we used the fact that $S_i \preceq \sqrt{m}\, I_{p_i}$ for each i, and that

$$Y^{-1} \circledast Y^{-1} + U^{-1} \circledast U^{-1} \succeq 8I. \qquad \square$$

In our complexity analysis, we will make the following assumptions:

A4. $p_{\max} \le m$, where $p_{\max} \;=\; \max\{p_i \mid i = 0, 1, \ldots\}$.

A5. Let $\bar{M}_i = \mathcal{B}_i^T \bar{H}_i^{-1} \mathcal{B}_i$. There exists a fixed constant $\tau \ge 1$ such that, for each $i = 0, 1, \ldots,$

$$\lambda_{\max}(\bar{M}_i) \;\le\; \tau \, \frac{\mathrm{Tr}(\bar{M}_i)}{p_i}.$$

Assumption A4 is made for technical reasons. It is used in the proof of Lemma 4.5. Such an assumption also appeared in the papers [3] and [10]. Note that Assumption A4 can be relaxed to $p_{\max} \le O(m)$. But, for simplicity, we fix the constant at 1.

Note that Assumption A5 holds trivially with $\tau = p_{\max}$. For the special case in which a single cut is used in each iteration, it holds with $\tau = 1$. Thus by fixing τ at an intermediate value between 1 and $p_{\max}$, we admit only cuts that are sufficiently good in the sense that the matrix $\bar{M}_i$ cannot have too many small eigenvalues. Of course, one may not want to fix τ at the extreme value 1, since then the criterion is likely to reject most of the cuts unless there are many mutually orthogonal (with respect to $\bar{H}_i^{-1}$) cuts.

The main advantage of having Assumption A5 is that in each oracle call we have an objective criterion to select only cuts that are useful from among a possibly large number of ineffective cuts. In this way, the number of cuts added in each iteration will not be unnecessarily large, and hence the computational time in each iteration will not grow as rapidly as in the case where the cuts are admitted unchecked. The choice of τ in practice would depend on the problem at hand. It should dynamically be adjusted as information on the quality of the cuts is obtained as the cutting algorithm progresses. If the choice of τ is too stringent and many good cuts are rejected, then we can progressively increase its value so that more good cuts are selected.

However, without a priori information on the quality of the cuts, we propose to choose τ to be a small constant, say 5, based on the following empirical observation. We conducted numerical experiments on random matrices of the form $V^T V$, where $V \in \mathbb{R}^{\bar{m} \times p}$, for $p = 1, \ldots, m$, and $m = 10, 20, \ldots, 260$. The elements of V are drawn independently from the standard normal distribution. We computed the ratio between the largest eigenvalue of $V^T V$ and $\mathrm{Tr}(V^T V)/p$ for each V, and found that these ratios are less than 2 for all of the 3510 cases tested.

Now let us continue with our complexity analysis. Let

$$w_i^2 \;=\; \mathrm{Tr}(\mathcal{B}_i^T D_i^{-1} \mathcal{B}_i).$$

Since

$$p_i\, \bar{r}_i^2 \;\le\; \tau\, \mathrm{Tr}(\mathcal{B}_i^T \bar{H}_i^{-1} \mathcal{B}_i) \;\le\; \tau\, \mathrm{Tr}(\mathcal{B}_i^T D_i^{-1} \mathcal{B}_i) \;=\; \tau w_i^2,$$

we have

$$\sum_{i=0}^{k-1} p_i\, \bar{r}_i^2 \;\le\; \tau \sum_{i=0}^{k-1} w_i^2. \tag{4.9}$$

Next, we establish an upper bound for the right-hand side of the above inequality. Its proof is modeled after that of [10, Lemma 3.5]. However, we have simplified the proof by considering all the cuts simultaneously instead of handling them one by one as in [10].

LEMMA 4.5.

$$\sum_{i=0}^{k-1} w_i^2 \;\le\; \frac{9m\bar{m}}{8} \ln\left(1 + \frac{n_k}{8m\bar{m}}\right).$$

Proof. From the equation

$$\det D_{i+1} \;=\; \det D_i \prod_{j=1}^{p_i} \left[1 + \frac{1}{m} \lambda_j(\mathcal{B}_i^T D_i^{-1} \mathcal{B}_i)\right],$$

we have

$$\ln\det D_{i+1} - \ln\det D_i = \sum_{j=1}^{p_i} \ln\left[1 + \frac{1}{m}\lambda_j(\mathcal{B}_i^T D_i^{-1}\mathcal{B}_i)\right]$$

$$\geq \frac{8}{9m}\sum_{j=1}^{p_i}\lambda_j(\mathcal{B}_i^T D_i^{-1}\mathcal{B}_i) \;=\; \frac{8}{9m}w_i^2, \tag{4.10}$$

where we used the fact that $\lambda_{\max}(\mathcal{B}_i^T D_i^{-1}\mathcal{B}_i) \leq \lambda_{\max}(\mathcal{B}_i^T\mathcal{B}_i)/8 = \|\mathcal{B}_i\|_F^2/8 = p_i/8$, and the inequality $\ln(1+x) \geq 8x/9$ for $0 \leq x \leq 1/8$. We also made use of Assumption A4, which yields that $p_i \leq m$.

From (4.10), it follows immediately that

$$\ln\det D_k - \ln\det D_0 \;\geq\; \frac{8}{9m}\sum_{i=0}^{k-1} w_i^2. \tag{4.11}$$

However,

$$\frac{1}{\bar{m}}\ln\det D_k \leq \ln\frac{\mathrm{Tr}(D_k)}{\bar{m}} \;=\; \ln\frac{1}{\bar{m}}\left[\mathrm{Tr}(D_0) + \frac{1}{m}\sum_{i=0}^{k-1}\mathrm{Tr}(\mathcal{B}_i\mathcal{B}_i^T)\right]$$

$$= \ln\frac{1}{\bar{m}}\left[8\bar{m} + \frac{1}{m}\sum_{i=0}^{k-1} p_i\right] \;=\; \ln\left(8 + \frac{n_k}{m\bar{m}}\right),$$

implying that

$$\ln\det D_k - \ln\det D_0 \;\leq\; \bar{m}\ln\left(1 + \frac{n_k}{8m\bar{m}}\right). \tag{4.12}$$

Combining (4.11) and (4.12), the lemma is proved. □

With the above lemma, we can now formally state a lower bound for $\phi_k(\Omega_k)$.

LEMMA 4.6. *Suppose that Assumptions* A1–A5 *hold. Then*

$$\phi_k(\Omega_k) \geq -\frac{1}{2}(2m+n_k)\ln\left[\frac{4m + 9\tau\, m\bar{m}\ln(1+\frac{n_k}{8m\bar{m}})}{8(2m+n_k)}\right]$$
$$-\frac{1}{2}\sum_{i=0}^{k-1} p_i\ln p_i - n_k\ln\theta - \sum_{i=0}^{k-1}\ln p_i,$$

where θ is the constant that appeared in (4.2).

Proof. The proof is similar to that of Theorem 10 in [10], after using (4.9) and Lemma 4.5. □

We will next estimate the number of oracle calls required to find a feasible point of Γ.

LEMMA 4.7. *Suppose that Assumptions* A1–A5 *hold. Then the analytic center cutting plane method stops with a feasible point before k violates the following inequality:*

$$\frac{\epsilon^2}{p_{\max}m\bar{m}} \;\leq\; \frac{4/\bar{m} + 9\tau\ln(1+\frac{n_k}{8m\bar{m}})}{8(2m+n_k)}\exp\left(\frac{2n_k\ln\theta + 2\sum_{i=0}^{k-1}\ln p_i}{n_k+2m}\right). \tag{4.13}$$

Proof. From Lemmas 4.3 and 4.6, we have

$$-(2m+n_k)\ln\epsilon \geq -\frac{1}{2}(2m+n_k)\ln\left[\frac{4m+9\tau m\bar{m}\ln(1+\frac{n_k}{8m\bar{m}})}{8(2m+n_k)}\right] -\frac{1}{2}\sum_{i=0}^{k-1}p_i\ln p_i - n_k\ln\theta - \sum_{i=0}^{k-1}\ln p_i.$$

Thus, the algorithm must terminate before k violates the above inequality; i.e., the algorithm must terminate before k violates the following inequality:

$$\frac{\epsilon^2}{m\bar{m}} \leq \frac{4/\bar{m}+9\tau\ln(1+\frac{n_k}{8m\bar{m}})}{8(2m+n_k)}\exp\left(\frac{\sum_{i=0}^{k-1}p_i\ln p_i + 2n_k\ln\theta + 2\sum_{i=0}^{k-1}\ln p_i}{2m+n_k}\right). \tag{4.14}$$

Since $\sum_{i=0}^{k-1}p_i\ln p_i \leq \sum_{i=0}^{k-1}p_i\ln p_{\max} = n_k\ln p_{\max}$, the algorithm must terminate before k violates the inequality in the lemma. ∎

THEOREM 4.8. *Suppose that Assumptions* A1–A5 *hold. Then the analytic center cutting plane method terminates in at most* $O^*(m^3\tau\, p_{\max}\ln p_{\max}/\epsilon^2)$ *Newton steps, where the notation* O^* *means that lower order terms are ignored. The total number of cuts added is not more than* $O^*(m^3\tau\, p_{\max}/\epsilon^2)$.

Proof. Ignoring lower order terms (assuming $k \gg m$) and by the assumption that τ is a constant independent of $p_{\max}$, the above lemma implies that the algorithm stops as soon as k satisfies

$$\frac{n_k}{\tau\,\ln(n_k/m^3)} \geq O\left(\frac{m^3 p_{\max}}{\epsilon^2}\right).$$

For large k, $\ln n_k$ is negligible compared to n_k; hence the algorithm requires at most

$$n_k = O^*\left(\frac{m^3\tau\, p_{\max}}{\epsilon^2}\right)$$

cuts. By Theorem 3.9, the total number of Newton steps is

$$O\left(\sum_{i=1}^{k}p_i\ln p_i\right) \leq O(n_k\ln p_{\max}) = O^*\left(\frac{m^3\tau\, p_{\max}\ln p_{\max}}{\epsilon^2}\right).$$

The theorem is proved. ∎

For feasibility problems in $\mathbb{R}^m_+$, $\bar{m}$ should be replaced by m in Lemma 4.7. Thus the complexity bound is $O(m^2\tau\, p_{\max}\ln p_{\max}/\epsilon^2)$ for the number of required Newton steps. This bound is better than the bounds obtained in [2], [5], and [10].

Acknowledgment. We thank the referees for their constructive comments that greatly helped to improve the paper.

REFERENCES

[1] J.-L. GOFFIN, Z.-Q. LUO, AND Y. YE, *Complexity analysis of an interior cutting plane method for convex feasibility problems*, SIAM J. Optim., 6 (1996), pp. 638–652.

[2] J.-L. GOFFIN AND J.-P. VIAL, *Multiple cuts in the analytic center cutting plane method*, SIAM J. Optim., 11 (2000), pp. 266–288.

[3] J.-L. GOFFIN AND J.-P. VIAL, *Convex Nondifferentiable Optimization: A Survey Focused on the Analytic Center Cutting Plane Method*, HEC/Logilab Technical report 99-02, University of Geneva, Geneva, Switzerland, 1999.

[4] R.A. Horn and C.R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1985.
[5] Z.-Q. Luo, *Analysis of a cutting plane method that uses weighted analytic center and multiple cuts*, SIAM J. Optim., 7 (1997), pp. 697–716.
[6] Y. Nesterov, *Cutting plane algorithms from analytic centers: Efficiency estimates*, Math. Programming, 69 (1995), pp. 149–176.
[7] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM Stud. Appl. Math., SIAM, Philadelphia, 1994.
[8] J. Sun, K.C. Toh, and G. Zhao, *An analytic center cutting plane method for semidefinite feasibility problems*, Math. Oper. Res., to appear.
[9] Y. Ye, *A potential reduction algorithm allowing column generation*, SIAM J. Optim., 2 (1992), pp. 7–20.
[10] Y. Ye, *Complexity analysis of the analytic center cutting plane method that uses multiple cuts*, Math. Programming, 78 (1997), pp. 85–104.
[11] Y. Ye, *Interior Point Algorithms: Theory and Analysis*, John Wiley & Sons, New York, 1997.